Computing an Optimal Entanglement Path with Throughput and Fidelity Considerations

Guoliang Xue, Nageswara S. V. Rao, Don Towsley, Gayane Vardoyan, Zunzheng Zhang, Xuanli Lin, Muneer Alshowkan, Joseph M. Lukens, Nicholas A. Peters and Saikat Guha

***Abstract*—Entanglement distribution is a core function of quantum networks essential for operations including teleportation, distributed quantum sensing, and multisite computation. Entanglement throughput and fidelity are two critical performance measures that depend on the quantum transmission along the links and swapping operations at the repeaters along the path. We study the problem of computing a end-to-end entanglement path that satisfies both fidelity and throughput requirements, leveraging qubit buffers at the nodes and considering the sequential swapping order. We show that the general problem of simultaneously satisfying both metrics to be NP-hard, and develop an algorithm to maximize throughput subject to a given fidelity threshold. We introduce the concepts of entanglement probability distribution and path domination and exploit them in the design of our algorithm. Extensive numerical results show that our algorithm can find optimal solutions in networks with thousands of nodes in less than a second. We also describe practical and possible implementation aspects of this algorithm in terms of devices and architecture support.**

Keywords: **Quantum networks, entanglement routing, entanglement swapping, throughput, fidelity, path metrics.**

1. Introduction

Entanglement distribution is a core function of quantum networks due to its essential role in implementing several important capabilities including teleportation, quantum key distribution, distributed quantum sensing, and multisite quantum computing [19, 26–28]. We consider quantum networks consisting of first- or second-generation quantum repeaters whose primary goal is to distribute entanglement between specified pairs of source and destination nodes. To accomplish this task, neighboring nodes first generate entanglement at the physical link level; these entangled states (which cover relatively short distances) then undergo *entanglement swapping* operations that, if successful, result in direct *end-to-end* (E2E) entanglement shared between the source and destination nodes. Nodes running quantum applications can then consume the generated entanglement as a fungible resource.

This research is sponsored in part by PiQSci project of Advanced Scientific Computing Research program, U.S. Department of Energy, and is performed at Oak Ridge National Laboratory managed by UT-Battelle, LLC for U.S. Department of Energy under Contract No. DE-AC05-00OR22725. This manuscript has been co-authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

The throughput and fidelity of the entanglement distribution are two important performance metrics of quantum networks [27, 28, 34]. They depend on the transmission of photons along the fiber links between the repeaters or routers on the E2E path and also swapping operations at them. Under the "sequential order" [6, 27], swapping operations are performed sequentially at the repeaters or routers starting from s and progressing towards t. We focus on the sequential swapping order and design an optimal algorithm that can compute an E2E entanglement path with maximum throughput subject to a fidelity constraint.

We formulate an abstraction of path computation with the primary goal of establishing entanglement, *i.e.*, Bell pairs of desired fidelity and throughput, between source node s and destination node t. We show that the general problem of satisfying both throughput and fidelity requirements is NP-hard and develop an algorithm to compute an s-t path that maximizes throughput subject to a fidelity constraint.

Our algorithm first generates a number of bound channels (defined in Section 3), attempts to generate entanglement on each of these bound channels, and then sequentially swaps them from s to t, resulting in an s-t entanglement with maximum throughput while ensuring the fidelity constraint. We introduce the concept of entanglement probability distribution (EPD) for an entanglement path, that summarizes essential properties of entanglement distribution. We define a dominance relation among entanglement paths, and prove that it holds as the entanglement paths are extended by an edge-width pair in the proposed algorithm. Its practical implementation requires devices that implement entanglement generation and swapping operations, together with an architecture that provides the needed state information and conventional network operations.

The main contributions of this paper are as follows.

- We formalize the entanglement path computation problem by specifying not only the E2E path, but also the number of bound channels to be used on each link of the path.
- We prove that the general problem of computing an s-t entanglement path satisfying both fidelity and throughput constraints is NP-hard.
- We present an algorithm for computing an s-t entanglement path that maximizes throughput subject to a given fidelity constraint. The algorithm starts at the source node and extends a partial entanglement path by appending an edge to it, one hop at a time.

- We introduce the concept of entanglement probability distribution and a dominance relation that prunes partial entanglement paths that cannot be extended to produce an optimal path. This technique significantly reduces the running time of the algorithm. Extensive numerical results show that our algorithm can compute an optimal solution in networks with thousands of nodes in less than a second.

The rest of this paper is organized as follows. We place the proposed work within the broader context of quantum networking and path computation algorithms in Section 2. We present a graph model of a quantum network and the definition of entanglement paths in Section 3. We define metrics for entanglement paths and formulate the problems in Section 4. Critical properties of entanglement paths are established in Section 5. The path computation algorithm and its analysis are presented in Section 6. Evaluation results are presented in Section 7. We conclude the paper, with implementation aspects of the algorithm and future research in Section 8. For ease of reading, proof of NP-hardness of the problem and differentiation of the model in this paper with flow-based models are deferred to the appendix.

2. Related Works

Practical realization of long-distance entanglement distribution in a quantum network requires a combination of hardware—e.g., sources, detectors, and repeaters [2, 33]—supported by architectures and protocols that specify path computation and routing [27, 29]. Entanglement paths have been studied extensively from various perspectives, including architecture, performance analysis, practical implementations, incorporation of physical parameters and algorithms [7, 36]. We briefly review works that are directly related to our path computation formulation. Our focus is on the algorithmic aspects of computing entanglement distribution paths by abstracting the properties of communication links and entanglement swapping operations using a graph model. The authors of [38] presented approximation algorithms for a similar problem which also employs entanglement purification. We present a practical optimal algorithm.

Pirandola *et al.* [20] study multi-path routing in a diamond topology. Schoute *et al.* [25] develop a comprehensive framework to study quantum network routing in ring or sphere topologies. Das *et al.* [11] compare different special topologies for entanglement routing. Pant *et al.* [19] develop and study entanglement routing methods in grid networks. Chakraborty *et al.* [4] study greedy routing methods in ring and grid networks. Caleffi [3] derives a closed-form expression for the expected entanglement rate over a chain of repeaters. Ciobanu *et al.* [8] study entanglement distribution in a multi-ring topology. The above works focus on special network topologies. Shi and Qian [26] study entanglement path computation in a general network. Our paper follows the model used in [19] and [26].

Different physical parameters are incorporated into entanglement paths in a number of works [20, 29]. Specifically, Caleffi [3] takes into consideration a comprehensive set of architectural parameters that affect the rate, such as memory coherence times and imperfect measurements. A routing algorithm is presented and shown to be optimal in entanglement rate, although its runtime scales poorly with the number of simple paths in the network. Moreover, an implicit assumption of the work is that all link-level entanglement generation takes place simultaneously—a potentially impractical assumption for quantum network architectures that adopt the "connectionless" paradigm and cannot rely on resource reservation. The author further points out that it does not optimize the swapping strategy for the underlying network architecture, an important consideration in the presence of decoherence.

In terms of algorithms, our work is related to both quickest and multi-path computations in networks [21, 35]; specifically, it is a special case of multiflow problems [1, 17] but more general than shortest path problems that can be solved by Dijkstra's algorithm [9]. Van Meter *et al.* [29] describe a class of path computation algorithms in quantum networks that can be solved using Dijkstra's algorithm by expressing costs as additive scalar weights on nodes and links. The presence of buffers makes our problem more complex since the throughput cannot be reduced to a single additive scalar edge- or node-cost needed by Dijkstra's algorithm in [29].

3. Graph Model of Quantum Networks

We present a graph model compatible with most network architectures [27, 28], that employ quantum repeaters that (a) possess quantum memories with sufficiently long coherence times to accommodate the amount of time needed to perform entanglement swapping across the diameter of the network in a *sequential* manner (*i.e.*, originating at the source node and progressing gradually towards the destination node), and (b) perform gates of adequately high fidelity to result in E2E entanglement of admissible quality for the requesting application.

A. Graph Representation of a Quantum Network

We represent a quantum network as an undirected graph $G = (\mathbb{V}, \mathbb{E}, \mathbb{C}, Q, p, w, q)$ with edge and vertex attributes:

- $\mathbb{V} = \{v_1, v_2, \ldots, v_n\}$ is the set of vertices, each corresponding to a quantum repeater or an end node;
- $\mathbb{E} \subseteq \{(u, v) | u, v \in \mathbb{V}, u \neq v\}$ is the set of undirected edges, each corresponding to an optical fiber link connecting nodes u and v;
- $\mathbb{C}_{u,v} \subseteq \{1, 2, \ldots, C\}$ is the set of available undirected quantum channels on edge (u, v), $\forall (u, v) \in \mathbb{E}$;
- $Q_v \geq 0$ is the number of available quantum memories at vertex v, $\forall v \in \mathbb{V}$;
- $p_{u,v} \in [0, 1]$ is the entanglement generation success probability for a channel in $\mathbb{C}_{u,v}$, $\forall (u, v) \in \mathbb{E}$, when required qubits are available at both u and v;

- $w_{u,v} \in [0,1]$ is the Werner parameter [5] for any entanglement over a quantum channel in $\mathbb{C}_{u,v}$, $\forall (u,v) \in \mathbb{E}$;
- $q_v \in [0,1]$ is the success probability of an entanglement swapping attempt at vertex v, $\forall v \in \mathbb{V}$.

In the above, $n = |\mathbb{V}|$ is the number of vertices, $m = |\mathbb{E}|$ is the number of edges, $C = \max_{(u,v)\in\mathbb{E}} |\mathbb{C}_{u,v}|$ is the maximum number of quantum channels an edge may have. Since the graph is undirected, (u,v) and (v,u) denote the same edge. As a result, $\mathbb{C}_{u,v} = \mathbb{C}_{v,u}$, $p_{u,v} = p_{v,u}$, and $w_{u,v} = w_{v,u}$ for each edge $(u,v) \in \mathbb{E}$.

We use the terms *vertex* and *node* interchangeably. Similarly, we use the terms *edges* and *links* interchangeably. Although the edges are undirected, for technical clarity, we impose a direction on a path. Therefore, u_0-u_1-$\cdots$-u_{h-1}-u_h denotes the h-hop path *from* u_0 *to* u_h via intermediate nodes $u_1, u_2, \ldots, u_{h-1}$. We abbreviate *quantum channels* as *channels*.

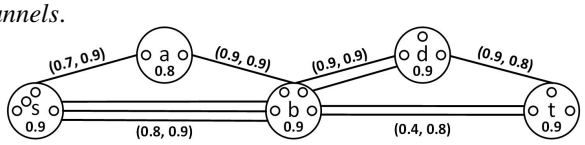


Figure 1. Example quantum network. A vertex v is illustrated by a circle, with v and q_v inside. The number of small circles inside the circle for vertex v denotes the number of available quantum memories at vertex v. For each edge $e = (u,v)$, the number of line segments connecting u and v denotes $|\mathbb{C}_{u,v}|$; the 2-tuple next to these line segments denotes $(p_{u,v}, w_{u,v})$.

Figure 1 shows a quantum network with $\mathbb{V} = \{a,b,d,s,t\}$ and $\mathbb{E} = \{(a,b),(a,s),(b,d),(b,s),(b,t),(d,t)\}$; $Q_a = 2$, $Q_b = 4$, $Q_d = 3$, $Q_s = 4$, $Q_t = 3$; $q_a = 0.8$, $q_b = 0.9$, $q_d = 0.9$, $q_s = 0.9$, $q_t = 0.9$; $|\mathbb{C}_{a,b}| = 1$, $|\mathbb{C}_{a,s}| = 1$, $|\mathbb{C}_{b,d}| = 2$, $|\mathbb{C}_{b,s}| = 3$, $|\mathbb{C}_{b,t}| = 2$, $|\mathbb{C}_{d,t}| = 1$; $p_{a,b} = 0.9$, $p_{a,s} = 0.7$, $p_{b,d} = 0.9$, $p_{b,s} = 0.8$, $p_{b,t} = 0.4$, $p_{d,t} = 0.9$; $w_{a,b} = 0.9$, $w_{a,s} = 0.9$, $w_{b,d} = 0.9$, $w_{b,s} = 0.9$, $w_{b,t} = 0.8$, $w_{d,t} = 0.8$. We will use this example network for illustration in this paper.

To aid the reader, we summarize the notations used in this paper in addition to the ones described in Section 3A.

Table I
ADDITIONAL NOTATIONS USED IN THIS PAPER

Notations	Description
$e(u,v)$	entanglement connecting nodes u and v.
$[(u,v),c]$	edge-width pair; (u,v) is the edge, c is the width.
$\pi_{s,t}$	an ePath connecting node s to node t.
$\pi \oplus [(u,v),c]$	extension of ePath π by edge-width pair $[(u,v),c]$.
$W(e(u,v))$	Werner parameter of entanglement $e(u,v)$.
$W(\pi)$	Werner parameter of ePath π.
$F(e(u,v))$	fidelity of entanglement $e(u,v)$.
$F(\pi)$	fidelity of ePath π.
$T(\pi)$	throughput of ePath π.
$R(\pi)$	residual quantum memory of ePath π.
$p_k^{[(u,v),c]}$	probability that $\mathrm{EG}((u,v),c)$ produces exactly k entanglements on edge (u,v).
$\mathbf{p}^\pi$	entanglement probability distribution (EPD) of ePath π: $\mathbf{p}^\pi = (\mathbf{p}_0^\pi, \mathbf{p}_1^\pi, \ldots, \mathbf{p}_C^\pi)$.
X_π	number of E2E entanglements generated on ePath π.
$X \geq_{\mathrm{st}} Y$	X is stochastically larger than or equal to Y.
$\pi_1 \trianglerighteq \pi_2$	π_1 is stochastically stronger than π_2: $X_{\pi_1} \geq_{\mathrm{st}} X_{\pi_2}$.
$\pi_1 \succeq \pi_2$	π_1 dominates π_2.

B. Entanglement Generation

A node $u \in \mathbb{V}$ can assign one of its qubits to a quantum channel in $\mathbb{C}_{u,v}$ on an adjacent edge (u,v) [19, 26], so that each qubit is assigned to at most one channel, and each end node of a channel is assigned at most one qubit. A channel that has qubits assigned to both end nodes is called a *bound channel* [6].

To *generate an entanglement* on a bound channel on link (u,v), adjacent repeaters u and v, equipped with quantum memories (e.g., atomic ensembles [24], ion traps [13], or color centers [23]), are optically interfaced across the link. A variety of approaches for entanglement generation can be used, including (i) detection of one or two photons at the midpoint, (ii) distribution of entangled photons from the midpoint to each node, or (iii) single-photon transmission across the full link [18]. Each method has its own advantages and disadvantages that depend on the physical platform and experimental conditions. Importantly, our formalism applies equally well to all of them; the only requirement is that the entanglement generation process between nodes u and v be *heralded* and characterized by some probability of success.

Definition 1 (Edge-width Pair): For edge (u,v) and a positive integer $c \leq \min\{|\mathbb{C}_{u,v}|, Q_u, Q_v\}$, the 2-tuple $[(u,v),c]$ is called *edge-width pair*, where (u,v) and c are the *edge* and *width*, respectively. □

Given an edge-width pair $[(u,v),c]$, we attempt to generate c entanglement connecting u and v simultaneously using c channels in $\mathbb{C}_{u,v}$. Because we are using c bound channels on edge (u,v), this requires c quantum memories at node u and c quantum memories at node v.

Let $\mathrm{EG}((u,v),c)$ denote the process of simultaneously attempting to generate entanglement on c bound channels on link (u,v). The probability that the operation $\mathrm{EG}((u,v),c)$ results in exactly k entanglement between u and v is

$$\mathbf{p}_k^{[(u,v),c]} = \begin{cases} \binom{c}{k} p_{u,v}^k (1-p_{u,v})^{c-k}, & 0 \leq k \leq c \\ 0, & c < k \leq C \end{cases} \tag{3.1}$$

under the statistical independence conditions on the entanglement generation process [6, 19, 26]. We assume that each generated entanglement $e(u,v)$ has Werner parameter $W(e(u,v)) = w_{u,v}$, hence its fidelity with respect to an ideal Bell state is $F(e(u,v)) = \frac{1+3W(e(u,v))}{4}$ [5]. Although a significant simplification, the Werner state assumption permits conservative "worst-case" fidelity estimates [14], facilitates straightforward swapping calculations [16, 30], and reflects many types of random independent noise sources that impact entanglement generation in practice [18].

C. Entanglement Swapping

Let $e(u,v)$ denote an entanglement between possibly non-adjacent nodes u and v. Two entanglement are said to be *adjacent* if they share a common end node. Let $e(x,y)$ and $e(y,z)$ be two adjacent entanglement. After y receives heralding signals from x and z indicating successful generation

of $e(x,y)$ and $e(y,z)$, y attempts a *swap* of $e(x,y)$ and $e(y,z)$ to obtain entanglement connection $e(x,z)$. As with entanglement generation, the specific swapping procedure depends on the physical system, but is often realized by optical excitation of co-located memories followed by photodetection (of either one or two photons) [13, 18, 24]. Let $W(e(x,y))$ and $W(e(y,z))$ denote the Werner parameters of $e(x,y)$ and $e(y,z)$, respectively. Conveniently, the swapped entanglement $e(x,z)$ is also a Werner state, with parameter equal to the product of the parameters for $e(x,y)$ and $e(y,z)$, *i.e.*, $W(e(x,z)) = W(e(x,y)) \times W(e(y,z))$ [16, 30]. The fidelity of $e(x,z)$ can be computed accordingly, *i.e.*, $F(e(x,z)) = \frac{1+3W(e(x,z))}{4}$.

Suppose there are l entanglement connecting nodes x and y, and r entanglement connecting nodes y and z; we can simultaneously attempt $\min\{l,r\}$ entanglement swaps at node y. The probability that these swaps result in exactly k entanglements connecting x and z is $\rho(l,r,k,q_y)$,

$$\rho(l,r,k,q_y) = \begin{cases} \binom{\min\{l,r\}}{k} q_y^k (1-q_y)^{\min\{l,r\}-k}, & k \le \min\{l,r\} \\ 0, & k > \min\{l,r\} \end{cases} \tag{3.2}$$

under the assumption that entanglement swapping processes are statistically independent at the repeater [6, 19, 26].

D. Entanglement Path

We are interested in computing an s-t entanglement path which is defined as follows:

Definition 2 (Entanglement Path): An entanglement path $\pi_{s,t}$ connecting node $s \in \mathbb{V}$ to node $t \in \mathbb{V}$ is an ordered sequence of edge-width pairs $[(u_0,u_1),c_1]$, $[(u_1,u_2),c_2]$, $\ldots$, $[(u_{h-1},u_h),c_h]$ such that

$$c_i \le |\mathbb{C}_{u_{i-1},u_i}|,\ i = 1,2,\ldots,h, \tag{3.3}$$

$$c_1 \le Q_{u_0},\ c_h \le Q_{u_h}, \tag{3.4}$$

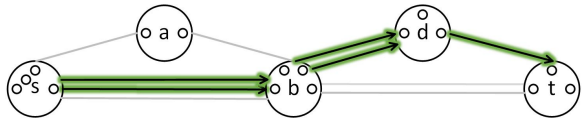
$$c_i + c_{i+1} \le Q_{u_i},\ i = 1,2,\ldots,h-1, \tag{3.5}$$

where $u_0 = s$, $u_h = t$, and u_0-u_1-$\cdots$-u_h is a simple path from $u_0 = s$ to $u_h = t$. □

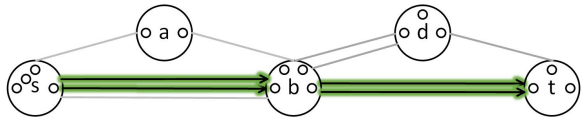
In the above, c_i denotes the number of simultaneous entanglement generations to be performed on link (u_{i-1},u_i), $i = 1,2,\ldots,h$. Since we need to have c_i bound channels on link (u_{i-1},u_i), $i = 1,2,\ldots,h$, we need to satisfy the quantum memory constraints (3.3)-(3.5). We also write π_{u_0,u_h} as $\pi_{u_0,u_h} = [(u_0,u_1),c_1]$-$[(u_1,u_2),c_2]$-$\cdots$-$[(u_{h-1},u_h),c_h]$. We abbreviate entanglement path to ePath, and abbreviate π_{u_0,u_h} to π if there is no ambiguity.

Although the quantum network is undirected, we impose a direction for ePaths for clarity: $\pi_{s,t}$ is an ePath from s to t. We assume that $u_0 = s$ and $u_h = t$ in the rest of this paper unless specified otherwise.

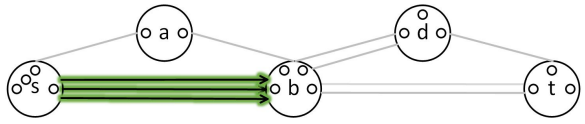
Figures 2(a)-2(e) illustrate different ePaths for the network in Figure 1. (a) and (b) illustrate two different ePaths from s to t; (c) and (d) illustrate two different ePaths from s to b; while (e) illustrates an ePath from s to d.



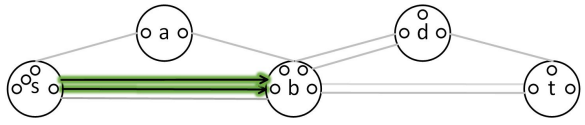
(a) An ePath from s to t



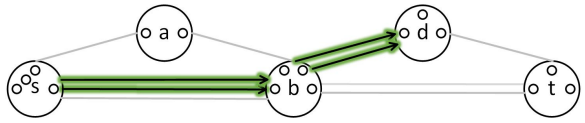
(b) Another ePath from s to t



(c) An ePath from s to b



(d) Another ePath from s to b



(e) An ePath from s to d

Figure 2. Different ePaths in the quantum network in Figure 1. Arrows indicate the imposed direction of ePaths. (a) ePath $[(s,b),2]$-$[(b,d),2]$-$[(d,t),1]$. (b) ePath $[(s,b),2]$-$[(b,t),2]$. (c) ePath $[(s,b),3]$. (d) ePath $[(s,b),2]$. (e) ePath $[(s,b),2]$-$[(b,d),2]$.

E. Entanglement Generation and Swapping along an ePath

Given an ePath $\pi_{u_0,u_h} = [(u_0,u_1),c_1]$-$[(u_1,u_2),c_2]$-$\cdots$-$[(u_{h-1},u_h),c_h]$, we can establish zero or more entanglement connecting nodes u_0 and u_h, using Algorithm 1, as follows.

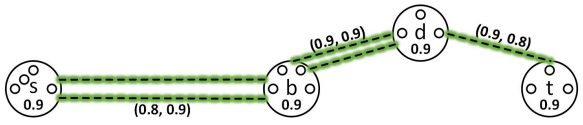
In Phase 1 (Lines 1-3), the protocol makes c_i entanglement generation attempts on c_i bound channels on edge (u_{i-1},u_i), for $i \in \{1,2,\ldots,h\}$. These probabilistic operations can be performed in parallel (or in any order). We use $\bar{c}_i$ to denote the number of entanglement *established* on link (u_{i-1},u_i) as a result of these c_i attempts, for $i \in \{1,2,\ldots,h\}$. Note that $\bar{c}_i$ is a random integer between 0 and c_i, for $i \in \{1,2,\ldots,h\}$.

In Phase 2 (Lines 4-11), the protocol performs entanglement swapping along the ePath in sequential order. At node u_i, there are $\tilde{c}_i$ entanglement connecting node u_0 to node u_i, and $\bar{c}_{i+1}$ entanglement connecting node u_i to node u_{i+1}. If $\tilde{c}_i = 0$ or $\bar{c}_{i+1} = 0$, the protocol stops, without establishing any entanglement connecting node u_0 to node u_h. Otherwise, the protocol performs $\min\{\tilde{c}_i, \bar{c}_{i+1}\}$ entanglement swap attempts at node u_i, establishing $\tilde{c}_{i+1}$ entanglement connecting node u_0 to node u_{i+1}. The process continues for up to $h-1$ iterations.

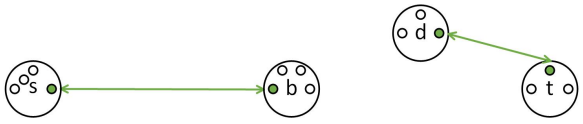
Suppose that ePath $\pi_{s,t}$ is the one illustrated in Figure 2(a). We have $h = 3$; $u_0 = s$, $u_1 = b$, $u_2 = d$, $u_3 = t$;

Algorithm 1: $\texttt{EGS}(\pi_{u_0,u_h})$

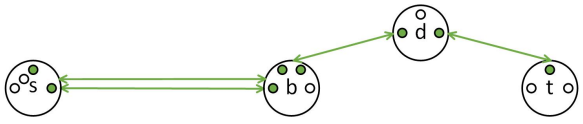
// Entanglement generation and swapping on ePath π_{u_0,u_h}
Input: ePath $\pi_{u_0,u_h} = [(u_0,u_1),c_1]$-$[(u_1,u_2),c_2]$-$\cdots$-$[(u_{h-1},u_h),c_h]$.
Output: Number of entanglement connecting u_0 and u_h corresponding to π_{u_0,u_h}
/* Phase 1: entanglement generation */
for $\forall i \in \{1,2,\ldots,h\}$ **do**
  Apply $\texttt{EG}((u_{i-1},u_i),c_i)$;
  $\bar{c}_i \leftarrow$ number of entanglement connecting u_{i-1} and u_i successfully established;
/* Phase 2: entanglement swapping */
$\tilde{c}_1 \leftarrow \bar{c}_1$;
for $i = 1,2,\ldots,h-1$ **do**
  if ($\tilde{c}_i == 0$ **or** $\bar{c}_{i+1} == 0$) **goto** 13;
  Let $e_1(u_0,u_i), e_2(u_0,u_i), \ldots, e_{\tilde{c}_i}(u_0,u_i)$ be the $\tilde{c}_i$ entanglement connecting u_0 and u_i;
  Let $e_1(u_i,u_{i+1}), e_2(u_i,u_{i+1}), \ldots, e_{\bar{c}_{i+1}}(u_i,u_{i+1})$ be the $\bar{c}_{i+1}$ entanglement connecting u_i and u_{i+1};
  for $\forall j = 1,2,\ldots,\min\{\tilde{c}_i,\bar{c}_{i+1}\}$ **do**
    Swap $e_j(u_0,u_i)$ with $e_j(u_i,u_{i+1})$ at node u_i.
  $\tilde{c}_{i+1} \leftarrow$ number of entanglement connecting u_0 and u_{i+1} successfully established;
stop: Output $\tilde{c}_h$; // Exit 1: established $\tilde{c}_h$ entanglement
stop: Output 0; // Exit 2: established 0 entanglement



(a) 2 entanglement generation attempts on link (s,b); 2 attempts on (b,d); 1 attempt on (d,t).



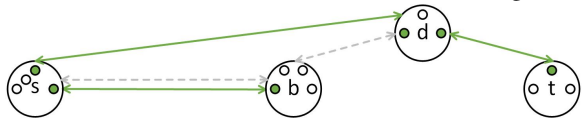
(b) Case with $(\bar{c}_1,\bar{c}_2,\bar{c}_3) = (1,0,1)$



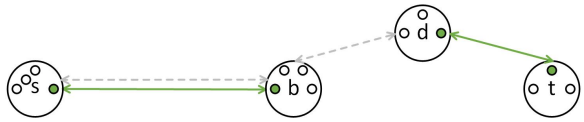
(c) Case with $(\bar{c}_1,\bar{c}_2,\bar{c}_3) = (2,1,1)$

Figure 3. Possible results of Phase 1. Dashed lines denote entanglement generation attempts. Lines with arrows denote established entanglement. (a) Entanglement generation attempts on the ePath $[(s,b),2]$-$[(b,d),2]$-$[(d,t),1]$; (b) The case $(\bar{c}_1,\bar{c}_2,\bar{c}_3) = (1,0,1)$; (c) The case $(\bar{c}_1,\bar{c}_2,\bar{c}_3) = (2,1,1)$.

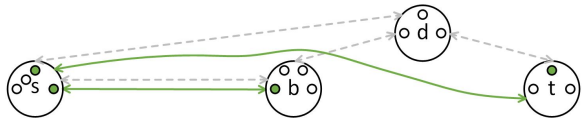
$c_1 = 2$, $c_2 = 2$, $c_3 = 1$. The entanglement generation attempts are illustrated in Figure 3(a). After Phase 1, there are a total of $2^2 \times 2^2 \times 2^1 = 32$ possible results. Here we make two entanglement generation attempts on two bound channels on link (s,b), two attempts on two bound channels on link (b,d), and one attempt on one bound channel on link (d,t). For the entanglement generations on link (s,b), it is possible for both to be successful, with a probability of $\binom{2}{2}p_{s,b}^2$; it is possible for one to be successful and the other unsuccessful, with a probability of $\binom{2}{1}p_{s,b}(1-p_{s,b})$; it is possible for both to be unsuccessful, with a probability of $\binom{2}{0}(1-p_{s,b})^2$. Similar analysis can be applied to the entanglement generations on links (b,d) and (d,t). The results on the three links along the path are independent. If we just consider the number of different values of the tuple $(\bar{c}_1,\bar{c}_2,\bar{c}_3)$, there are a total of 18 cases: where $\bar{c}_1 \in \{0,1,2\}$, $\bar{c}_2 \in \{0,1,2\}$, and $\bar{c}_3 \in \{0,1\}$. The probability of having the super-case with $(\bar{c}_1,\bar{c}_2,\bar{c}_3) = (1,0,1)$ is $\left[\binom{2}{1} \times p_{s,b}(1-p_{s,b})\right] \times \left[\binom{2}{0} \times (1-p_{b,d})^2\right] \times \left[\binom{1}{1} \times p_{d,t}\right] = 0.32 \times 0.01 \times 0.9 = 0.00288$. This is shown in Figure 3(b). The probability of having the case with $(\bar{c}_1,\bar{c}_2,\bar{c}_3) = (2,1,1)$ is $\left[\binom{2}{2} \times p_{s,b}^2\right] \times \left[\binom{2}{1} \times p_{b,d}(1-p_{b,d})\right] \times \left[\binom{1}{1} \times p_{d,t}\right] = 0.64 \times 0.18 \times 0.9 = 0.10368$. This is shown in Figure 3(c).



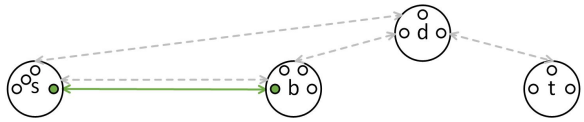
(a) Swap at $i = 1$ succeeds



(b) Swap at $i = 1$ fails



(c) Swap at $i = 2$ succeeds



(d) Swap at $i = 2$ fails

Figure 4. Possible results of Phase 2, assuming Phase 1 results in the case illustrated in Figure 3(c). (a) Swap at $i = 1$ (at node b) succeeds; (b) Swap at $i = 1$ (at node b) fails; (c) Swap at $i = 1$ (at node b) succeeds, then swap with $i = 2$ (at node d) succeeds; (d) Swap at $i = 1$ (at node b) succeeds, then swap with $i = 2$ (at node d) fails.

Now assume that Phase 1 resulted in the case shown in Figure 3(c). We perform Phase 2. For $i = 1$, we have $\tilde{c}_1 = 2$ and $\bar{c}_2 = 1$. Therefore, we perform $\min\{\tilde{c}_1,\bar{c}_2\} = 1$ entanglement swap at node b. If the swap is successful, we have an entanglement connecting node s to node d as

illustrated in Figure 4(a); otherwise, we have the situation as illustrated in Figure 4(b) and the algorithm stops. Now assume that the swap for $i = 1$ is successful. For $i = 2$, we have $\tilde{c}_2 = 1$ and $\bar{c}_3 = 1$. Therefore, we perform $\min\{\tilde{c}_2, \bar{c}_3\} = 1$ entanglement swap at node d. If the swap is successful, we have one entanglement connecting node s to node t as illustrated in Figure 4(c); otherwise, we have the situation as illustrated in Figure 4(d).

4. Path Metrics and Problem Formulation

In this section, we study various metrics of an ePath, including residual quantum memory, fidelity, entanglement probability distribution, and throughput. We also formulate the problems to be studied in this paper.

A. Fidelity and Residual Quantum Memory of an ePath

The Werner parameter and fidelity of an ePath $\pi_{u_0,u_h} = [(u_0,u_1),c_1]\text{-}[(u_1,u_2),c_2]\text{-}\cdots\text{-}[(u_{h-1},u_h),c_h]$ can be computed as

$$W(\pi_{u_0,u_h}) = \Pi_{i=1}^{h} W(e(u_{i-1},u_i)) = \Pi_{i=1}^{h} w_{u_{i-1},u_i}, \quad (4.1)$$

$$F(\pi_{u_0,u_h}) = \frac{1+3W(\pi_{u_0,u_h})}{4}. \quad (4.2)$$

The residual quantum memory of an ePath is the number of quantum memories at the destination node of the ePath that remain available, excluding those allocated to the ePath.

Definition 3 (Residual Quantum Memory): Let $\pi_{u_0,u_h} = [(u_0,u_1),c_1]\text{-}[(u_1,u_2),c_2]\text{-}\cdots\text{-}[(u_{h-1},u_h),c_h]$ be an ePath connecting u_0 to u_h. The *residual quantum memory* of π_{u_0,u_h} is

$$R(\pi_{u_0,u_h}) = Q_{u_h} - c_h. \quad (4.3)$$

As a technical convention, the residual quantum memory of an empty ePath π_{u_0,u_0} is Q_{u_0}. □

We illustrate the concepts of Werner parameter, fidelity, and residual quantum memory of an ePath with the aid of Figure 2. Note that the vertex attributes and edge attributes of the quantum network are given in Figure 1. For the ePath $\pi_1 = [(s,b),2]\text{-}[(b,d),2]\text{-}[(d,t),1]$ shown in Figure 2(a), the Werner parameter is $W(\pi_1) = W([(s,b),2]) \times W([(b,d),2]) \times W([(d,t),1]) = w_{s,b} \times w_{b,d} \times w_{d,t} = 0.9\times0.9\times0.8 = 0.648$; the fidelity is $F(\pi_1) = \frac{1+3\times W(\pi_1)}{4} = 0.736$; the residual quantum memory is $R(\pi_1) = Q_t - 1 = 2$. *Note that the residual quantum memory of an ePath is only concerned with the amount of quantum memory at the destination node still available after the allocation to the ePath. It is not concerned with the other nodes on the ePath.*

B. EPD and Throughput of an ePath

Let $\pi_{u_0,u_h} = [(u_0,u_1),c_1]\text{-}[(u_1,u_2),c_2]\text{-}\cdots\text{-}[(u_{h-1},u_h),c_h]$ be an ePath connecting node u_0 to node u_h. Each time we execute the protocol in Algorithm 1, the number of u_0-u_h entanglement generated may be different from other executions. Therefore, the result of the execution is a random variable. We call the expectation of this random variable the *throughput* of ePath π_{u_0,u_h}, and denote it by $T(\pi_{u_0,u_h})$.

Definition 4 (EPD of an ePath): Let $\pi_{u_0,u_h} = [(u_0,u_1),c_1]\text{-}[(u_1,u_2),c_2]\text{-}\cdots\text{-}[(u_{h-1},u_h),c_h]$ be an ePath connecting node u_0 to node u_h. Let $\mathbf{p}_k^{\pi_{u_0,u_h}}$ be the probability of having exactly k entanglement connecting u_0 and u_h when we execute the protocol in Algorithm 1, $k = 0,1,\ldots,C$. We call the $(C+1)$-dimensional vector $\mathbf{p}^{\pi_{u_0,u_h}} = (\mathbf{p}_0^{\pi_{u_0,u_h}}, \mathbf{p}_1^{\pi_{u_0,u_h}}, \ldots, \mathbf{p}_C^{\pi_{u_0,u_h}})$ the *entanglement probability distribution* (EPD) of π_{u_0,u_h}. □

Theorem 1: Let $h \geq 1$ be an integer. Let $\pi_{u_0,u_h} = [(u_0,u_1),c_1]\text{-}[(u_1,u_2),c_2]\text{-}\cdots\text{-}[(u_{h-1},u_h),c_h]$ be an h-hop ePath connecting u_0 to u_h. The EPD of π_{u_0,u_h} can be computed recursively as follows:

$$\mathbf{p}_k^{\pi_{u_0,u_1}} = \mathbf{p}_k^{[(u_0,u_1),c_1]}, \qquad k = 0,\ldots,C; \quad (4.4)$$

$$\mathbf{p}_k^{\pi_{u_0,u_i}} = \sum_{l=k}^{C}\sum_{r=k}^{C} \mathbf{p}_l^{\pi_{u_0,u_{i-1}}} \mathbf{p}_r^{[(u_{i-1},u_i),c_i]} \times \rho(l,r,k,q_{u_{i-1}}),$$
$$k = 0,\ldots,C; i = 2,\ldots,h, \quad (4.5)$$

where the vector $\mathbf{p}^{[(u_{i-1},u_i),c_i]}$ is computed by (3.1), and $\rho(l,r,k,q)$ is computed by (3.2). □

Proof. The protocol in Algorithm 1 tries c_i entanglement generations on edge (u_{i-1},u_i). Therefore $\mathbf{p}^{[(u_{i-1},u_i),c_i]}$ is computed by (3.1), for $i = 1,2,\ldots,h$.

Algorithm 1 performs entanglement swapping sequentially from left to right (at vertices $u_1,u_2,\ldots,u_{h-1}$, respectively). While performing entanglement swaps at vertex u_{i-1}, the probability of resulting in exactly k entanglement connecting u_0 and u_i is computed by (4.5), as the algorithm performs $\min\{l,r\}$ entanglement swaps at vertex u_{i-1} for $l = k,k+1,\ldots,C$ and $r = k,k+1,\ldots,C$. □

Definition 5 (Throughput of an ePath): Let π be an ePath, with EPD $\mathbf{p}^{\pi}$. We call

$$T(\pi) = \sum_{k=1}^{C} k \times \mathbf{p}_k^{\pi} \quad (4.6)$$

the throughput of π. □

Recall that $C = 3$ in our example network in Figure 1. By convention, the EPD of the empty ePath $[(s,s),0]$ at node s is $(0,0,0,1)$. The throughput, Werner parameter, and fidelity of this empty ePath are 3, 1, and 1, respectively. The EPD of the ePath $[(s,b),2]\text{-}[(b,d),2]\text{-}[(d,t),1]$ shown in Figure 2(a) is $(0.2694, 0.7306, 0, 0)$. The throughput, Werner parameter, and fidelity of this ePath are 0.7306, 0.6480 and 0.7360, respectively.

C. Single-pair ePath Problems

A fundamental problem in quantum networking is to find an ePath connecting a source node s to a destination node t whose fidelity is at least a given fidelity threshold and whose throughput is at least a given throughput threshold. We formally define this problem as follows.

Definition 6 (FnT-constrained ePath): Given a source node s and a destination node t in the quantum network

G, together with a fidelity constraint $\mathcal{F}$ and a throughput constraint $\mathcal{T}$, the *fidelity- and throughput-constrained ePath problem* asks for an s-t ePath $\pi_{s,t}$ with fidelity at least $\mathcal{F}$ and throughput at least $\mathcal{T}$.

An ePath has fidelity at least $\mathcal{F}$ if and only if its Werner parameter is at least $\frac{4\mathcal{F}-1}{3}$. Therefore, the fidelity constraint can be equivalently specified as a Werner parameter constraint.

The general FnT-constrained ePath problem is NP-hard, as stated in the following theorem. We leave the proof of this theorem in Appendix A for a smooth flow of the paper.

Theorem 2: The FnT-constrained ePath problem is NP-hard. □

The FnT-constrained ePath problem is a feasibility problem. One of its optimization problems is the following Fidelity-constrained Max-throughput ePath problem.

Definition 7 (F-constrained Max-throughput ePath): Given a source node s and a destination node t in the quantum network G, together with a fidelity constraint $\mathcal{F}$, the *fidelity-constrained maximum-throughput ePath problem* asks for an s-t ePath $\pi_{s,t}$ that has the maximum throughput among all s-t ePaths whose fidelity is at least $\mathcal{F}$. □

The above two problems are closely related. The fidelity- and throughput-constrained ePath problem is a feasibility problem. It asks for an s-t ePath that satisfies the fidelity and throughput constraints. The fidelity-constrained maximum-throughput ePath problem is an optimization problem. It asks for an s-t ePath that has the maximum throughput among all s-t ePaths satisfying the fidelity constraint. The FnT-constrained ePath problem is feasible if and only if the throughput of the maximum throughput s-t ePath subject to the fidelity constraint is at least as large as the throughput threshold. Therefore an algorithm for the F-constrained Max-throughput ePath problem solves both problems.

By properties of the Werner state, any ePath has a fidelity at least $\frac{1}{4}$. Therefore, when the fidelity constraint $\mathcal{F}$ is less than or equal to $\frac{1}{4}$, the fidelity-constrained maximum throughput ePath problem reduces to the (unconstrained) maximum-throughput ePath problem.

Symmetric to the Fidelity-constrained Max-throughput ePath problem, we can also study the throughput-constrained max-fidelity ePath problem. Our algorithm can be modified to solve the throughput-constrained max-fidelity ePath problem. Due to space limitations, we concentrate on the fidelity-constrained max-throughput problem in this paper.

For the quantum network in Figure 1, the ePath connecting s to t with maximum throughput is the ePath illustrated in Figure 2(a), with throughput 0.7306, Werner parameter 0.6480, and fidelity 0.7360. The maximum fidelity ePath connecting s to t has a fidelity 0.7900, and it is illustrated in Figure 2(b), with throughput 0.6451, Werner parameter 0.7200, and fidelity 0.7900. The maximum-throughput s-t ePath with fidelity at least 0.70 is the ePath in Figure 2(a). There is no ePath from s to t with fidelity at least 0.75 and throughput at least 0.70.

5. Extension and Domination of ePaths

In this section, we present the extension operation and dominance relation of ePaths. Given an s-y ePath $\pi_{s,y}$, and an edge-width pair $[(y,z),c]$, we can extend $\pi_{s,y}$ by $[(y,z),c]$ to obtain an s-z ePath $\pi_{s,z}$, provided that certain conditions are met. Path extension is an operation used in our algorithm. Path domination is an important binary relation on ePaths that is used by our algorithm to significantly reduce search space.

A. Extensions of an ePath

We can extend an ePath by appending an edge-width pair to the ePath, provided that constraints similar to (3.3)–(3.5) on quantum memory and quantum channels are satisfied. This is formally defined as follows.

Definition 8 (Extension of an ePath): Let $\pi_{u_0,u_h} = [(u_0,u_1),c_1]$-$[(u_1,u_2),c_2]$-$\cdots$-$[(u_{h-1},u_h),c_h]$ be an h-hop u_0-u_h ePath. Let $[(u_h,u_{h+1}),c_{h+1}]$ be an edge-width pair such that

$$c_{h+1} \le |\mathbb{C}_{u_h,u_{h+1}}|, \tag{5.1}$$

$$c_{h+1} \le R(\pi_{u_0,u_h}), \tag{5.2}$$

$$c_{h+1} \le Q_{u_{h+1}}. \tag{5.3}$$

The extension of π_{u_0,u_h} by the edge-width pair $[(u_h,u_{h+1}),c_{h+1}]$ is the $(h+1)$-hop u_0-u_{h+1} ePath $[(u_0,u_1),c_1]$-$[(u_1,u_2),c_2]$-$\cdots$-$[(u_{h-1},u_h),c_h]$-$[(u_h,u_{h+1}),c_{h+1}]$. The extension of π *by* $[(u,v),c]$ is denoted by $\pi \oplus [(u,v),c]$. □

We illustrate extensions of an ePath with the aid of Figure 5. Figure 5(a) corresponds to the empty ePath $\pi_0 = [(s,s),0]$ at node s. Figure 5(b) shows an extension of π_0 by the edge-width pair $[(s,b),2]$. The resulting ePath is $\pi_1 = \pi_0 \oplus [(s,b),2] = [(s,b),2]$. Figure 5(c) shows an extension of π_1 by the edge-width pair $[(b,d),2]$. The resulting ePath is $\pi_2 = \pi_1 \oplus [(b,d),2] = [(s,b),2]$-$[(b,d),2]$. Figure 5(d) shows an extension of π_2 by the edge-width pair $[(d,t),1]$. The resulting ePath is $\pi_3 = \pi_2 \oplus [(d,t),1] = [(s,b),2]$-$[(b,d),2]$-$[(d,t),1]$.

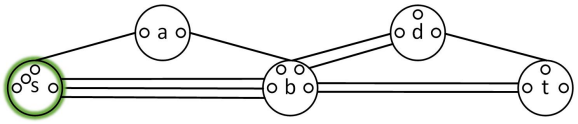
B. Path Domination

Let X_π denote the number of E2E entanglement generated across the ePath π by Algorithm 1. The throughput of π is $T(\pi) = \mathbb{E}[X_\pi]$ as defined in Definition 5.

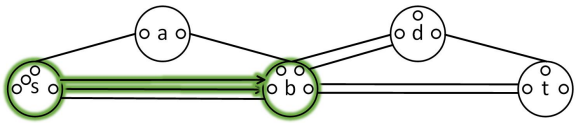
Definition 9 (Stochastic Ordering): Let $X, Y \in \mathbb{R}$ be random variables with arbitrary distributions. We say that X is stochastically larger than Y, denoted by $X \ge_{st} Y$ (or $Y \le_{st} X$), iff $\mathbb{E}[f(X)] \ge \mathbb{E}[f(Y)]$ for all increasing functions $f : \mathbb{R} \to \mathbb{R}$. □

Observe that this implies $P(X < x) \le P(Y < x), \forall x \in \mathbb{R}$. If X and Y have the same distributions, we write $X =_{st} Y$.

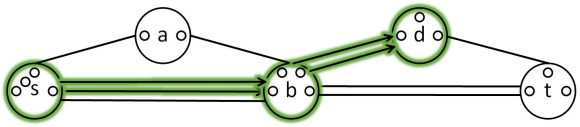
Let π_1 and π_2 be two ePaths. We say that π_1 is *stochastically stronger* than π_2, if $X_{\pi_1} \ge_{st} X_{\pi_2}$, and denote this by $\pi_1 \trianglerighteq \pi_2$ (or $\pi_2 \trianglelefteq \pi_1$).



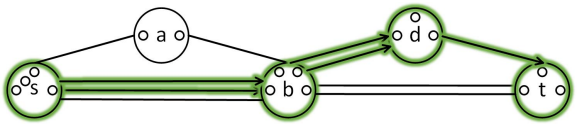
(a) The empty ePath at node s.



(b) Extension by edge (s, b) and width 2.



(c) Further extension by edge (b, d) and width 2.



(d) Further extension by edge (d, t) and width 1.

Figure 5. Illustration of extensions of an ePath. (a) The empty ePath at node s. (b) Extension of the ePath in (a) by $[(s, b), 2]$; (c) Extension of the ePath in (b) by $[(b, d), 2]$; (d) Extension of the ePath in (c) by $[(d, t), 1]$.

It follows from the above definition that $T(\pi_1) \geq T(\pi_2)$ whenever $\pi_1 \trianglerighteq \pi_2$.

Definition 10 (Path Domination): Let π_1 and π_2 be two ePaths connecting node u to node v. We say π_1 *dominates* π_2 (or π_2 *is dominated by* π_1) if $\pi_1 \trianglerighteq \pi_2$, $R(\pi_1) \geq R(\pi_2)$, and $W(\pi_1) \geq W(\pi_2)$. We use $\pi_1 \succeq \pi_2$ (or $\pi_2 \preceq \pi_1$) to denote the fact that π_1 dominates π_2. □

Theorem 3: Let π be an ePath connecting node x to node y. Let π' be the ePath obtained by extending π by the edge-width pair $[(y, z), c]$, i.e., $\pi' = \pi \oplus [(y, z), c]$. Then $\pi \trianglerighteq \pi'$.

Proof. The number of E2E entanglement over path π' is

$$\begin{aligned} X_{\pi'} &\leq \min\{X_\pi, c\} \\ &\leq X_\pi. \end{aligned}$$

Hence $X_{\pi'} \leq_{st} X_\pi$, which proves $\pi \trianglerighteq \pi'$. □

Theorem 4: Let π_1 and π_2 be two ePaths connecting node x to node y. We extend both ePaths by the same edge-width pair $[(y, z), c]$. If $\pi_1 \succeq \pi_2$, then $\pi_1 \oplus [(y, z), c] \succeq \pi_2 \oplus [(y, z), c]$. □

Proof. Let X_{π_1} and X_{π_2} denote the number of E2E entanglement created over ePaths π_1 and π_2 respectively. Note that $\pi_1 \succeq \pi_2$ implies $\pi_1 \trianglerighteq \pi_2$, which in turn implies $X_{\pi_1} \geq_{st} X_{\pi_2}$. By [22, Proposition 9.2] we can construct random variables X_1' and X_2' such that X_i' and X_{π_i} have the same distributions, $i = 1, 2$, and X_1' is greater than or equal to X_2', i.e., $X_1' \geq X_2'$. Let $\pi_i' = \pi_i \oplus [(y, z), c]$, $i = 1, 2$. Let Z_1 and Z_2 denote the number of swaps attempted when extending paths π_1 and π_2. They are given by $Z_i = \min\{X_i', c\}$, $i = 1, 2$, which satisfy $Z_1 \geq Z_2$. Let X_i'' denote the final number of E2E Bell pairs across extended ePath π_i, $i = 1, 2$; X_i'' is a binomial random variable with parameters q and Z_i. Here q is the probability of a successful swap. Since $Z_1 \geq Z_2$, it follows from [22, Problem 9.9] that $X_1'' \geq_{st} X_2''$ and, hence, $X_{\pi_1'} \geq_{st} X_{\pi_2'}$, which proves $\pi_1 \oplus [(y, z), c] \trianglerighteq \pi_2 \oplus [(y, z), c]$.

Since $W(\pi_1) \geq W(\pi_2)$, we have

$$\begin{aligned} W(\pi_1 \oplus [(y, z), c]) &= W(\pi_1) \times w_{y,z} && (5.4) \\ &\geq W(\pi_2) \times w_{y,z} && (5.5) \\ &= W(\pi_2 \oplus [(y, z), c]). && (5.6) \end{aligned}$$

Since $R(\pi_1 \oplus [(y, z), c]) = R(\pi_2 \oplus [(y, z), c])$, we have proved that $\pi_1 \oplus [(y, z), c] \succeq \pi_2 \oplus [(y, z), c]$. □

Let $\pi_1 = [(s, b), 1]$ and $\pi_2 = [(s, a), 1]$-$[(a, b), 1]$ be two ePaths connecting s to b (in the example quantum network in Figure 1). The EPD of π_1 is $(0.2000, 0.8000, 0, 0)$. The EPD of π_2 is $(0.4960, 0.5040, 0, 0)$. Hence π_1 is stochastically stronger than π_2. Since $R(\pi_1) = R(\pi_2) = 3$ and $W(\pi_1) = 0.9 > 0.81 = W(\pi_2)$, we conclude that π_1 dominates π_2. Note that the throughput of π_1 is larger than the throughput of π_2, $T(\pi_1) = 0.8000 > 0.5040 = T(\pi_2)$.

Suppose we extend both π_1 and π_2 by the edge-width pair $[(b, d), 1]$ to get $\pi_1' = \pi_1 \oplus [(b, d), 1] = [(s, b), 1]$-$[(b, d), 1]$ and $\pi_2' = \pi_2 \oplus [(b, d), 1] = [(s, a), 1]$-$[(a, b), 1]$-$[(b, d), 1]$. The EPD of π_1' is $(0.3520, 0.6480, 0, 0)$. The EPD of π_2' is $(0.5918, 0.4082, 0, 0)$. We observe that $\pi_1' \trianglerighteq \pi_2'$. We also observe that $\pi_1 \trianglerighteq \pi_1'$ and $\pi_2 \trianglerighteq \pi_2'$.

While the throughput of the ePath $\pi_3 = [(s, b), 2]$-$[(b, d), 2]$-$[(d, t), 1]$ shown in Figure 2(a) is larger than that of the ePath $\pi_4 = [(s, b), 2]$-$[(b, t), 2]$ shown in Figure 2(b), π_3 is not stochastically stronger than π_4. Hence neither of them dominates the other.

6. Computing Maximum Throughput ePath with Fidelity Constraint

In this section, we present an algorithm for computing an s-t ePath with maximum throughput subject to a fidelity constraint, as defined in Definition 7. Our definition of throughput follows the well-known *sequential swapping order* [6, 27] which is best suited for hop-by-hop path finding algorithms, as the EPD of $\pi \oplus [(u, v), c]$ can be computed using the EPD of π and the EPD of $[(u, v), c]$ (refer to Theorem 1) with the sequential swapping order.

A. Description of the Algorithm

Our algorithm creates and manages a set of objects of type `ePATH`. Among others, an `ePATH` object has the following fields: `T` is the throughput of the ePath; `W` is the Werner parameter of the ePath; `endV` is the end vertex of the ePath; `endC` is the width of the last edge-width pair; `p[0:C]` is the EPD of the ePath; `pred` is the predecessor of the ePath. All of the above are computed when an ePath is created. In addition, an `ePATH` object also has a field `active`, an ePath is active if and only if it is not dominated by any ePath computed by the algorithm.

There is an array `V[1:n]` of nodes. For each node i, $V[i]$→Q is the quantum memory at node i; $V[i]$→q is the success probability of entanglement swapping at node i; $V[i]$→`activePath` is a list of s-i ePaths that are still active; $V[i]$→`inactivePath` is a list of s-i ePaths that are no longer active; $V[i]$→`adj` is the adjacency list of node i. The algorithm uses a priority queue PQ, which contains pointers to objects of type `ePATH`, with `T` as the key. The algorithm is listed as Algorithm 2.

Lines 1-5 perform initialization of the attributes. The empty ePath is inserted to PQ. In the **while**-loop, the algorithm extracts an ePath π with the maximum throughput value from the priority queue. If π is an s-t ePath, it is an optimal solution, and the algorithm stops. Otherwise, if π is active (not dominated), we compute all feasible extensions (Werner parameter constraint not violated) of π, discard dominated ePaths, and insert the non-dominated extensions into PQ.

We use Figure 6 to illustrate the steps of Algorithm 2 while computing a max-throughput s-t ePath with Werner parameter at least 0.7 (fidelity at least 0.775) in the quantum network shown in Figure 1. Table II shows the ePaths computed in the computation of the optimal ePath from s to t. Each row of the table shows the ePath, its predecessor, throughput, Werner parameter, and residual quantum memory.

In Figure 6, we highlight an ePath being inserted into PQ in red (then black after its insertion); we highlight the ePath being extracted from PQ in blue (then purple after its extraction). In Table II, we highlight an ePath that is dominated by other ePaths (hence not inserted into PQ) in blue.

After initialization, the empty ePath $\pi_1 = [(s, s), 0]$ with EPD $= (0, 0, 0, 1)$ was inserted into both PQ and the active list at node s. At this moment, π_1 is the only ePath on PQ. This is shown in Figure 6(a). The algorithm then enters the while-loop in Line 6.

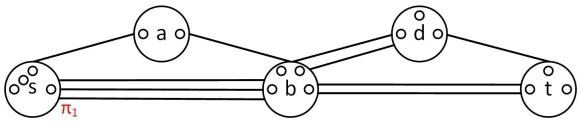
In Line 7, π_1 is extracted. Extending π_1 by $[(s, a), 1]$, we obtain the ePath $\pi_2 = [(s, a), 1]$ with EPD $= (0.3, 0.7, 0, 0)$. π_2 is inserted into PQ and the active list at node a. Extending π_1 by $[(s, b), 3]$, we obtain the ePath $\pi_3 = [(s, b), 3]$ with EPD $= (0.008, 0.096, 0.384, 0.512)$. π_3 is inserted into PQ and the active list at node b. Extending π_1 by $[(s, b), 2]$, we obtain the ePath $\pi_4 = [(s, b), 2]$ with EPD $= (0.04, 0.32, 0.64, 0)$. π_4 is inserted into PQ and the active list at node b. Extending π_1 by $[(s, b), 1]$, we obtain the ePath $\pi_5 = [(s, b), 1]$ with EPD $= (0.2, 0.8, 0, 0)$. π_5 is inserted into PQ and the active list at node b. This is shown in Figure 6(b).

The next ePath extracted from PQ is π_3. Extending π_3 by $[(b, a), 1]$, we obtain the ePath $\pi_6 = [(s, b), 3]$-$[(b, a), 1]$ with EPD $= (0.1965, 0.8035, 0, 0)$. π_6 is inserted into PQ and the active list at node a. Extending π_3 by $[(b, d), 1]$, we obtain the ePath $\pi_7 = [(s, b), 3]$-$[(b, d), 1]$ with EPD $= (0.1965, 0.8035, 0, 0)$. π_7 is inserted into PQ and the active list at node d. Extending π_3 by $[(b, t), 1]$, we obtain the ePath $\pi_8 = [(s, b), 3]$-$[(b, t), 1]$ with EPD $= (0.6429, 0.3571, 0, 0)$.

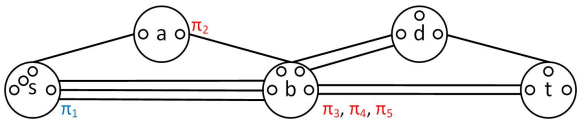
Algorithm 2: OPTePath$(n, C, G, s, t, \overline{W})$

Input: Network G given in its adjacency lists `V[1:n]`, source vertex s, destination vertex t.
Output: A max throughput s-t ePath with Werner parameter at least $\overline{W}$.

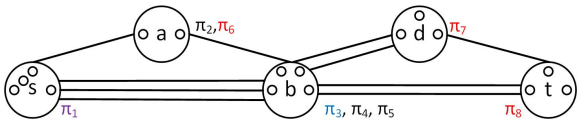
```
/* Preparation and Initialization:                          */
1  Create an empty priority queue PQ;
2  for ∀v ∈ {1, 2, ..., n} do
3      V[v]->activePath := NULL;
       V[v]->inactivePath := NULL;
4  V[s]->activePath contains a unique ePATH object
     with active = 1; endV = s; endC = 0;
     pred = NULL; W = 1; T = C; p[C] = 1.0; p[k] = 0.0
     for k = 0, 1, 2, ..., C − 1; RQM = V[s]->Q.
5  Insert the ePATH on V[s]->activePath into PQ.
/* Extraction and Extension:                                */
6  while PQ ≠ ∅ do
7      Path := ExtractMax(PQ); y := Path->endV;
8      if (Path->active == 0) continue;
9      if (y == t) goto 32; // optimal s-t ePath
10     if (Path->RQM == 0) then
11         Move Path from V[y]->activePath to
             V[y]->inactivePath;
12         continue; // Path cannot be extended
13     π_y := s to y path corresponding to Path;
14     for ∀edge ∈ V[y]->adj do
15         z := edge->endV; CH := edge->CH;
             p := edge->p; W := edge->W;
             Bound := min{Path->RQM, V[z]->Q, CH};
16         if (z ∈ π_y or W(Path)·W < W̄ or Bound ≤ 0)
               then continue // not extending
17         for ∀c = 1, 2, ..., Bound do
18             newPath := Path ⊕ [(y, z), c];
19             for ∀zPath ∈ V[z]->activePath do
20                 if (zPath ⪰ newPath) then
21                     newPath->active := 0;
                           // newPath is not
                           useful.
22                     Insert newPath into
                         V[z]->inactivePath;
23                     goto Line 30;
24             for ∀zPath ∈ V[z]->activePath do
25                 if (newPath ⪰ zPath) then
26                     zPath->active := 0; // zPath
                           is no longer useful.
27                     Move zPath from
                         V[z]->activePath to
                         V[z]->inactivePath;
28             newPath->active := 1;
29             Insert newPath into V[z]->activePath;
                 Insert newPath into PQ.
30             continue;
31 Exit: No s-t ePath with Werner parameter at least W̄;
     // Exit 1: no ePath
32 Exit: Path is a max throughput s-t ePath with Werner
     parameter at least W̄; // Exit 2: optimal
     ePath found
```



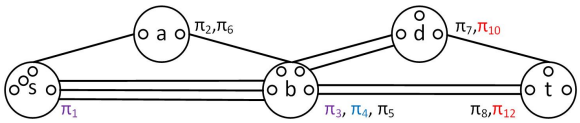
(a) ePath π_1 is the only ePath in PQ.



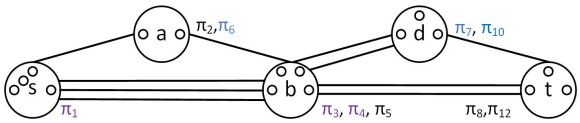
(b) π_1 is extracted; $\pi_2, \pi_3, \pi_4, \pi_5$ are created.



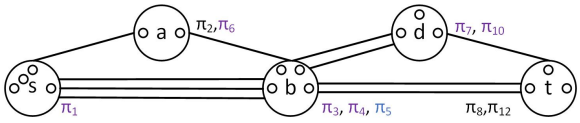
(c) π_3 is extracted; π_6, π_7, π_8 are created.



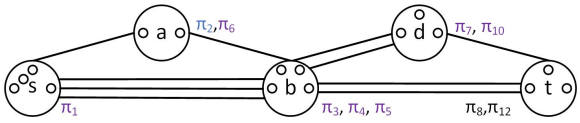
(d) π_4 is extracted; $\pi_9, \pi_{10}, \pi_{11}, \pi_{12}, \pi_{13}$ are created.



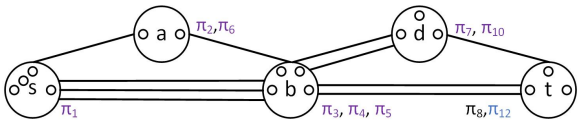
(e) π_{10}, π_7 and π_6 are extracted.



(f) π_5 is extracted; $\pi_{14}, \pi_{15}, \pi_{16}, \pi_{17}, \pi_{18}$ are created.



(g) π_2 is extracted; π_{19} is created.



(h) Optimal s-t ePath π_{12} is extracted; algorithm stops.

Figure 6. Walk-through example of Algorithm 2. A maximum-throughput s-t ePath with fidelity at least 0.775 (Werner parameter at least 0.7) is computed through hop-by-hop extensions.

Table II
EPATHS COMPUTED IN THE RUNNING EXAMPLE

ePath	Description	Pred	T	W	R
π_1	the empty ePath at node s	NULL	3.0000	1.0000	4
π_2	$[(s,a),1]$	π_1	0.7000	0.9000	1
π_3	$[(s,b),3]$	π_1	2.4000	0.9000	1
π_4	$[(s,b),2]$	π_1	1.6000	0.9000	2
π_5	$[(s,b),1]$	π_1	0.8000	0.9000	3
π_6	$[(s,b),3]$-$[(b,a),1]$	π_3	0.8035	0.8100	1
π_7	$[(s,b),3]$-$[(b,d),1]$	π_3	0.8035	0.8100	2
π_8	$[(s,b),3]$-$[(b,t),1]$	π_3	0.3571	0.7200	2
π_9	[(s, b), 2]-[(b, a), 1]	π_4	0.7776	0.8100	1
π_{10}	[(s, b), 2]-[(b, d), 2]	π_4	1.3219	0.8100	1
π_{11}	[(s, b), 2]-[(b, d), 1]	π_4	0.7776	0.8100	2
π_{12}	[(s, b), 2]-[(b, t), 2]	π_4	0.6451	0.7200	1
π_{13}	[(s, b), 2]-[(b, t), 1]	π_4	0.3456	0.7200	2
π_{14}	[(s, b), 1]-[(b, a), 1]	π_5	0.6480	0.8100	1
π_{15}	[(s, b), 1]-[(b, d), 2]	π_5	0.7128	0.8100	1
π_{16}	[(s, b), 1]-[(b, d), 1]	π_5	0.6480	0.8100	2
π_{17}	[(s, b), 1]-[(b, t), 2]	π_5	0.4608	0.7200	1
π_{18}	[(s, b), 1]-[(b, t), 1]	π_5	0.2880	0.7200	2
π_{19}	[(s, a), 1]-[(a, b), 1]	π_2	0.5040	0.8100	3

π_8 is inserted into PQ and the active list at node t. This is shown in Figure 6(c).

The next ePath extracted from PQ is π_4. Extending π_4 by $[(b,a),1]$, we obtain the ePath $\pi_9 = [(s,b),2]$-$[(b,a),1]$ with EPD $= (0.2224, 0.7776, 0, 0)$. This time we find $\pi_9 \preceq \pi_6$. Therefore π_9 is not inserted into PQ and the active list at a. Extending π_4 by $[(b,d),2]$, we obtain the ePath $\pi_{10} = [(s,b),2]$-$[(b,d),2]$ with EPD $= (0.0980, 0.4821, 0.4199, 0)$. Therefore, π_{10} is inserted into PQ and the active list at node d. Extending π_4 by $[(b,d),1]$, we obtain the ePath $\pi_{11} = [(s,b),2]$-$[(b,d),1]$ with EPD $= (0.2224, 0.7776, 0, 0)$. We find that $\pi_{11} \preceq \pi_7$. Therefore, π_{11} is not inserted into PQ and the active list at node d. Extending π_4 by $[(b,t),2]$, we obtain the ePath $\pi_{12} = [(s,b),2]$-$[(b,t),2]$ with EPD $= (0.43783, 0.47923, 0.08294, 0)$. π_{12} is inserted into PQ and the active list at node t. Extending π_4 by $[(b,t),1]$, we obtain the ePath $\pi_{13} = [(s,b),2]$-$[(b,t),1]$ with EPD $= (0.6544, 0.3456, 0, 0)$. We find that $\pi_{13} \preceq \pi_8$. Hence π_{13} is not inserted into PQ and the active list at node t. This is shown in Figure 6(d).

The next ePath extracted from PQ is π_{10}. The algorithm does not extend π_{10} by $[(d,t),1]$ because $W(\pi_{10}) \times w_{d,t} = 0.81 \times 0.8 = 0.648 < \overline{W} = 0.70$. The next ePath extracted from PQ is π_7. For the same reason, the algorithm does not extend π_7. The next ePath extracted from PQ is π_6. This ePath cannot be extended either. This is shown in Figure 6(e).

The next ePath to be extracted from PQ is π_5. Extending π_5 by $[(b,a),1]$, we obtain the ePath $\pi_{14} = [(s,b),1]$-$[(b,a),1]$ with EPD $= (0.3520, 0.6480, 0, 0)$. We find that $\pi_{14} \preceq \pi_6$. Hence π_{14} is not inserted in PQ and the active list at node a. Extending π_5 by $[(b,d),2]$, we obtain the ePath $\pi_{15} = [(s,b),1]$-$[(b,d),2]$ with EPD $= (0.2872, 0.7128, 0, 0)$. We find that $\pi_{15} \preceq \pi_{10}$. Hence π_{15} is not inserted into PQ and the active list at node d. Extending π_5 by $[(b,d),1]$, we obtain the ePath $\pi_{16} = [(s,b),1]$-$[(b,d),1]$ with EPD $= (0.3520, 0.6480, 0, 0)$. We

find that $\pi_{16} \preceq \pi_7$. Hence π_{16} is not inserted into PQ and the active list at node d. Extending π_5 by $[(b,t),2]$, we obtain the ePath $\pi_{17} = [(s,b),1]$-$[(b,t),2]$ with EPD $= (0.5392, 0.4608, 0, 0)$. We find that $\pi_{17} \preceq \pi_{12}$. Hence π_{17} is not inserted into PQ and the active list at node t. Extending π_5 by $[(b,t),1]$, we obtain the ePath $\pi_{18} = [(s,b),1]$-$[(b,t),1]$ with EPD $= (0.7120, 0.2880, 0, 0)$. We find that $\pi_{18} \preceq \pi_8$. Hence π_{18} is not inserted into PQ and the active list at node t. This is shown in Figure 6(f).

The next ePath to be extracted from PQ is π_2. Extending π_2 by $[(a,b),1]$, we obtain the ePath $\pi_{19} = [(s,a),1]$-$[(a,b),1]$ with $EPD = (0.4960, 0.5040, 0, 0)$. We find that $\pi_{19} \preceq \pi_5$. Hence π_{19} is not inserted into PQ and the active list at node b. This is shown in Figure 6(g).

The next ePath to be extracted from PQ is π_{12}. We find that $\pi_{12} = [(s,b),2]$-$[(b,t),2]$ is an s-t ePath. Therefore, in Line 9 the algorithm realizes that π_{12} is a max-throughput ePath from s to t with Werner parameter at least 0.70. The algorithm stops. This is shown in Figure 6(h).

B. Analysis of the Algorithm

In this section, we analyze the properties of Algorithm 2. First, we prove that the algorithm extracts ePaths in non-increasing order of the throughput. Then, we prove that the first s-t ePath extracted from the priority queue is the optimal solution.

Lemma 1 (Monotonicity): Let the ePaths extracted from PQ by Algorithm 2 be $\pi_1, \pi_2, \ldots, \pi_L$. Then

$$T(\pi_l) \geq T(\pi_r),\ 1 \leq l < r \leq L. \tag{6.1}$$

In other words, the algorithm extracts ePaths with non-increasing throughput. □

Proof. The ePath π_1 is the empty ePath $[(s,s),0]$, with throughput C. All other ePaths extracted are extensions of π_1. It follows from Theorem 3 that

$$T(\pi_1) \geq T(\pi_r), r = 2, 3, \ldots, L \tag{6.2}$$

This shows that (6.1) is true for $l = 1$.

Assume that (6.1) is true for $l = j$, where j is an integer such that $1 \leq j < L$. We prove that (6.1) is also true for $l = j + 1$.

At the time ePath π_{j+1} is extracted from PQ, for any $r > j + 1$, π_r is either on PQ or not inserted to PQ yet. If π_r is on PQ, we must have $T(\pi_r) \leq T(\pi_{j+1})$, because π_{j+1} has the largest throughput among all ePaths on PQ. If π_r is not on PQ, it must be an extension of an ePath whose throughput is no larger than $T(\pi_{j+1})$, hence has a throughput no larger than $T(\pi_{j+1})$ (by Theorem 3). Therefore (6.1) is also true for $l = j + 1$. By the principle of mathematical induction, we have proved the lemma. □

Theorem 5 (Optimality): If there is no s-t ePath with Werner parameter at least $\overline{W}$, Algorithm 2 correctly states so. If there is an s-t ePath with Werner parameter at least $\overline{W}$, Algorithm 2 finds a max-throughput s-t ePath with Werner parameter at least $\overline{W}$. □

Proof. If there is no s-t ePath with a Werner parameter at least $\overline{W}$, the algorithm must stop in Line 31. In the rest, we assume that there is an s-t ePath with Werner parameter at least $\overline{W}$.

Let $\pi_{u_0,u_H} = [(u_0,u_1),c_1]$-$[(u_1,u_2),c_2]$-$\cdots$-$[(u_{H-1},u_H),c_H]$ be a max throughput s-t ePath with Werner parameter at least $\overline{W}$, where $u_0 = s$ and $u_H = t$. Let $\pi_{u_0,u_h} = [(u_0,u_1),c_1]$ -$[(u_1,u_2),c_2]$-$\cdots$-$[(u_{h-1},u_h),c_h]$ be the h-th prefix of π_{u_0,u_H}, $h = 0, 1, 2, \ldots, H$, where the 0-th prefix of π_{u_0,u_H} is the empty ePath $([u_0,u_0],0)$. For simplicity, we denote π_{u_0,u_h} by π_h, i.e., $\pi_h = \pi_{u_0,u_h}$, $h = 0, 1, 2, \ldots, H$.

It follows from Theorem 3 that

$$T(\pi_0) \geq T(\pi_1) \geq \cdots \geq T(\pi_H). \tag{6.3}$$

We claim that *for each $h = 0, 1, 2, \ldots, H$, either π_h is inserted into PQ or a u_0-u_h ePath π'_h is inserted into PQ during the execution of Algorithm 2, where $\pi'_h \succeq \pi_h$.*

The claim is true for $h = 0$, since π_0 is the first ePath inserted into PQ.

Assume that the claim is true for $h = 0, 1, \ldots, j$, where $j < H$. We will prove that the claim is also true for $h = j+1$.

If π_j is extracted, then either π_{j+1} is entered into PQ (because the extension $\pi_{j+1} = \pi_j \oplus [(u_j,u_{j+1}),c_{j+1}]$ is considered by the algorithm) or a u_0-u_{j+1} ePath π'_{j+1} has already been inserted into PQ, where $\pi'_{j+1} \succeq \pi_{j+1}$.

If an u_0-u_j ePath π'_j that dominates π_j is extracted, then the extension $\pi'_j \oplus [(u_j,u_{j+1}),c_{j+1}]$ is considered by the algorithm. Note that $\pi'_j \oplus [(u_j,u_{j+1}),c_{j+1}] \succeq \pi_j \oplus [(u_j,u_{j+1}),c_{j+1}]$ because $\pi'_j \succeq \pi_j$. Therefore an ePath that dominates π_{j+1} must have been inserted into PQ. This proves that the claim is also true for $h = j + 1$. By the principle of mathematical induction, the claim is true.

Following the above claim, either π_H or an ePath π'_H such that $\pi'_H \succeq \pi_H$ is on PQ. Note that $\pi'_H \succeq \pi_H$ implies that π'_H is also a max-throughput u_0-u_H ePath with Werner parameter at least $\overline{W}$. Hence the algorithm finds an optimal solution when one exists. This proves the theorem. □

7. Performance Evaluation

To evaluate the performance of our proposed algorithm, we have conducted extensive evaluations. We measure the scalability of our algorithm and compare it with other algorithms in terms of solution quality and running time. All binaries are compiled with `g++ 13.1.0` and executed on an Ubuntu 20.04 server equipped with two Intel Xeon 6226R CPUs (32 cores, 2.9 GHz) and 512 GiB RAM.

A. Algorithms

We denote our algorithm by `OPT-D`, here `D` emphasizes that the algorithm uses ePath domination. To study the effectiveness of the ePath domination property, we also implemented a version of the algorithm without ePath domination (with Lines

19-27 removed), denoted by OPT−B. For comparison, we implemented a version of the Q-CAST algorithm [26], denoted by QCAST. Q-CAST is designed for multi-pair problems without fidelity constraint. We made a slight modification of Q-CAST to deal with fidelity constraint. Specifically, we introduced a Werner parameter to each edge, and added fidelity as a feasibility constraint: when an edge is relaxed in the path finding algorithm, we only update the candidate path at a node when the new path has a larger throughput without violating the fidelity constraint. We also implemented a greedy algorithm, denoted by GREEDY, which is a Dijkstra-like algorithm that extends highest-throughput partial paths that do not violate fidelity constraint. The difference between GREEDY and QCAST is that QCAST uses uniform edge width along a path while GREEDY uses non-uniform edge width along a path. The major difference between our algorithm and the baseline algorithms is that the baseline algorithms maintain a single entanglement path at each node. Our algorithm maintains a list of non-dominated entanglement paths at each node. This is key to why our algorithm is optimal (and requires more time). We did not compare with [38] because our focus is an optimal algorithm for single pair without entanglement purification, while [38] focuses on approximation algorithms for the multi-pair problem which also employs entanglement purification.

B. Experimental Setup

Topologies. We synthesize Waxman [32] graphs G_n with $n \in \{20, 40, 80, 160, 320, 640, 1280, 2560\}$ using parameters $\alpha = 0.4$ and $\beta = 0.15$; nodes are placed uniformly at random in a unit square, and each pair (u, v) is connected with probability $\beta \exp\big(-d(u,v)/\alpha L\big)$, where d is the Euclidean distance and L is the maximum pairwise distance in the current instance.
Physical parameters. Each node v has a quantum memory size $Q_v \sim \mathcal{U}\{1, \ldots, 10\}$, and edge (u, v) has $|\mathbb{C}_{u,v}| = \min(Q_u, Q_v)$ parallel channels. Werner parameter $w_{u,v}$, success probability of entanglement $p_{u,v}$, and success probability of entanglement swapping q_v are drawn i.i.d. from $\mathcal{U}[0, 1]$.
Fidelity constraints. The fidelity constraints are determined by the Werner parameter thresholds. We used $\overline{W} = 0, 0.25, 0.5, 0.75$ in our evaluations. Recall that $F = \frac{1+3W}{4}$. These four values of Werner parameters correspond to fidelity constraints of $\frac{1}{4}$, $\frac{7}{16}$, $\frac{5}{8}$, and $\frac{13}{16}$, respectively.
Workload and metrics. For every graph we evaluate up to $\min\big(10000,\ n(n-1)\big)$ E2E pairs chosen uniformly at random, at different fidelity thresholds $\overline{W}$. For each pair we record (i) wall-clock time t, and (ii) achieved throughput T. Reported values are arithmetic means over all experimented (s, t) pairs.

C. Results and Observations

Scalability of OPT−D. Figure 7 plots the average runtime of OPT−D, $t_{\text{OPT−D}}$ versus n for four fidelity thresholds $\overline{W} \in \{0, 0.25, 0.5, 0.75\}$. Note that $\overline{W} = 0$ corresponds to throughput maximization without fidelity constraint. The runtime grows roughly linearly with n: from 4 ms at $n = 20$ to 2.74 s at $n = 2560$. At $n = 2560$, tightening $\overline{W}$ decreases the average runtime significantly. For $\overline{W} = 0, 0.25, 0.5$, and 0.75, the average runtime is 2.73 s, 0.32 s, 18.8 ms, and 0.94 ms, respectively, all but the unconstrained case under one second.

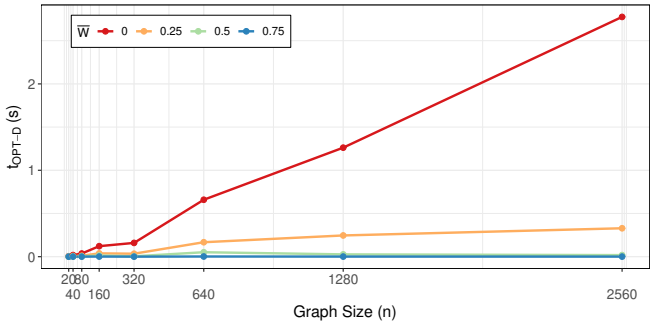


Figure 7. Average completion time over up to $\min(10000, n(n-1))$ randomly selected ordered (s, t) pairs for each n and W, using OPT−D.

We observe that *the running time of our algorithm empirically scales well with n, for n as large as* 2560. We attribute this to the effectiveness of the path domination property, to be elaborated later in this section. Another observation is that the running time decreases as the fidelity constraint tightens. This is because the search space reduces as the fidelity constraint tightens.

Throughput vs fidelity. Figure 8 shows how the throughput from node 1 to node 2560 (blue) and the average throughput over $10,000$ random E2E pairs (red) degrade as $\overline{W}$ increases, for a quantum network with 2560 nodes. One can prove that the s-t throughput is non-increasing as the fidelity constraint tightens up. The same can be said for the average throughput over multiple E2E pairs. We observe that our result matches the theory.

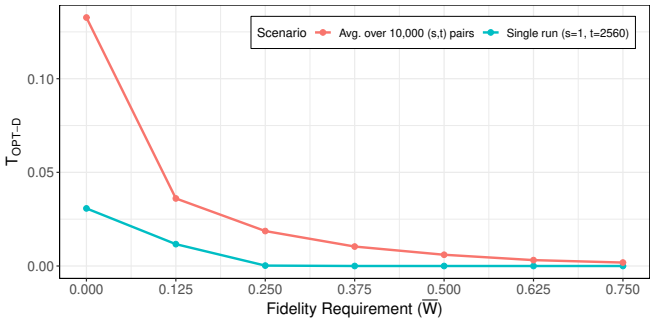


Figure 8. (Orange) Average $\text{T}_{\text{OPT−D}}$ of $10,000$ randomly selected (s, t) pairs in different $\overline{W}$, $n = 2560$. (Blue) Single run with $s = 1, t = 2560, n = 2560$.

Effect of path domination pruning. We found that both OPT−B and OPT−D achieve the same optimal throughput, while OPT−B takes longer than OPT−D.

Table III shows the difference in running time for computing a maximum throughput ePath from node 5 to node 3 in the quantum network with 20 nodes, for different fidelity constraints. As can be seen in the table, for $\overline{W} = 0$, OPT−D

$\overline{W}$	Algorithm	Runtime (s)	ePaths
0.000	OPT−B	4353.7	1.566e7
	OPT−D	0.01541	699
0.250	OPT−B	0.09124	4279
	OPT−D	0.00465	225
0.500	OPT−B	0.00773	479
	OPT−D	0.00160	105
0.750	OPT−B	0.00088	29
	OPT−D	0.00037	23

Table III
COMPARISON OF TWO ALGORITHMS WITH $n = 20, s = 5, t = 3$, AND $\overline{W} = \{0, 0.25, 0.5, 0.75\}$.

took 0.01541 seconds and used 699 ePath objects, while OPT−B took 4353.7 seconds and used $1.566e7$ ePath objects. Clearly, the running time of OPT−B can be exponential in the worst case. This extreme case magnifies the effectiveness of the pruning strategy in OPT−D.

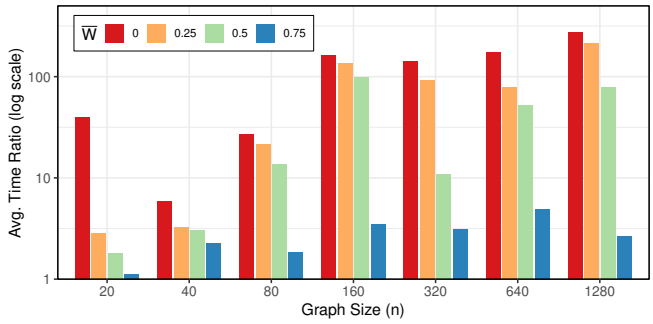


Figure 9. Average time ratio of $\min(10000, n(n-1))$ randomly selected (s, t) pairs in graphs with $n = \{20, 40, 80, 160, 320, 640, 1280\}$, $\overline{W} = 0, 0.25, 0.5, 0.75$ using OPT−B algorithm over OPT−D.

To study the effectiveness of the pruning strategy without excessive use of time and memory, we compute the average of $\min(1000, \frac{t_{\text{OPT-B}}}{t_{\text{OPT-D}}})$ over $\min(10000, n(n-1))$ runs for $n = 20, 40, 80, 160, 320, 640, 1280$ and $\overline{W} \in \{0, 0.25, 0.5, 0.75\}$. Figure 9 reports the mean on a $\log_{10}$ y-axis. Disabling pruning incurs a 10-200× penalty at $\overline{W} = 0$ and remains greater than 5× even at $\overline{W} = 0.5$. Note that we limit OPT−B's approximate run time to be 1000 times that of OPT−D for logistical reasons, as some test cases would consume more than 512 GB of RAM available. Therefore, the average time ratio presented in Figure 9 is a lower bound of the true ratio.

Solution quality vs other algorithms. Our algorithm OPT−D is the first optimal algorithm for solving the general problem studied in this paper. To gain some insight into the improvement of our algorithm over existing algorithms (not guaranteeing an optimal solution), we compare OPT−D with Greedy and QCAST.

Figure 10 compares $\frac{\text{T}_{\text{Greedy}}}{\text{T}_{\text{OPT-D}}}$ and $\frac{\text{T}_{\text{QCAST}}}{\text{T}_{\text{OPT-D}}}$ on the largest topology ($n = 2560$). Ratios are averaged over pairs for which OPT−D finds positive throughput and excludes infeasible pairs. We observe that OPT−D outperforms both QCAST and Greedy.

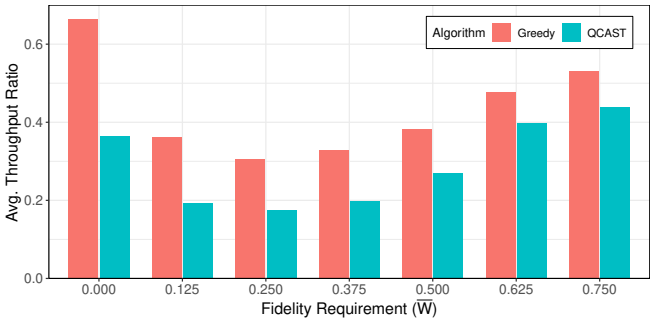


Figure 10. Average ratio of throughput over 10,000 randomly selected pairs using Greedy and QCAST over OPT−D in different $\overline{W}$, $n = 2560$.

8. CONCLUSIONS AND FUTURE RESEARCH

In this paper, we presented an extensive study of the problem of computing an E2E entanglement path subject to both fidelity and throughput constraints. In our problem, the fidelity and entanglement generation success probability may vary from link to link. Also, the entanglement swapping success probability may vary from node to node. The number of available channels on links and the number of available quantum memories at nodes are also taken into consideration. We prove that this problem is NP-hard. We design an algorithm for computing an E2E entanglement path with maximum throughput subject to fidelity constraint. To achieve this goal, we develop and prove the path domination theory which makes our algorithm very effective. We have restricted our attention of the sequential swapping order, where we proved optimality. Other swapping orders may lead to higher throughput of the ePath computed. However, computing the optimal swapping order of an ePath is challenging. Designing algorithms to compute an ePath with maximum throughput under other swapping orders remains open.

The integration of the proposed path computation into a deployed quantum network can be achieved by maintaining a dynamic global network state. Once established, these paths remain in place for extended duration. Also, in addition to optical links and nodes that comprise the entanglement paths, a conventional network is utilized for supporting path computations and also conventional communications needed for entanglement swapping.

In the current paper, we have not considered the impact of qubit decoherence [12]. A natural approach is to factor this into the fidelity constraint. We are working on this in a future paper. We have concentrated on single path routing for single pair in this paper. It can be shown that higher E2E throughput can be achieved via multi-path routing. Extending the techniques developed here to the case of multi-pair and multi-path routing remains an open research problem.

Appendix A: PROOF OF THEOREM 2

We prove that the FnT-constrained ePath problem is NP-hard by a reduction from the 2-*additive metrics path problem* (2APP), which is known to be NP-hard [31].

An instance $\mathcal{I}_1$ of the 2APP problem is given by an undirected graph $G = (V, E, \alpha, \beta)$, a source node $s \in V$, and destination node $t \in V$, and two positive numbers $\overline{\alpha} > 0$ and $\overline{\beta} > 0$. Here for each edge $e \in E$, $\alpha(e) \geq 0$ and $\beta(e) \geq 0$ are two edge attributes. The problem asks for an s-t path π in G such that $\sum_{e\in\pi} \alpha(e) \leq \overline{\alpha}$ and $\sum_{e\in\pi} \beta(e) \leq \overline{\beta}$.

We construct an instance $\mathcal{I}_2$ of the FnT-constrained ePath problem as follows. We set $\mathcal{T} = \exp(-\overline{\alpha})$ and $\mathcal{F} = \frac{1+3\times\exp(-\overline{\beta})}{4}$. For each vertex $v \in V$, we set $Q_v = 2$ and $q_v = 1$. For each edge $(u, v) \in E$, we set

$$|\mathbb{C}_{u,v}| = 1, \tag{A.1}$$

$$p(u, v) = \exp(-\alpha(u, v)), \tag{A.2}$$

$$w(u, v) = \exp(-\beta(u, v)). \tag{A.3}$$

Assume that u_0–u_1–$\cdots$–u_h is an s-t path in G such that $\sum_{i=1}^{h} \alpha(u_{i-1}, u_i) \leq \overline{\alpha}$ and $\sum_{i=1}^{h} \beta(u_{i-1}, u_i) \leq \overline{\beta}$. Then $[(u_0, u_1), 1]$–$[(u_1, u_2), 1]$–$\cdots$–$(u_{h-1}, u_h), 1]$ is a feasible solution to $\mathcal{I}_2$.

Assume that $[(u_0, u_1), c_1]$–$\cdots$–$(u_{h-1}, u_h), c_h]$ is a feasible solution to $\mathcal{I}_2$. Since $\mathcal{T} > 0$, we must have $c_i \geq 1$ for all $i = 1, 2, \ldots, h$. Since $|\mathbb{C}_{u,v}| = 1$ for each $(u, v) \in E$, we must have $c_i \leq 1$ for all $i = 1, 2, \ldots, h$. The throughput of $[(u_0, u_1), c_1]$–$\cdots$–$(u_{h-1}, u_h), c_h]$ is at least $\mathcal{T}$ and the fidelity of $[(u_0, u_1), c_1]$–$\cdots$–$(u_{h-1}, u_h), c_h]$ is at least $\mathcal{F}$ implies that u_0–u_1–$\cdots$–u_h is a feasible solution to $\mathcal{I}_1$. Since the 2APP problem is NP-hard [31], we have proved that the FnT-constrained ePath problem is also NP-hard. □

Appendix B: Differentiation with Flow-based Models

Chakraborty et al. [5] studied a closely related problem. Here we use an example to demonstrate that the problem studied in [5] is *different* from the problem studied in this paper.

Reference [5] assumes that all links have the same fidelity. Under this assumption, together with the assumption that the swapping success probability is a constant across all nodes, the paper was able to transform the fidelity constraint to hop-count constraint, leading to a polynomial time algorithm. In our paper, we do not enforce these constraints.

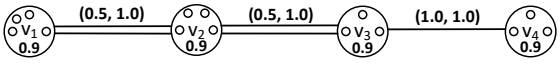


Figure 11. Quantum network A.

Figure 11 shows a quantum network with four vertices and three links. The links (v_1, v_2) and (v_2, v_3) both have two available channels, each with entanglement generation success probability 0.5 and fidelity 1.0. Link (v_3, v_4) has one available channel, with entanglement generation success probability 1.0 and fidelity 1.0. Nodes v_1 and v_2 each have 4 quantum memories. Nodes v_3 and v_4 each have 3 quantum memories. The entanglement swapping success probability for all nodes is 0.9. We want to compute the maximum throughput between v_1 and v_4. The v_1-v_4 ePath with maximum throughput is $[(v_1, v_2), 2]$–$[(v_2, v_3), 2]$–$[(v_3, v_4), 1]$, with EPD $(0.539312, 0.460688, 0)$ and throughput 0.460688. The model in [5] computes the capacity of a link e with entanglement generating success probability p_e by $c_e = \mu_s(1-(1-p_e)^m)$, where $m \geq 1$ is a positive integer and $\mu_s > 0$ is the entanglement generation rate. Hence the capacities of the links $(v_1, v_2), (v_2, v_3), (v_3, v_4)$ are $2(1 - 0.5^m)\mu_s, 2(1 - 0.5^m)\mu_s$, and μ_s, respectively. Using the calculations of [5], the maximum v_1-v_4 throughput is $0.9\times0.9\times\min\{2(1-0.5^m)\mu_s, 2(1-0.5^m)\mu_s, \mu_s\} = 0.81\mu_s$ for any $m \geq 1$. For $\mu_s = 1$, the maximum v_1-v_4 throughput computed by [5] is 0.81 which is **larger than** the maximum v_1-v_4 throughput computed in this paper.

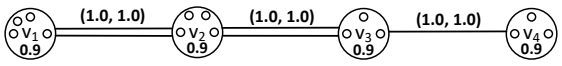


Figure 12. Quantum network B.

Let us modify the quantum network in Figure 11 by changing the entanglement generation success probability of the links (v_1, v_2) and (v_2, v_3) from 0.5 to 1.0. This is illustrated in Figure 12. Our paper finds the v_1-v_4 ePath with maximum throughput to be $[(v_1, v_2), 2]$–$[(v_2, v_3), 2]$–$[(v_3, v_4), 1]$, with EPD $(0.109, 0.891, 0)$ and throughput 0.891. The model in [5] computes the maximum throughput from v_1 to v_4 to be $0.81\mu_s$, for any $m \geq 1$. For $\mu_s = 1$, the maximum v_1-v_4 throughput computed using the model in [5] is 0.81, which is **smaller than** the maximum v_1-v_4 throughput computed in this paper.

The above examples show that the model in [5] and the model in this paper are different.

There are many papers that use flow-based models [10, 15, 37]. These flow-based models are also different from the model in our paper. For the network in Figure 11, the model in [37] computes a capacity of $\sqrt{0.9}, 0.9, \sqrt{0.9}$ for the links $(v_1, v_2), (v_2, v_3)$, and (v_3, v_4), respectively, leading to a v_1-v_4 throughput of 0.9. For the network in Figure 12, the model in [37] computes a capacity of $2 \times \sqrt{0.9}, 2 \times 0.9, \sqrt{0.9}$ for the links $(v_1, v_2), (v_2, v_3)$, and (v_3, v_4), respectively, leading to a v_1-v_4 throughput of $\sqrt{0.9}$. In both cases, the model in [37] computes an upper-bound on the throughput computed in our model. The same is true for many flow-based papers.

References

[1] R. K. Ahuja, T. L. Magnati, and J. B. Orlin. *Network Flows*. Prentice Hall, Englewood Cliffs, NJ, 1993.

[2] K. Azuma, S. E. Economou, D. Elkouss, P. Hilaire, L. Jiang, H.-K. Lo, and I. Tzitrin. Quantum repeaters: From quantum networks to the quantum internet. *Reviews of Modern Physics*, 95(4), Dec. 2023.

[3] M. Caleffi. Optimal routing for quantum networks. *IEEE Access*, 5:22299–22312, 2017.

[4] K. Chakraborty, A. Dahlberg, F. Rozpedek, and S. Wehner. Distributed Routing in a Quantum Internet. In *APS March Meeting Abstracts*, volume 2019 of *APS Meeting Abstracts*, page L28.005, Jan. 2019.

[5] 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*, 1:1–21, 2020.

[6] A. Chang and G. Xue. Order matters: On the impact of swapping order on an entanglement path in a quantum network. In *NetSciQCom2022*, pages 1–6, 2022.

[7] L. Chen, K. Xue, J. Li, Z. Li, R. Li, N. Yu, Q. Sun, and J. Lu. Redp: Reliable entanglement distribution protocol design for large-scale quantum networks. *IEEE JSAC*, 42(7):1723–1737, 2024.

[8] B. C. Ciobanu, T. A. Calafeteanu, A. B. Popa, R. Tătăroiu, and P. G. Popescu. Optimal entanglement distribution within a multi-ring topology. *Scientific Reports*, 15(1):16459, May 2025.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT Press, 2022.

[10] W. Dai, T. Peng, and M. Z. Win. Optimal remote entanglement distribution. *IEEE Journal on Selected Areas in Communications*, 38(3):540–556, 2020.

[11] S. Das, S. Khatri, and J. P. Dowling. Robust quantum network architectures and topologies for entanglement distribution. *Phys. Rev. A*, 97:012335, Jan 2018.

[12] M. G. de Andrade, E. A. Van Milligen, L. Bacciottini, A. Chandra, S. Pouryousef, N. K. Panigrahy, G. Vardoyan, and D. Towsley. On the analysis of quantum repeater chains with sequential swaps. *arXiv preprint arXiv:2405.18252*, 2024.

[13] L.-M. Duan and C. Monroe. Colloquium: Quantum networks with trapped ions. *Rev. Mod. Phys.*, 82:1209–1224, Apr 2010.

[14] W. Dür, M. Hein, J. I. Cirac, and H.-J. Briegel. Standard forms of noisy quantum operations via depolarization. *Phys. Rev. A*, 72:052326, Nov 2005.

[15] H. Gu, Z. Li, R. Yu, X. Wang, F. Zhou, J. Liu, and G. Xue. Fendi: Toward high-fidelity entanglement distribution in the quantum internet. *IEEE/ACM Trans. Netw.*, 2024.

[16] A. G. Iñesta, G. Vardoyan, L. Scavuzzo, and S. Wehner. Optimal entanglement distribution policies in homogeneous repeater chains with cutoffs. *npj Quantum Inf.*, 9(1), 05 2023.

[17] F. P. Kelly. Mathematical modelling of the Internet. In *Proc. Int. Congress on Industrial and applied mathematics*. Frank Kelly, 1999. http://www.statslab.cam.ac.uk/ frank/mmi.html.

[18] W. J. Munro, K. Azuma, K. Tamaki, and K. Nemoto. Inside quantum repeaters. *IEEE Journal of Selected Topics in Quantum Electronics*, 21(3):78–90, 2015.

[19] 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*, 5(1):1–9, 2019.

[20] S. Pirandola, R. Laurenza, C. Ottaviani, and L. Banchi. Fundamental limits of repeaterless quantum communications. *Nature Communications*, 8(15043), 2017.

[21] N. S. Rao. Probabilistic quickest path algorithm. *Theoretical Computer Science*, 312(2-3):189–201, 2004.

[22] S. M. Ross. *Stochastic Processes*. Wiley, 1996.

[23] M. Ruf, N. H. Wan, H. Choi, D. Englund, and R. Hanson. Quantum networks based on color centers in diamond. *J. Appl. Phys.*, 130(7):070901, 08 2021.

[24] N. Sangouard, C. Simon, H. de Riedmatten, and N. Gisin. Quantum repeaters based on atomic ensembles and linear optics. *Rev. Mod. Phys.*, 83:33–80, Mar 2011.

[25] E. Schoute, L. Mancinska, T. Islam, I. Kerenidis, and S. Wehner. Shortcuts to quantum network routing. *arXiv preprint arXiv:1610.05238*, 2016.

[26] S. Shi and C. Qian. Concurrent entanglement routing for quantum networks: Model and designs. In *Proc. ACM SIGCOMM*, pages 62–75, 2020.

[27] R. Van Meter. *Quantum Networking*. John Wiley & Sons, 2014.

[28] R. Van Meter, R. Satoh, N. Benchasattabuse, K. Teramoto, T. Matsuo, M. Hajdušek, T. Satoh, S. Nagayama, and S. Suzuki. A quantum internet architecture. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 341–352. IEEE, 2022.

[29] R. Van Meter, T. Satoh, T. D. Ladd, W. J. Munro, and K. Nemoto. Path selection for quantum repeater networks. *Networking Science*, 3(1-4):82–95, 2013.

[30] G. Vardoyan and S. Wehner. Quantum network utility maximization. *arXiv:2210.08135*, 2022.

[31] Z. Wang and J. Crowcroft. QoS routing for supporting resource reservation. *IEEE JSAC*, 14:1228–1234, 1996.

[32] B. M. Waxman. Routing of multipoint connections. *IEEE JSAC*, 6(9):1617–1622, 1988.

[33] S. Wehner, D. Elkouss, and R. Hanson. Quantum internet: A vision for the road ahead. *Science*, 362(6412), 2018.

[34] M. M. Wilde. *Quantum Information Theory*. Cambridge University Press, 2nd edition, 2017.

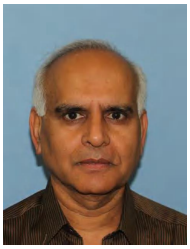
[35] G. Xue, W. Zhang, J. Tang, and K. Thulasiraman. Polynomial time approximation algorithms for multi-constrained qos routing. *IEEE/ACM Trans. Netw.*, 16(3):656–669, 2008.

[36] Q. Zhang, N. Di Cicco, M. Ibrahimi, R. C. Almeida, A. Gatto, R. Boutaba, and M. Tornatore. Link configuration for fidelity-constrained entanglement routing in quantum networks. In *IEEE INFOCOM 2025*, pages 1–10. IEEE, 2025.

[37] Y. Zhao and C. Qiao. Redundant entanglement provisioning and selection for throughput maximization in quantum networks. In *IEEE INFOCOM 2021*, pages 1–10. IEEE, 2021.

[38] Y. Zhao, G. Zhao, and C. Qiao. E2e fidelity aware routing and purification for throughput maximization in quantum networks. In *IEEE INFOCOM 2022*, pages 480–489. IEEE, 2022.

Guoliang Xue (Member 1996, Senior Member 1999, Fellow, 2011) is a Professor of Computer Science at Arizona State University. His research interests span the areas of wireless networks, quantum networking, security and privacy, and optimization. He received the IEEE Communications Society William R. Bennett Prize in 2019. He has served as VP-Conferences of the IEEE Communications Society, and on the editorial boards of many IEEE/ACM journals. He served as the Steering Committee Chair of IEEE INFOCOM from 2020 to 2025.

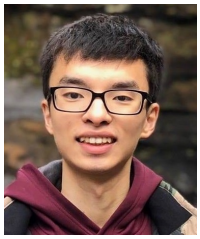


Nageswara S. V. Rao is a Corporate Fellow at Oak Ridge National Laboratory, where he joined in 1993. He received B. Tech from National Institute of Technology, Warangal, India, M.E. from School of Automation, Indian Institute of Science, Bangalore, India, and PhD in computer science from Louisiana State University. His research areas include high performance and quantum networking, information fusion, machine learning, and federations of science instruments. He is a Fellow of IEEE, and received 2005 IEEE Computer Society Technical Achievement Award and 2014 R&D 100 Award.

Don Towsley is currently a Distinguished Professor at the University of Massachusetts, Amherst. He received a B.A. in Physics and a Ph.D. in Computer Science from University of Texas. His research focuses on quantum communications and networking, and distributed quantum sensing and computing. He is a Fellow of AAAS, ACM and IEEE and has awards including the 2007 IEEE Koji Kobayashi Award, the 2023 Network Science Society Euler Award, and several test-of-time awards and conference best paper awards.

Gayane Vardoyan is an Assistant Professor at the Manning College of Information and Computer Sciences, University of Massachusetts, Amherst. From 2022 to 2024, she was an Assistant Professor at QuTech's Quantum Internet Division, and EEMCS, TU Delft. She received her PhD in computer science from the University of Massachusetts, Amherst, and her BS in Electrical Engineering and Computer Sciences from the University of California at Berkeley. Previously, she worked at the Argonne National Lab and the Computation Institute at the University of Chicago. Her research interests focus on the modeling and performance analysis of distributed quantum systems.

Zunzheng Zhang (Graduate Student Member) received the B.S. and M.S. degrees in Electronic Engineering from Nanjing University in 2021 and 2024. He is currently pursuing the Ph.D. degree in computer science at Arizona State University. His research interests include network optimization and quantum networks.



Xuanli Lin (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in computer science from Arizona State University in 2018 and 2020, respectively, where he is currently pursuing the Ph.D. degree in computer science. His research interests include network optimization, resilient offloading networks, and the Internet of Things.

Muneer Alshowkan received the Ph.D. degree in Computer Science and Engineering from the University of Bridgeport, Bridgeport, CT, USA, in 2017, and the M.S. degree in Information, Network, and Computer Security from the New York Institute of Technology, New York, NY, USA, in 2011. He joined Oak Ridge National Laboratory (ORNL), Oak Ridge, TN, USA, in 2020 as a Postdoctoral Research Associate and is currently a Research Scientist. His research interests include computer science and engineering, quantum networking, and quantum communications. He is a recipient of the R&D 100 Award.

Joseph M. Lukens received the B.S. degree in electrical engineering and physics in 2011 from the University of Alabama, Tuscaloosa, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, Indiana, in 2015. Employed as a Wigner Fellow and Research Scientist at Oak Ridge National Laboratory (ORNL) from 2015–2022 and then as Senior Director of Quantum Networking at Arizona State University (2022–2024), he joined Purdue University in January 2025 as an associate professor, where he maintains a joint faculty appointment at ORNL. His research interests encompass a variety of topics in photonic quantum information processing, optical networking, and Bayesian inference.

Nicholas A. Peters (Senior Member 2017) is the Section Head for Quantum Information Science and a Distinguished Scientist at Oak Ridge National Laboratory, which he joined in 2015. Before that he worked for the Telcordia Technologies family of companies. He received the B.A. degree summa cum laude in physics and mathematics from Hillsdale College, in 2000 and the M.S. and Ph.D. degrees in physics from The University of Illinois Urbana-Champaign, in 2002 and 2006, respectively. His research spans photonic quantum computing, sensing, and networking.

Saikat Guha received the B.Tech. degree in Electrical Engineering from IIT Kanpur, India, in 2002, and the S.M. and Ph.D. degrees in EECS from MIT in 2004 and 2008, respectively. He is Clark Distinguished Chair Professor of Electrical and Computer Engineering at the University of Maryland, College Park, and serves as Director of the National Science Foundation Center for Quantum Networks. His research interests include investigating fundamental quantum limits on optics-based information processing with applications to communications, imaging and computation, and with specific emphasis to structured realizations of optical systems that can approach those performance limits. He was a co-recipient of the 2011 and 2016 Excellence in Engineering and Technology (EiET) Award, Raytheon Company's highest technical honor. Saikat is an IEEE Fellow (class of 2025).