# From Entanglement Purification Scheduling to Fidelity-constrained Entanglement Routing

Ziyue Jia, Lin Chen
School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, 510006, China
Guangdong Provincial Key Laboratory of Information Security Technology, Guangzhou 510006, China
Email: jiazy5@mail2.sysu.edu.cn, chenlin69@mail.sysu.edu.cn

***Abstract*—Recently emerged as a disruptive networking paradigm, quantum networks rely on the mysterious quantum entanglement to teleport qubits without physically transferring quantum particles. However, the state of quantum systems is extremely fragile due to environment noise. A promising technique to combat against quantum decoherence is entanglement purification. To fully exploit its benefit, two fundamental research questions need to be answered: (1) given an entanglement path, what is the optimal entanglement purification schedule? (2) how to compute min-cost end-to-end entanglement paths subject to fidelity constraint? In this paper, we give algorithmic solutions to both questions. For the first question, we develop an optimal entanglement purification scheduling algorithm for the single-hop case and analyze the PURIFY-AND-SWAP strategy in the multi-hop case by establishing the closed-form condition for its optimality. For the second question, we design a polynomial-time algorithm constructing an $\epsilon$-optimal fidelity-constrained path. The effectiveness of our algorithms are also numerically demonstrated by extensive simulations.**

## I. Introduction

Four decades ago in 1982, Richard Feynman postulated in his visionary article, *Simulating Physics with Computers* [15], that to simulate quantum systems one would need to build quantum computers, marking the birth of quantum computing. Today, quantum computing turns 40, we are still far from seeing Feynman's dream come true. For example, solving many fundamental problems in physics and chemistry requires hundreds of thousands or millions of qubits' computation power in order to correct errors arising from noise [3]. Any single quantum computer today cannot fulfill such tasks. We hence need to interconnect multiple quantum computers to scale up the number of qubits, thus forming a *quantum network*, or more ambitiously, a *quantum Internet* [35].

A quantum network interconnects quantum devices by exploiting fundamental quantum mechanical phenomena such as superposition, entanglement, and quantum measurement to exchange quantum information in the form of qubits, thereby achieve processing capabilities beyond what is possible with classical computer systems and networks [35]. A quantum network can form a virtual quantum machine of a large number of qubits, scaling with the number of interconnected devices.

This work is supported in part by National Science Foundation of China under Grant 62172455, Pearl River Talent Program under Grant 2019QN01X140, and Guangdong Provincial Key Laboratory of Information Security Technology (No. 2023B1212060026). L. Chen is the corresponding author.

979-8-3503-5171-2/24/$31.00 ©2024 IEEE

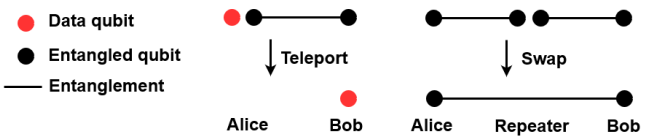


Fig. 1. Quantum teleportation and entanglement swapping

This, in turn, leads to an exponential boost of computing power with just a linear amount of the physical resources, i.e., the number of connected quantum devices. Therefore, a quantum network can support many ground-breaking applications lying beyond the capability of its classical counterparts, such as quantum communication, clock synchronization, secure remote computation, and distributed consensus.

However, designing quantum networks is a technically challenging task that we have never encountered in computer science and engineering. This is because quantum networks follow the laws of quantum mechanics, such as no-cloning, quantum measurement, quantum entanglement and teleportation [23]. These quantum phenomena have no counterpart in classical networks, and pose highly non-trivial constraints on the network design and optimization. Therefore, designing and optimizing quantum networks cannot be achieved by extrapolating the classical models to their quantum analogues, but requires a fundamental paradigm shift.

Technically, the most fundamental characteristic of quantum networks, compared to classical networks, is the famous and mysterious *quantum entanglement*, allowing to realize quantum teleportation so as to transmit a qubit without physically transferring the particle storing it, as illustrated in Figure 1.

However, distributed qubits inevitably interact with the environment, causing *decoherence*. Consequently, the initially maximally entangled quantum state will decohere into a mixed state, thus degrading or even paralyzing the corresponding quantum communication session. To combat against decoherence, there are three solutions: quantum error correction [5], entanglement purification [13], and quantum communication based on decoherence-free subspace [12]. Currently, entanglement purification appears to be the most convenient and deployable solution that has already been demonstrated both theoretically and experimentally [24].

Entanglement purification, also termed as entanglement

distillation, is to extract from a pool of low-fidelity[1] Einstein-Podolsky-Rosen (EPR) pairs a high-fidelity EPR pair[2]. Though seemingly intuitive to understand, exploiting entanglement purification to establish high-fidelity end-to-end entanglement paths is by no means a trivial task due to the following three technical challenges.

**Entanglement purification scheduling**. In order to produce a high-fidelity EPR pair, we may need to perform multiple rounds of entanglement purification. Figure 2 illustrates two entanglement purification scheduling policies (SYMMETRIC and PUMPING) resulting in different final fidelity. Their purification success probabilities are highlighted in red font. A natural question is to find the optimal entanglement purification scheduling policy.

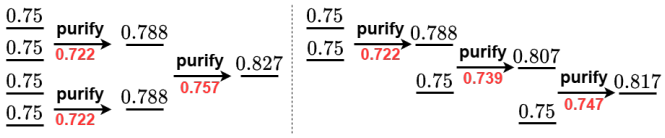


Fig. 2. We dispose 4 EPR pairs of fidelity 0.75. The left and right subfigures depict 2 purification scheduling policies producing 2 final EPR pairs of different fidelities. Their purification success probabilities are highlighted in red font.

**Interaction between entanglement purification and entanglement swapping**. Suppose that Alice and Bob are connected by a quantum repeater Charlie, with whom they have both established an EPR pair, as depicted in Figure 3. We have two different ways to produce a final EPR pair between Alice and Bob. The first one, termed as PURIFY-AND-SWAP, consists of performing entanglement purification between Alice and Charlie, and Bob and Charlie, and then entanglement swapping to stitch the two purified EPR pairs. The second one, termed as SWAP-AND-PURIFY, consists of performing entanglement swapping at Charlie to produce two EPR pairs between Alice and Bob, and then entanglement purification to get a final purified EPR pair. In fact, the above two schemes represent two extremities regarding the interaction between entanglement purification and entanglement swapping. The strategy space increases exponentially as the quantum network scales. Clearly, we face the problem of finding the optimal joint entanglement purification scheduling and swapping strategy.

**Fidelity-constrained entanglement path optimization**. At the routing level, we face the problem of finding an optimal entanglement path between Alice and Bob satisfying end-to-end fidelity constraint. By optimal, we mean to minimize the cost of the path, e.g., in terms of path delay, the total number of consumed EPR pairs. Our path optimization problem differs from its classical peer problems in that the number of qubits at each quantum nodes is limited, hence limiting the number of EPR pairs it can build with its neighbors. This non-standard

[1]In quantum information theory, *fidelity* is the probability that a set of qubits are actually in the state we believe that they ought to be in.

[2]cf. Section II for more detailed presentation

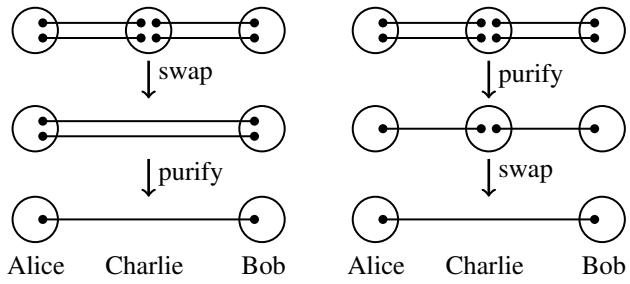


Fig. 3. SWAP-AND-PURIFY VS. PURIFY-AND-SWAP

constraint poses non-trivial technical challenges and thus calls for new algorithmic techniques that cannot build on existing path optimization solutions.

Driven by the above non-classic technical challenges, we embark in this paper to build a comprehensive algorithmic framework on entanglement purification scheduling and routing. Our main contributions are articulated as follows.

- **Entanglement purification scheduling**. Given a pool of elementary EPR pairs, we develop an optimal entanglement purification scheduling algorithm of polynomial-time complexity. For the multi-hop case along an entanglement path, we establish the close-form condition under which PURIFY-AND-SWAP is optimal.
- **Fidelity-constrained entanglement path optimization**. we design a polynomial-time algorithm constructing an $\epsilon$-optimal fidelity-constrained path.

We expect that our results obtained in this paper constitutes a small but systematic step towards building non-classical entanglement provisioning and routing algorithms to support high-fidelity end-to-end quantum information transfer by exploiting the limited quantum resources in an optimal and cost-effective manner in order to fully realize the unrivallable capabilities offered by quantum networks.

**Roadmap.** The paper is organized as follows. In Section II, we formulate and analyze the problem of entanglement purification scheduling. In Section III, we further analyze the problem of fidelity-constrained entanglement path optimization. Section IV presents simulation results. We review related work in Section V. Section VI concludes the paper.

## II. ENTANGLEMENT PURIFICATION SCHEDULING

As mentioned in the Introduction, the state of quantum systems is extremely fragile due to environment noise. Errors result in continuous degradation of our knowledge about the state of the quantum system. A widely applied quantitative metric about the quality of quantum states is *fidelity*. Ranging from 0 to 1, fidelity is essentially the probability of a qubit or a set of qubits being in the state we believe that they ought to be in. To improve the fidelity of an EPR pair, *entanglement purification* is usually applied. It consumes lower-fidelity EPR fairs to obtain a higher-fidelity one, as illustrated in Figure 4.[3]

[3]Generically, purification is the process of improving our knowledge of the state by testing propositions. This improvement is reflected as an increase in the fidelity in the density matrix representing our knowledge about the state.

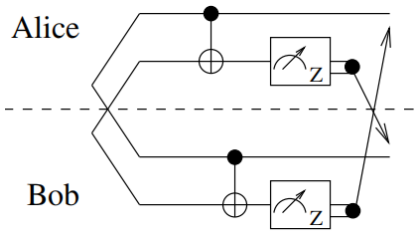


Fig. 4. Illustration of entanglement purification [21]. Alice and Bob each holds one half of two EPR pairs, the first pair to be purified and the second the pair to be sacrificed. The second pair is measured. The arrows indicate classical message exchange of the measurement results. If the measurement results are the same, i.e., 00 or 11, the purification is successful.

### A. Technical Background on Entanglement Purification

We consider the generic scenario where entanglement purification takes as input two imperfect EPR pairs, both in the Werner state [36]:

$$\rho_i = \frac{1-f_i}{3} I_4 + \frac{4f_i - 1}{3} |\Phi^+\rangle\langle\Phi^+| \quad i = 1, 2,$$

where $f_i$ denotes the fidelity of EPR pair $i$. Two CNOT operations are performed between the qubits at the same end of the two EPR pairs held by Alice and Bob, followed by a measurement on the two qubits of the second EPR pair in the computational basis. If the particles are found in the same state, the first EPR pair is kept, otherwise it is discarded. The fidelity of the purified EPR pair, denoted by $P(f_1, f_2)$, and the purification success probability, denoted by $F(f_1, f_2)$, can be computed in (1) and (2), respectively [22]. If multiple rounds of entanglement purification are performed, we need to turn the purified EPR pair to the Werner state by executing a depolarization operation.

$$\begin{aligned} F(f_1, f_2) &= \frac{f_1 f_2 + \frac{1}{9}(1-f_1)(1-f_2)}{f_1 f_2 + \frac{1}{3}(f_1 + f_2 - 2f_1 f_2) + \frac{5}{9}(1-f_1)(1-f_2)} \\ &= \frac{10 f_1 f_2 - f_1 - f_2 + 1}{8 f_1 f_2 - 2f_1 - 2f_2 + 5} \qquad (1) \\ P(f_1, f_2) &= f_1 f_2 + \frac{1}{3}(f_1 + f_2 - 2f_1 f_2) + \frac{5}{9}(1-f_1)(1-f_2) \\ &= \frac{8}{9} f_1 f_2 - \frac{2}{9}(f_1 + f_2) + \frac{5}{9} \qquad (2) \end{aligned}$$

To fully exploit the benefit of entanglement purification under limited quantum resources, two critical research questions should be carefully addressed.

**Q1**: How to schedule entanglement purification? We have demonstrated in Introduction that different entanglement purification scheduling policies may produce purified EPR pairs of different fidelity. Moreover, entanglement purification and swapping should be carefully orchestrated so as to optimize the overall performance.

**Q2**: How to integrate entanglement purification into entanglement provisioning and routing so as to build min-cost end-to-end entanglements with given QoS requirements in terms of fidelity and end-to-end entanglement throughput? We address this problem in the next section.

### B. Single-hop Entanglement Purification Scheduling

We start by solving the one-hop entanglement purification scheduling problem defined below.

**Problem 1.** *Alice and Bob is connected by a quantum link $e$, over which they have established $N$ elementary EPR pairs of fidelity $f_e$. We seek an entanglement purification scheduling policy producing the maximal number of EPR pairs whose fidelity is at least $f_\theta$.*

To get more insight into the purification scheduling problem, we can model a purification schedule as a tree, which we term as purification tree, as exemplified in Figure 5. We denote an entanglement mapping to an internal node resulting from a sub-tree with $k-1$ leaves as a level-$k$ entanglement. An elementary entanglement is a level-1 entanglement which is also a leaf. Therefore, the problem of purification scheduling maps to constructing the purification tree leading to maximal number of purified entanglements, subject to the fidelity constraint $f_\theta$. Clearly, enumerating all the possible trees is inefficient because there are an exponential number of possible trees, given that the number of non-isomorphic trees for a given number of nodes is the well-known Cayley number which scales exponentially w.r.t. the tree size. Therefore, we need a dedicated design, which is the central task of our work.

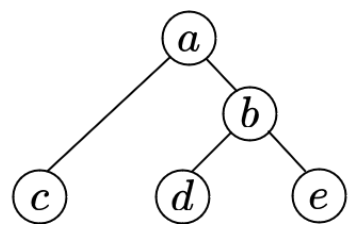


Fig. 5. Example of a purification tree

**Data structure**. We create a list $\mathcal{L}$ to store the candidate purification trees. Each entry $l$ of $\mathcal{L}$ is a quadruple representing a purification tree or sub-tree. Specifically, in the quadruple $l \triangleq (b_l, \hat{f}_l, \hat{\xi}_l, T_l)$:

- $b_l$ denotes the number of leaves in the tree;
- $\hat{f}_l$ denotes the discretized fidelity of the root of the tree;
- $\xi_l$ denotes the expected percentage of the EPR pairs of fidelity $f_l$ generated in the purification scheduling process. In other words, there are in average $\xi_l N$ EPR pairs of fidelity $f_l$ produced in the purification scheduling process. $\hat{\xi}_l$ denotes the discretized value approximating $\xi_l$, as detailed in our algorithm.
- $T_l$ records the entire purification tree or sub-tree $l$.

Given two entries $l_i = (b_i, \hat{f}_i, \hat{\xi}_i, T_i)$, $i = 1, 2$, we say that $l_1$ *dominates* $l_2$ if $b_1 \le b_2$, $\hat{f}_1 \ge \hat{f}_2$ and $\hat{\xi}_1 \ge \hat{\xi}_2$. The dominance is strict if at least one inequality holds strictly. Intuitively, an optimal purification tree is not strictly dominated by any other tree. So we can safely remove all the dominated entries in $\mathcal{L}$.

**Algorithm**. Our algorithm is depicted in Algorithm 1 consisting of two major parts.

We first compute the lower-bound on the number of leaves of a purification tree such that any purification tree reaching this bound generates EPR pairs whose fidelity is at least $f_\theta$. To that end, we define $\Gamma(i)$, $1 \le i \le N$, as the minimal fidelity achievable from $i$ EPR pairs of fidelity $f_e$. The boundary condition is given as $\Gamma(1) = f_e$. We can compute $\Gamma(i)$ by dynamic programming as below.

$$\Gamma(i) = \min_{1 \le k \le \lfloor i/2 \rfloor} F(\Gamma(k), \Gamma(i-k)).$$

It then follows that $N' = \arg\min_i \Gamma(i) \ge f_\theta$ is the lower-bound we are looking for.

The core part of our algorithm is the second **for** loop computing $\mathcal{L}$. At a high level, by executing the loop, our algorithm builds and searches the purification trees from the leaves. As we will never purify any EPR pair whose fidelity already reaches $f_\theta$, the maximal number of leaves of the purification tree we search is upper-bounded by $2(N'-1)$, with the worst case being the situation where two EPR pairs, each representing a purification tree of $N'-1$ leaves are purified to produce an EPR pair of sufficient fidelity. In each iteration of the second for loop, we scan each pair of entries $l_1, l_2 \in \mathcal{L}$ and build a new tree $l_3$ with $l_1$ and $l_2$ being its children[4]. If $l_3$ is not dominated by any entry in $\mathcal{L}$, we add $l_3$ to $\mathcal{L}$. Noticing the discretization process to compute $\hat{\xi}_3$ and $\hat{f}$, for each pair of $(b, \hat{\xi}, \hat{f})$, there are at most $\frac{1}{\Delta\xi \cdot \Delta f}$ entry in $\mathcal{L}$ as we remove all the dominated entries. Therefore, there are at most $\frac{n}{\Delta\xi \cdot \Delta f}$ entries in $\mathcal{L}$, where $n = \min\{N, 2(N'-1)\}$. Each entry containing a purification tree, the overall space complexity of our algorithm sums up to $O\left(\frac{n}{\Delta\xi \cdot \Delta f}\right)$. On the other hand, in each iteration, we grow the tree by merging two sub-trees, each mapping to an entry in $\mathcal{L}$. The overall time complexity sums up to $O\left(\frac{n^3}{\Delta\xi^2 \Delta f^2}\right)$.

It remains to analyze the optimality of Algorithm 1. The main source of efficiency loss comes from discretization which may accumulate along the purification tree from its leaves up to root. Theorem 1 establishes that, with sufficient discretization granularity, Algorithm 1 outputs a solution approaching optimum. Due to the page limit, the detailed proof is shown in appendix of [17].

**Theorem 1.** *Let $T'$ denote the tree output by our algorithm and $l' = (b', \hat{f}', \hat{\xi}', T')$ denote the corresponding entry in $\mathcal{L}$. Under the condition $\Delta\xi \le b'\xi'\epsilon$ and $\Delta f \le b' f' \epsilon$, Algorithm 1 outputs an $\epsilon-$optimal solution of Problem 1.*

It follows from Theorem 1 that $\Delta\xi$ depends on $\xi'$ and $b'$, which poses a problem, as we need to set $\Delta\xi$ before knowing $\xi'$ and $b'$. A practical way to set $\Delta\xi$ is to pick a purification schedule, e.g., PARALLEL, denoted by $l_0 \triangleq (b_0, f_0, \xi_0, T_0)$ and set $\Delta\xi = b_0\xi_0\epsilon$. As our algorithm achieves quasi-optimality, it holds that $\xi' b' \simeq \hat{\xi}' b' \ge \xi_0 b_0$. Hence we have $\Delta\xi = b_0\xi_0\epsilon < b'\xi'\epsilon$. $\Delta f$ can be set similarly to satisfy the condition in Theorem 1.

[4]By slightly abusing terminology, we use the entry also to denote the purification tree corresponding to it.

**Algorithm 1:** Entanglement purification scheduling

**Input:** $N$, $f_e$, $f_\theta$
**Output:** a purification tree

**Initialization**: $\Gamma(i) \leftarrow f_e$, $1 \le i \le N$, $\mathcal{L} \leftarrow \{(1, f_e, 1, T_1)\}$
**for** $i = 2$ ***to*** $N$, $k = 1$ ***to*** $\lfloor i/2 \rfloor$ **do**
  **if** $\Gamma(i) < F(\Gamma(k), \Gamma(i-k))$ **then**
    $\Gamma(i) \leftarrow F(\Gamma(k), \Gamma(i-k))$
$N' \leftarrow \arg\min_i \Gamma(i) \ge f_\theta$
**for** $i = 1$ ***to*** $\min\{N, 2(N'-1)\}$ **do**
  **foreach** *pair of entries* $l_i = (b_i, \hat{f}_i, \hat{\xi}_i, T_i) \in \mathcal{L}$, $i = \{1, 2\}$, *satisfying* $b_1 + b_2 \le \min\{N, 2(N'-1)\}$ **do**
    $\hat{f} \leftarrow \left\lceil \frac{F(f_1, f_2)}{\Delta f} \right\rceil \Delta f$
    $\hat{\xi}_3 \leftarrow \left\lceil \frac{P(f_1, f_2) \min\{\hat{\xi}_1, \hat{\xi}_2\}}{\Delta\xi} \right\rceil \Delta\xi$
    create a new tree $T_3$ with the two children being $T_1$ and $T_2$
    $l_3 \leftarrow (b_1 + b_2, \hat{f}, \hat{\xi}_3, T_3)$
    **if** $l_3$ *is not dominated by any entry in* $\mathcal{L}$ **then**
      add $l_3$ to $\mathcal{L}$
**if** *there exists at least an entry in* $\mathcal{L}$ *whose fidelity is at least* $f_\theta$ **then**
  let $l' \triangleq (b', \hat{f}', \hat{\xi}', T')$ denote the entry with maximal value of $\hat{\xi}'/b'$ among them
  **return** the purification tree $T'$
**else**
  **return** no feasible purification schedule

**Remark 1.** *It is insightful to compare our results with the existing works [9] on the simple bit flip error model, where*

$$F(f_1, f_2) = \frac{f_1 f_2}{f_1 f_2 + (1-f_1)(1-f_2)}.$$

*In this case, by algebraically rewriting the above equation as*

$$1 - \frac{1}{F(f_1, f_2)} = \left(1 - \frac{1}{f_1}\right) \cdot \left(1 - \frac{1}{f_2}\right),$$

*we can check that the entanglement purification map is associative. Hence, any entanglement purification schedule will output a purified EPR pair of the same fidelity. In contrast, we consider the generic Pauli channels, where the need for twirling makes the map* (1) *non-associative [14], calling for optimal design of entanglement purification scheduling policy.*

### C. Multi-hop Entanglement Purification Scheduling

We tackle the more challenging multi-hop situation with Alice and Bob separated by quantum repeaters and rely on entanglement swapping to establish end-to-end EPR pair. Entanglement purification scheduling should be jointly optimized with entanglement swapping scheduling, formulated below.

**Problem 2.** *Given that Alice and Bob are connected by an entanglement path $P$, we seek an optimal joint entanglement purification and swapping scheduling policy maximizing the fidelity of the final EPR pair between them.*

To solve Problem 2, we investigate a scheduling policy called PURIFY-AND-SWAP, which consists of firstly running Algorithm 1 to produce a purified elementary EPR pair of maximal fidelity at each hop and then performing entanglement swapping at each quantum repeater to stitch the purified elementary EPR pairs into an end-to-end one.

Structurally, PURIFY-AND-SWAP can be decomposed to two phases consisting of exclusively entanglement purification and entanglement swapping. Therefore, it is practically easy to deploy and tractable to analyze. In contrast, there are more complex policies where entanglement purification and swapping are intertwined in a nested way, rendering the deployment relatively complex and the performance analysis highly intractable. However, despite the neatness of PURIFY-AND-SWAP, its optimality cannot be guaranteed in general. Motivated by this observation, we investigate the following question: under what conditions are PURIFY-AND-SWAP optimal? Our main results in this subsection is the closed-form conditions to ensure its optimality.

Denote $P$ the entanglement path between Alice and Bob. Suppose we have established an EPR pair of fidelity $f_e$ over each link $e \in P$. We can compute the fidelity of the end-to-end EPR pair formed via entanglement swapping along $P$ as [13]

$$f(P) = \frac{1}{4}\left(1 + 3\prod_{e\in P}\frac{4f_e - 1}{3}\right). \tag{3}$$

Mathematically, it is more convenient to write (3) as

$$\frac{4f(P) - 1}{3} = \prod_{e\in P}\frac{4f_e - 1}{3}.$$

We first present an auxiliary lemma, whose proof, due to the page limit, consists of algebraic operations and is detailed in appendix of [17].

**Lemma 1.** *Consider* 3 *quantum nodes as shown in Figure 3, where there are two EPR pairs between Alice and Charlie of fidelity $f_1$ and $f_2$, and two EPR pairs between and Bob and Charlie of fidelity $f_3$ and $f_4$.*

- *If $f_1, f_2 \geq 0.5$ and $f_3, f_4 \geq 0.7$, then* PURIFY-AND-SWAP *outperforms* SWAP-AND-PURIFY *in terms of fidelity.*
- *If $f_i \geq 0.7$ for $1 \leq i \leq 4$ and the swapping success probability is upper-bounded by* $0.818$*, then* PURIFY-AND-SWAP *outperforms* SWAP-AND-PURIFY *in success probability, i.e., the probability of establishing an end-to-end EPR pair between Alice and Bob.*

**Remark 2.** *We make the following clarification:*

- *Regarding the fidelity, the condition is relatively tight such that $f_1 = 0.5$, $f_2 = 1$, $f_3 = 0.699$, $f_4 = 1$ forms a counter-example that* SWAP-AND-PURIFY *outperforms* PURIFY-AND-SWAP *by* $0.00004$ *in fidelity. When the condition does not hold, we report numerically by scanning $f_i \in [0.5, 0.7]$ with stepsize* $0.001$ *that* PURIFY-AND-SWAP *outperforms* SWAP-AND-PURIFY *in* $100\%$ *cases.*
- *Regarding swapping success probability, due to the limitation of current BSM scheme with linear optics, this probability does not exceed* $0.5$*, i.e., far less than* $0.818$ *[1], validating the condition empirically. As in the first case, we trace the success probability with $f_i \in [0.5, 0.7]$ with stepsize* $0.001$ *and report that* PURIFY-AND-SWAP *always outperforms* SWAP-AND-PURIFY.

Armed with Lemma 1, we next state the main result of this subsection. The proof consists of showing that if PURIFY-AND-SWAP is outperformed by another policy, then we can construct a sub-instance of the system where SWAP-AND-PURIFY outperforms PURIFY-AND-SWAP, which leads to a contradiction with Lemma 1. Due to the page limit, the detailed proof is provided in the appendix of [17].

**Theorem 2.** *If the fidelity of elementary EPR pairs is at least* $0.7$ *for each link of $P$ and the swapping success probability does not exceed 0.818, then* PURIFY-AND-SWAP *is optimal.*

## III. Fidelity-constrained Entanglement Routing

We have addressed the first research question regarding entanglement scheduling along a predetermined entanglement path. We proceed to the second question on how to integrate entanglement purification into entanglement provisioning and routing so as to build min-cost end-to-end entanglement paths with given QoS requirements.

We model the quantum network as an undirected graph $G \triangleq (V, E)$, where $V$ denotes the set of quantum nodes, i.e., quantum terminals and quantum repeaters, $E$ denotes the set of quantum channels. We note that a quantum channel between a pair of nodes $u$ and $v$ induces a communication channel if and only if $u$ and $v$ successfully establish a quantum entanglement. Each node $v$ disposes $Q_v$ qubits, allowing it to create at most $Q_v$ quantum entanglements with its neighbors. The aforementioned quantum network model is essentially the same as the one in research [10], [31], but we added considerations for fidelity, purification and swapping. Let $p_v$ denote the entanglement swap success probability of node $v$. As the qubits are limited resources, quantum entanglements need to be carefully provisioned to support communication sessions among quantum nodes.

Consider a communication session between Alice and Bob requesting $q_0$ qubits of fidelity at least $f_0$. Let $\mathcal{P}$ denote the set of entanglement paths between them. Given an edge $e \in P \in \mathcal{P}$, we define a non-decreasing cost function $C_e(m_e)$, where $m_e$ denotes the number of elementary EPR pairs over $e$ that are used to build the end-to-end EPR pair along $P$. The cost of path $P$ is defined as $C(P) \triangleq \sum_{e\in P} C_e(m_e)$. Below we give two concrete examples.

- It is often meaningful to minimize the number of consumed EPR pairs to achieve a given end-to-end fidelity. In

this case, we can simply set $C_e = m_e$ or, in the weighted case, $C_e = w_e m_e$, with $w_e$ being the associated weight.

- When multiple EPR pairs are involved over an edge, entanglement purification needs to be executed in multiple rounds. In this case, $C_e(m_e)$ can model the delay to obtain a purified EPR pair over $e$. $C(P)$ is thus the entire delay to establish the end-to-end EPR pair.

We call an entanglement path *feasible* if the following conditions are satisfied.

- The corresponding end-to-end EPR pairs are constructed by PURIFY-AND-SWAP, where entanglement purification follows Algorithm 1, subject to the qubit resource limit of each quantum node;
- The expected number of successfully established end-to-end EPR pairs is at least $q_0$;
- The fidelity of the end-to-end EPR pairs is at least $f_0$.

We seek to solve the following QoS-constrained entanglement path optimization problem.

**Problem 3.** *The min-cost feasible entanglement path optimization problem seeks a feasible entanglement path between Alice and Bob of minimal cost.*

**Remark 3.** *By seeking an optimal entanglement path we mean to find not only the path, but also the number of EPR pairs over each edge of the path involved in entanglement purification.*

Problem 3 is NP-hard, as stated in Lemma 2. To prove the hardness, we can focus on a degenerated problem instance where $C_e(m_e) = w_e m_e$ and each quantum repeater disposes two qubits, rendering entanglement purification impossible, because it needs to establish two EPR pairs with its preceder and successor in path $P$, leaving no more qubits for entanglement purification. $q_0$ is set to a very small real number that that any path satisfies the entanglement throughput constraint. In this situation, both path cost and fidelity can be formulated as additive metrics. The problem thus degenerates to the constrained min-cost path problem which is NP-hard.

**Lemma 2.** *The fidelity-constrained min-cost entanglement path problem is* NP*-hard.*

Given the hardness result, we focus on approximation algorithm design. A technical challenge distinguishing Problem 3 from the constrained path optimization problems studied in the literature lies in the feasibility requirement, where the number of qubits used at each node $v$ cannot exceed its capacity $Q_v$.

To model the feasibility constraint, we construct an auxiliary graph $G' \triangleq (V', E')$ based on $G$.

- **Vertices**: For each vertex $v \in V - \{s, t\}$, we create $Q_v - 1$ vertices $v_i$, $i \in [1, Q_v - 1]$. $v_i$ corresponds to $v$ disposing $i$ available qubits for entanglement. Note that any $v \in V - \{s, t\}$ is a quantum repeater connecting two nodes in the $st-$path, $v$ can take at most $Q_v - 1$ qubits to entangle with any neighbor. We then create $Q_s$ vertex $s_i$, $i \in [1, Q_s]$, representing $s$ and $Q_t$ vertices $t_i$, $i \in [0, Q_t - 1]$, representing $t$. Finally, we create a virtual source node $s'$ and virtual destination node $t'$.
- **Edges**: For each edge $(u, v) \in E$, we create edges $(u_i, v_j)$ if $Q_v - j \leq i$ and $Q_v - j \leq C_l$, where $C_l$ is link capacity.The rationale behind our construction is below: $(u_i, v_j)$ represents $Q_v - j$ EPR pairs between $u$ and $v$. This is feasible as (1) $u_i$ implicates that $u$ disposes $i$ qubits; (2) since $v_j$ represents $v$ disposing $j$ available qubits, it can establish $Q_v - j$ EPR pairs with $u$ as long as $Q_v - j \leq i$ and $Q_v - j \leq C_l$. We set the cost of $(u_i, v_j)$ as
$$C'_{(u_i, v_j)} = C_{(u,v)}(Q_v - j).$$
Finally, we create an edge between $s'$ and $s_i$ for each $1 \leq i \leq Q_t$, $t_i$ and $t'$ for each $0 \leq i \leq Q_t - 1$ and set the edge cost to zero and fidelity to one.

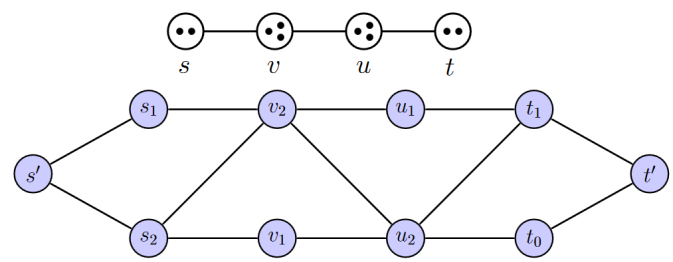


Fig. 6. An example of $G$ (upper) and $G'$ (lower): $s$ and $t$ dispose 2 qubits, while $v$ and $u$ dispose 3. Path $s' - s_2 - v_1 - u_2 - t_0 - t'$ corresponds to path $s - v - u - t$ by purifying 2 EPR pairs at the edges $(s, v)$ and $(u, t)$.

The following lemma follows from our construction of $G'$.

**Lemma 3.** *There is a bijective map between each feasible path from $s'$ to $t'$ and a feasible path between $s$ and $t$ in $G$.*

Our fidelity-constraint path optimization algorithm, whose pseudo-code is given in Algorithm 2, consists of working on $G'$ and then transform the output path to a feasible path in $G$. To facilitate algorithm design, given a path $P$ of fidelity $f(P)$, we define its pseudo-fidelity $\phi(P)$ as

$$\phi(P) \triangleq \ln \frac{4f(P) - 1}{3}.$$

The introduction of pseudo-fidelity allows us to treat the path fidelity as an additive metric. We also transform the fidelity threshold $f_0$ to the corresponding pseudo-fidelity threshold $\phi_0$.

Moreover, let $\lambda_e$ denote the expected number of purified EPR pairs produced over edge $e \in P$, and let $\lambda(P)$ denote the expected end-to-end EPR pairs generated along $P$ by letting the quantum routers perform entanglement swapping. We have

$$\lambda(P) = \min_{e \in P} \quad \lambda_e \prod_{v \in P, v \neq \{s,t\}} p_v.$$

We note that the above computed $\lambda(P)$ actually gives its lower-bound, as if we carefully optimize the order, under which entanglement swapping is performed at the intermediate routers, we can achieve better end-to-end throughput [8]. The price to pay is the additional delay to establish the end-to-end entanglements, as we need to separate the swapping operations into rounds. Our algorithm can be applied to such optimization

as long as the optimized throughput can be expressed as a function of $P$ and $\lambda_e$ along $P$.

To turn $\lambda(P)$ into an additive metric, we define the log-throughput $\psi_e \triangleq \ln \lambda_e$, the log-swap-success-probability of node $v$ is termed as $\psi_v \triangleq \ln p_v$. Accordingly, we have

$$\psi(P) = \min_{e \in P} \psi_e + \sum_{v \in P, v \neq \{s,t\}} \psi_v$$

**Data structure**. We maintain a list $\mathcal{L}_v$ at each vertex $v \in V'$ containing quintuples as entries. Each quintuple $l$ represents a feasible path $P_l$ from $s'$ to $v$ that is not dominated by any other quadruple. Specifically, in a quintuple $l \triangleq (c, \hat{\phi}, \psi_b, \hat{\psi}, path)$, $c$ denotes the cost of $P_l$, $\hat{\phi}$ denotes the discretized pseudo-fidelity of $P_l$ with stepsize $\Delta\phi$, $\psi_b = \min_{e \in P}\{\psi_e\}$ denotes the bottleneck edge log-throughput of $P_l$, $\hat{\psi}$ denotes the discretized log-throughput of $P$ with stepsize $\Delta\psi$. The algorithm operates in $G'$, and we need to translate the sequence of visited nodes of path in $G'$ back to $G$, recording this sequence in the path list for loop-free. The entries are indexed by the couple $(\hat{\phi}, \hat{\psi})$. Given two entries $l_i = (c_i, \hat{\phi}_i, \psi_{b,i}, \hat{\psi}_{l,i}, path)$, $i = 1, 2$, we say that $l_1$ dominates $l_2$ if $c_1 \leq c_2$, $\hat{\phi}_1 \geq \hat{\phi}_2$, and $\hat{\psi}_{l,1} \geq \hat{\psi}_{l,2}$. The dominance is strict if at least one inequality holds strictly. Intuitively, an optimal feasible path is not strictly dominated by any feasible path. So we can safely remove dominated triples.

**Algorithm**. The core part of Algorithm 2 is the main loop that iteratively scans all the vertices and their non-dominated labels using a min-heap priority queue $\Gamma$ and updates the corresponding entries. Initially, the queue contains the labels of $s$. The algorithm iteratively processes labels from the queue. For each label $l$ dequeued from $\Gamma$, the algorithm updates the visited label set associated with node $v$, denoted by $\Omega_v$. once the current node $u_i$ is the target node $t'$, meaning that the algorithm find the minimum cost from $s$ to $t$. The algorithm terminates once $u_i = t'$ or $\Gamma$ becomes empty. Update labels procedure involves expanding the label of current node to neighboring nodes. For each neighbor $v_j$ of the current node $u_i$, new labels are generated by iterating over all possible possible discrete values of the fidelity threshold $\hat{\phi}'$, i.e., $\frac{\hat{\phi}}{\Delta\phi} < k \leq \left\lfloor \frac{\hat{\phi} - \phi_0}{\Delta\phi} \right\rfloor$. Then for each $\hat{\phi}'$, we perform the following actions.

- We invoke Algorithm 1 with $Q_v - j$ EPR pairs and fidelity threshold $\hat{\phi}'$ to compute the throughput $\psi_e$, which is the maximal throughput over $e$ satisfying the fidelity constraint, and this procedure can be pre-computed .
- We then update the discretized throughput of the path $P$ concatenated by $e$ by comparing the bottleneck throughput of $P$, $\psi_b$, with $\psi_e$ and updating $\hat{\psi}'$ accordingly.
- If the log-throughput of the new path satisfies the throughput constraint, we then create a quintuple $l'$ corresponding to the new path and add it to $\mathcal{L}_{v_j}$ if it is not dominated by any other entry.
- By the above steps, we maintain in $\mathcal{L}_{v_j}$ the min-cost feasible $sv_j-$path for each $(\hat{\phi}, \hat{\psi})$ couple.
- Once the scan and update process is terminated, Algorithm 2 selects the min-cost entry in $\mathcal{L}_{t'}$ and outputs the corresponding path $P$ and the related entanglement purification scheme. As the vertices $t_i$, $1 \leq i \leq Q_t$, map to $t$, the output is an $st-$path.

We next prove that Algorithm 2 outputs an $st-$path that can be arbitrarily close to optimum.

**Definition 1.** *Let $P^*$ denote the optimal path solving Problem 3. An $st-$path $P$ is called an $\epsilon$-optimal feasible path if the following inequality holds.*

$$C(\hat{P}) \leq C(P^*),\ \phi(\hat{P}) \geq (1-\epsilon)\phi_0,\ \psi(\hat{P}) \geq (1-\epsilon)\psi_0.$$

The rationale behind the above definition is to allow a small quantity $\epsilon$ of overflow on the fidelity and throughput constraints. Theorem 3, whose proof is omitted here, proves that Algorithm 2 outputs an $\epsilon$-optimal feasible path.

**Theorem 3.** *If $\Delta\phi \leq \epsilon\phi_0/|V|$ and $\Delta\psi \leq \epsilon\psi_0/|V|$, Algorithm 2 outputs an $\epsilon$-optimal feasible path.*

**Space and time complexity of Algorithm 2**. Recall the construction of $G'$, each node $v \in G$ generates $Q_v$ nodes in $G'$. For each node in $G'$, we need to maintain a list of at most $\frac{\phi_0\psi_0}{\Delta\phi\Delta\psi}$ entries. Invocation of Algorithm 1 can be pre-computed and stored for each pair of number of qubits and threshold, taking $O(\phi_0 \max_v Q_v/\Delta\phi)$ space. Noticing $|V'| = O(Q)$ and $|E'| = O(Q^2)$ with $Q \triangleq \sum_{v \in V} Q_v$ and setting $\Delta\phi$ and $\Delta\psi$ based on Theorem 3, the total space overhead sums up to $O(|V|^2 Q\epsilon^{-2})$. The time complexity is dominated by the main loop, with the heap manipulations, which sums up to $O\left((|V'| + |E'|)\log V' \frac{\phi_0\psi_0}{\Delta\phi\Delta\psi}\right)$, i.e., $O\left(Q^2 \log Q |V|^2 \epsilon^{-2}\right)$. Though polynomial, the time complexity can be significantly high for large $Q$. In this setting, we can decrease the size of the auxiliary graph to reduce the complexity in the following way. For each node, instead of creating $Q_v - 1$ vertices in the auxiliary graph, we set a stepsize $\Delta Q$ and build $Q_v/\Delta Q$ nodes in $G'$, thus reducing the graph size by factor $\Delta Q$, with $\Delta Q = 1$ being the degenerated case we analyzed in this subsection. The price to pay is the loss of efficiency in the solution we find, as we increase the granularity from 1 to $\Delta Q$. In practice, we can tune $\Delta Q$ to trade off the efficiency against complexity.

## IV. Performance Evaluation

We study the scheduling and routing algorithms at the network layer of quantum networks. Routing algorithms leverage the link layer model [10], [26]. As mentioned in research [31], [16], entanglement routing can be evaluated through simulated experiments as long as the physical and link layers accurately reflect the physical facts.

Therefore, we conduct two sets of simulations to empirically evaluate our entanglement purification scheduling and routing algorithms so as to get more insights on these two fundamental problems that cannot be fully captured by our analytical demonstrations presented in the paper.

**Algorithm 2:** computing fidelity-constrained min-cost path

**Input:** $G'$, $s'$, $t'$, $\psi_0 = \ln q_0$, $\phi_0 = \phi(f_0)$
**Output:** a feasible entanglement path

**foreach** $v \in V'$ **do**
  **if** $v = s'$ **then**
    $l \leftarrow (0, 0, \infty, \infty, \{s\})$
    $\mathcal{L}_v \leftarrow \{l\}$
    $\Gamma.enqueue((c, l, v))$
  **else**
    $\mathcal{L}_v \leftarrow \emptyset$
  $\Omega_v \leftarrow \emptyset$
**while** $\Gamma$ *is not empty* **do**
  $(c, l, u_i) \leftarrow \Gamma.dequeue()$
  **if** $l$ *not in* $\Omega_{u_i}$ **then**
    add $l$ to $\Omega_{u_i}$
    **if** $u_i = t'$ **then**
      break
    **foreach** $v_j$ *in* $neighbor(u_i)$ **do**
      **if** $v$ *not in path* **then**
        **for** $k = 1$ ***to*** $\left\lfloor \frac{\hat{\phi} - \phi_0}{\Delta\phi} \right\rfloor$ **do**
          $\hat{\phi}' \leftarrow \hat{\phi} - k\Delta\phi$
          invoke Algorithm 1 with $Q_v - j$ EPR pairs and threshold $-k\Delta\phi$ to compute the throughput $\psi_e$, so $\psi_e = \ln((\hat{\xi}'/b') \cdot (Q_v - j))$
          **if** $\psi_e \leq \psi_b$ **then**
            $\hat{\psi}' \leftarrow \left\lceil \frac{\psi_{v_j} + \hat{\psi} + \psi_e - \psi_b}{\Delta\psi} \right\rceil \cdot \Delta\psi$;
          **else**
            $\hat{\psi}' \leftarrow \left\lceil \frac{\psi_{v_j} + \hat{\psi}}{\Delta\psi} \right\rceil \cdot \Delta\psi$;
          **if** $\hat{\psi}' \geq \psi_0$ **then**
            $l' \leftarrow (c + C_e(Q_j - j), \hat{\phi}', \min\{\psi_e, \psi_b\}, \hat{\psi}', path + [v])$;
            **if** $l'$ *is not dominated by any entry in* $\mathcal{L}_{v_j}$ **then**
              add $l'$ to $\mathcal{L}_{v_j}$, add $l'$ to $\Omega_{v_j}$;
              $\Gamma.enqueue((c + C_e(Q_j - j), v_j, l'))$;
  **if** *no entry in* $\mathcal{L}_{t'}$ *maps to a feasible path* **then**
    **return** no feasible path;
  **return** the path corresponding to the entry in $\mathcal{L}_{t'}$ with minimum cost;

### A. Entanglement Purification Scheduling

Our first set of simulations focus on entanglement purification scheduling investigated in Section II. For single-hop scheduling, we compare Algorithm 1 with the two purification strategies depicted in Figure 2, with the left strategy following a symmetrical pattern by parallelizing the purification process and the right strategy adopting a serial and recursive pattern. We term them as SYMMETRIC and PUMPING strategies, respectively. Figure 7 traces the fidelity of purified EPR pair with different number of input EPR pairs. We observe that for SYMMETRIC and PUMPING, the fidelity gain is much less pronounced when the number of input EPR pairs is large. The output fidelity becomes almost constant after 10 input EPR pairs. In contrast, our algorithm is still able to achieve a relatively large fidelity gain. To better understand this phenomenon, we examine eq. (1) and find that the fidelity gain decreases with input fidelity. Therefore, simple recursive purification scheduling strategies can bring significant fidelity gain at the first few rounds but can hardly increase the fidelity afterwards. Our simulation results thus demonstrate the importance of designing an optimal strategy as Algorithm 1.

We then proceed to the multi-hop scenario to investigate the interaction between entanglement purification and entanglement swapping. Specifically, we compare PURIFY-AND-SWAP against SWAP-AND-PURIFY and another composite strategy, called SWAP-PURIFY-SWAP, described as follows.

1) Divide the entire path into $h$ portions of length $\lfloor l/h \rfloor$ or $\lceil l/h \rceil$ with $l$ denoting the path length;
2) Perform SWAP-AND-PURIFY within each portion;
3) Stitch the $h$ purified EPR pairs into a final EPR pair.

Figure 8 traces the simulation results with different number of hops and elementary EPR pairs per hop, with randomly selected fidelity values in $[0.85, 0.99]$. We observe that PURIFY-AND-SWAP always outperforms the other two strategies, which is consistent to our theoretical result. SWAP-AND-PURIFY performs the worst among the simulated strategies. The performance of SWAP-PURIFY-SWAP depends on the parameters $h$. When $h$ approaches $l$, the performance of SWAP-PURIFY-SWAP approaches that of PURIFY-AND-SWAP, with $h = l$ degenerating to PURIFY-AND-SWAP. This result demonstrates that it is usually better to perform entanglement purification over individual links, i.e., a relatively local operation, before entanglement swapping, which is a more global operation.

### B. Fidelity-constrained Entanglement Routing

Our second set of simulations focus on entanglement routing, by integrating our purification scheduling algorithm.

For experiment setting, we use NetworkX to randomly generate a Waxman random graph with 300 nodes and link capacity of 10, and a $5 \times 5$ grid network with link capacity of 15. Consistent with the experiment setting in the literature [20], we configure the fidelity of links to follow a normal distribution within the range of $[0.8, 1]$ and randomly selected source and destination. The objective is to find the min-cost path from a random source to destination while meeting the

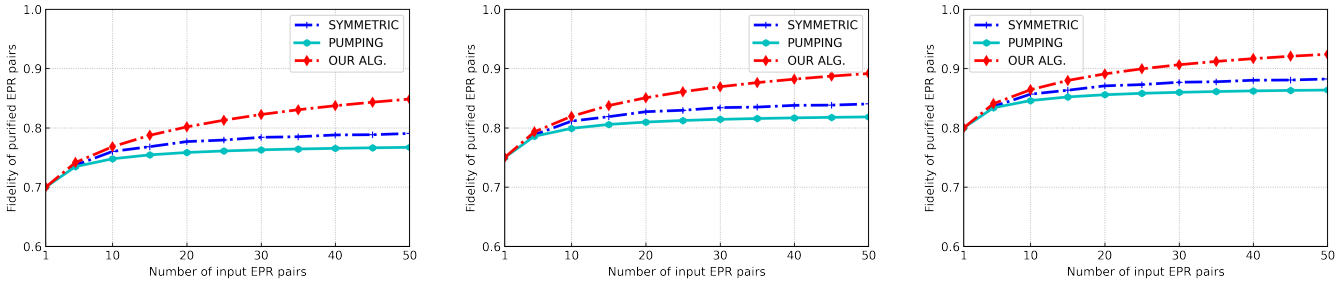


Fig. 7. Performance comparison of single-hop entanglement purification scheduling algorithms: the fidelity of elementary EPR pairs is 0.7, 0.75, and 0.8 in the left, middle, and right figures.

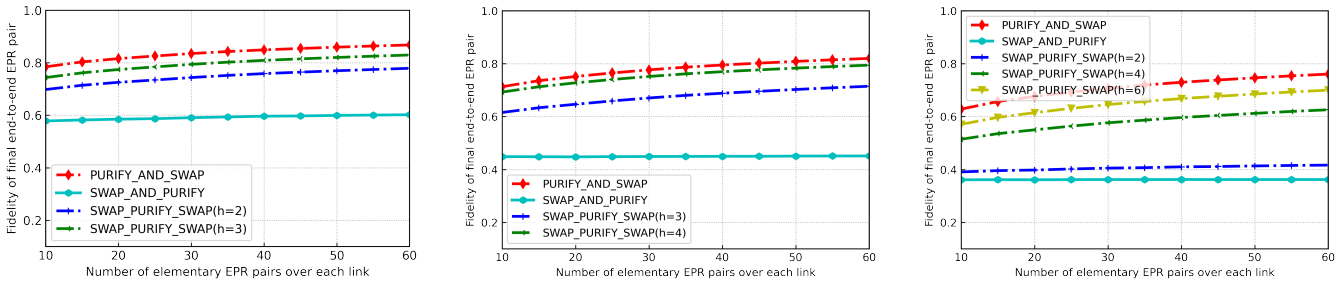


Fig. 8. Performance comparison of multi-hop entanglement purification scheduling algorithms: left: $l = 6$, middle: $l = 9$, right: $l = 12$.

throughput threshold of 1 and the specified fidelity threshold. Since fidelity thresholds below 0.8 are generally considered to lack practical relevance, we evaluate the performance of the simulated algorithms at fidelity thresholds of 0.8, 0.85, 0.9 for Waxman graph (Figure 9) and grid network (Figure 10). We evaluate our algorithm with two state-of-the-art quantum routing algorithms Q-PATH and Q-LEAP, both designed to find end-to-end paths with fidelity and throughput guarantee [20]. To further analyze path-searching performance, we design another comparison algorithm, called Q-STEP, which employs the same PUMPING purification strategy as Q-PATH and Q-LEAP while the same path computation strategy as our algorithm. We set the cost to be the number of qubits consumed in the corresponding path.

Figure 9 and 10 compare of our algorithm under different discretization stepsizes $\Delta\phi$ against Q-STEP, Q-PATH and Q-LEAP. We observe from the results that our algorithm achieve the best performance in terms of success probability and path cost, which is coherent to our theoretical quasi-optimality guarantee. Furthermore, the performance gap underscores the importance of carefully optimizing the entanglement path computation by holistically integrating purification scheduling in a cross-layer manner, which is the central focus of our work.

More specifically, we observe from the results that, as the fidelity threshold increases, the success probability of the simulated algorithms decreases, and the path cost increases. This can be explained by the fact that, when the fidelity constraint becomes more stringent, we need more quantum resources to perform additional rounds of entanglement purification, leading to higher cost. If the fidelity threshold is too high, it is impossible to construct feasible end-to-end entanglement, which explains the result that the success probability is less than 1 for some cases in the simulations. The higher success probability of Q-STEP against Q-PATH and Q-LEAP demonstrates the performance gain brought by the path computation strategy in our algorithm, as Q-STEP employs the same purification scheduling strategy as Q-PATH and Q-LEAP. The higher success probability of our algorithm against Q-STEP further demonstrates the performance gain brought by the purification scheduling strategy we develop.

We conclude our simulation by comparing the running time of the simulated algorithms. We observe that our algorithm incurs higher time overhead because our approach seeks a near-optimal solution. However, this overhead can be tuned by the discretization stepsize such that, when $\Delta\phi = 0.01$ and $0.02$ in the simulated networks, the running time does not exceed 5 seconds. The typical lifespan of a qubit extends to tens of seconds. Recent advancements in the field, as highlighted by research [34], have shown that the qubit lifetime can surpass one hour. Our algorithm allows for a trade-off between precision and efficiency by adjusting the discretization step size parameter, $\Delta\phi$. As a result, even when executed on demand, our algorithm is suitable for medium-sized quantum networks today and holds the potential for deployment in large-scale quantum networks in the foreseeable future.

## V. Related Works

Establishing end-to-end entanglements reliably and efficiently is probably one of the most fundamental tasks in quantum networks. Research on entanglement routing was initiated by the physics community, with propositions for specific network topologies such as lines, rings, and grids [28], [29], [32]. Their main focus was to develop feasible ways to achieve end-to-end entanglements to support quantum information transmission. However, relying on specific network topologies significantly limits their generality and practicality.

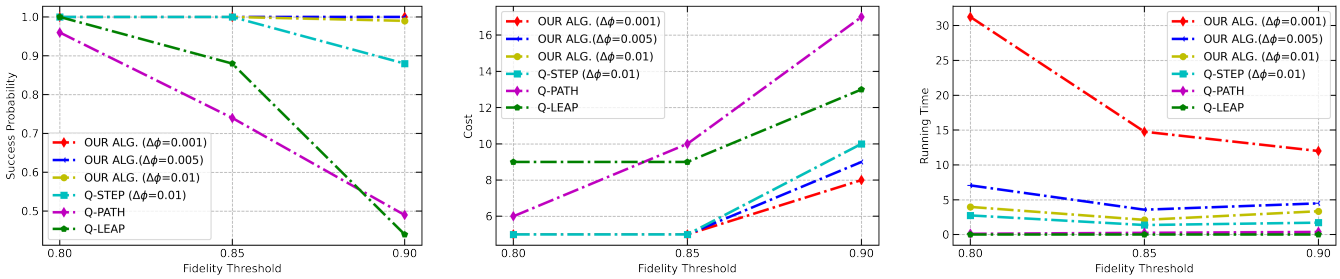


Fig. 9. Performance of our algorithm with Q-STEP, Q-PATH and Q-LEAP (channel capacity = 10, Waxman topology with 300 nodes): left: success probability, middle: cost; right: running time

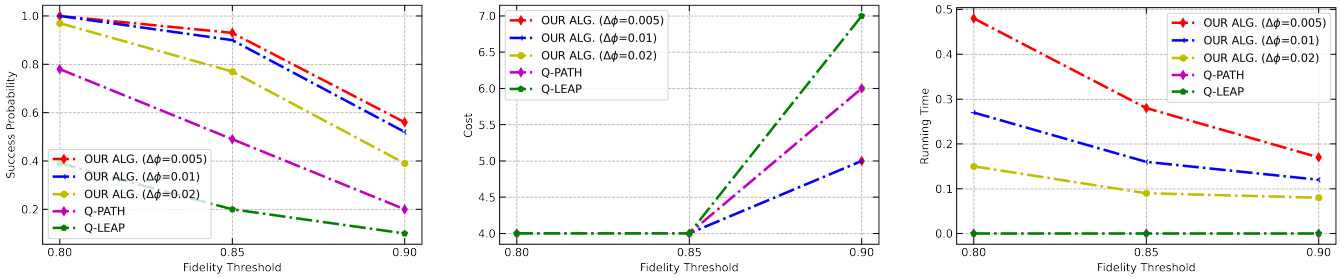


Fig. 10. Performance of our algorithm with Q-STEP, Q-PATH and Q-LEAP (channel capacity = 15, Grid $5 \times 5$): left: success probability, middle: cost; right: running time

More recently, entanglement routing began to capture research interests from the networking community, resulting in a handful of propositions for generic network topology, e.g., [30], [37], [27], [7]. To further ensure end-to-end fidelity, a few works have integrated entanglement purification into the process of entanglement path optimization, by solving the corresponding constrained routing problems [38], [20], [33], among which [38], [20] considered problem with network-layer perspective, but they only considered simple PUMPING purification scheme or enumerated all combinations.

Purification scheduling is a critical yet largely unexplored area, essential for enhancing the fidelity of final purified entanglements. A recent comprehensive survey on entanglement purification and routing [19] highlights the significant challenge in deriving an optimal entanglement purification schedule, labeling it as highly intractable. Due to this complexity, commonly used schemes are typically simple and intuitive algorithms, such as PUMPING [14], [4] and SYMMETRIC [6], [2], [11], which lack any guarantee of optimality, as noted in recent studies [38], [20], [33], [18], [25]. Recent research [9] has considered optimal entanglement purification scheduling and routing. However, it only solve the optimal purification scheduling in the bit-flip case and does not address purification scheduling in the Werner case. As demonstrated in their paper, purification scheduling in the Werner case is more challenging due to the lack of structural properties present in the bit-flip case. Recent work [25], [18], by experiments, discover the throughput of PURIFY-AND-SWAP outperforms SWAP-AND-PURIFY. However, these works do not give closed-from condition and theorem prove for the optimality of PURIFY-AND-SWAP. These gaps in research motivates our work to develop an algorithmic framework for joint entanglement scheduling and routing.

## VI. Conclusion and Perspective

In this paper, we first design an optimal entanglement purification scheduling framework for a single hop. We then analyze the optimality of PURIFY-AND-SWAP swapping strategy in multi-hop scenarios. Finally, we combine these insights to develop an polynomial time algorithmic framework to computing minimum-cost end-to-end entanglement path, subject to $\epsilon-$optimality in fidelity and throughput constraints, focusing on entanglement purification scheduling and entanglement routing. Our main technicalities hinge on the structural properties of the quantum operations, notably entanglement purification and entanglement swapping, combined with algorithmic tools in combinatorial optimization. Our algorithmic frameworks are highly flexible, allowing for a balance between accuracy and computation time through the choice of $\Delta\phi$. Our algorithmic framework can also serve as a key functional blocks for entanglement configuration, enabling high-fidelity quantum information transmission by effectively leveraging limited quantum resources, thus fully harnessing the unmatched potential of quantum networks. The superiority of our algorithm frameworks are also demonstrated by extensive simulations.

Looking ahead, we plan to extend our framework by incorporating measurement error to account for imperfect measurements. One direction for future research is to factor in decoherence time, further enhancing the realism of the model. Another avenue is to allow EPR pairs across different entanglement paths to be purified, and to design a joint purification scheduling and path optimization algorithm in this more complex context.

## REFERENCES


[1] M. J. Bayerbach, S. E. D'Aurelio, P. van Loock, and S. Barz. Bell-state measurement exceeding 50% success probability with linear optics. *Science Advances*, 9(32):eadf4080, 2023.

[2] C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, and W. K. Wootters. Purification of noisy entanglement and faithful teleportation via noisy channels. *Physical Review Letters*, 76:722–725, Jan 1996.

[3] K. Bourzac. 4 tough chemistry problems that quantum computers will solve. *IEEE Spectrum*, 54(11):7–9, 2017.

[4] H.-J. Briegel, W. Dür, J. I. Cirac, and P. Zoller. Quantum repeaters: the role of imperfect local operations in quantum communication. *Physical Review Letters*, 81(26):5932, 1998.

[5] T. A. Brun. *Quantum Error Correction*. Oxford University Press, 2020.

[6] V. Bužek and M. Hillery. Quantum copying: Beyond the no-cloning theorem. *Physical Review A*, 54:1844–1852, Sep 1996.

[7] 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.

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

[9] L. Chen and Z. Jia. On optimum entanglement purification scheduling in quantum networks. *IEEE Journal on Selected Areas in Communications*, pages 1–14, 2024.

[10] A. Dahlberg, M. Skrzypczyk, T. Coopmans, L. Wubben, F. Rozpundefineddek, M. Pompili, A. Stolk, P. Pawełczak, R. Knegjens, J. de Oliveira Filho, R. Hanson, and S. Wehner. A link layer protocol for quantum networks. In *Proc. SIGCOMM*, 2019.

[11] D. Deutsch, A. Ekert, R. Jozsa, C. Macchiavello, S. Popescu, and A. Sanpera. Quantum privacy amplification and the security of quantum cryptography over noisy channels. *Physical Review Letters*, 77:2818–2821, Sep 1996.

[12] U. Dorner, A. Klein, and D. Jaksch. A quantum repeater based on decoherence free subspaces. *Quantum Information and Computing*, 8(5):468–488, may 2008.

[13] W. Dur and H.-J. Briegel. *Chapter 12 in Quantum Information: Purification and Distillation*, pages 231–263. John Wiley and Sons, 2016.

[14] W. Dür, H.-J. Briegel, J. I. Cirac, and P. Zoller. Quantum repeaters based on entanglement purification. *Physical Review A*, 59:169–181, Jan 1999.

[15] R. P. Feynman. Simulating Physics with Computers. *International Journal of Theoretical Physics*, 21(6-7):467–488, June 1982.

[16] X. Hong, K. Xu, and M. Gerla. Scalable routing protocols for mobile ad hoc networks. *IEEE Network*, 16(4):11–21, 2002.

[17] Z. Jia and L. Chen. From entanglement purification scheduling to fidelity-constrained entanglement routing. *arXiv*, 2024.

[18] Z. Jia and L. Chen. On fidelity-oriented entanglement distribution for quantum switches. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, pages 1–1, 2024.

[19] H. Leone, N. R. Miller, D. Singh, N. K. Langford, and P. P. Rohde. Qunet: Cost vector analysis and multi-path entanglement routing in quantum networks. *arXiv*, 2021.

[20] J. Li, M. Wang, K. Xue, R. Li, N. Yu, Q. Sun, and J. Lu. Fidelity-guaranteed entanglement routing in quantum networks. *IEEE Transactions on Communications*, 70(10):6748–6763, 2022.

[21] R. V. Meter. *Quantum Networking*. Wiley, 2014.

[22] 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.

[23] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[24] J.-W. Pan, C. Simon, C. Brukner, and A. Zeilinger. Entanglement purification for quantum communication. *Nature*, (6832):1067–1070, 2001.

[25] N. K. Panigrahy, T. Vasantam, D. Towsley, and L. Tassiulas. On the capacity region of a quantum switch with entanglement purification. *Proc. INFOCOM*, 2023.

[26] 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*, (1), 2019.

[27] 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*, 2019.

[28] S. Pirandola. End-to-end capacities of a quantum communication network. *Communications Physics*, 3:1–10, 2019.

[29] E. Schoute, L. Mancinska, T. Islam, I. Kerenidis, and S. Wehner. Shortcuts to quantum network routing. *arXiv*, 2021.

[30] S. Shi and C. Qian. Concurrent entanglement routing for quantum networks: Model and designs. In *Proc. SIGCOMM*, 2020.

[31] S. Shi, X. Zhang, and C. Qian. Concurrent entanglement routing for quantum networks: Model and designs. *IEEE/ACM Transactions on Networking*, 32(3):2205–2220, 2024.

[32] G. Vardoyan, S. Guha, P. Nain, and D. Towsley. On the stochastic analysis of a quantum entanglement distribution switch. *IEEE Transactions on Quantum Engineering*, 2:1–16, 2021.

[33] M. Victora, S. Krastanov, A. Sanchez de la Cerda, S. Willis, and P. Narang. Purification and entanglement routing on quantum networks. *arXiv*, 2020.

[34] P. Wang, C. Luan, M. Qiao, et al. Single ion qubit with estimated coherence time exceeding one hour. *Nature Communications*, 12(233), 2021.

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

[36] R. F. Werner. Quantum states with Einstein-Podolsky-Rosen correlations admitting a hidden-variable model. *Physical Review A*, 40:4277–4281, Oct 1989.

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

[38] Y. Zhao, G. Zhao, and C. Qiao. E2E fidelity aware routing and purification for throughput maximization in quantum networks. In *Proc. INFOCOM*, 2022.