Swapping and Purification Scheme Optimization for Entanglement Distribution in Quantum Networks

Jiyao Liu, *Student Member, IEEE*, Xinwen Zhang, Xinliang Wei, *Member, IEEE*, Xuanzhang Liu, Yuzhou Chen, Hongchang Gao, and Yu Wang, *Fellow, IEEE*

***Abstract*—Swapping and purification are the two fundamental building blocks for high-fidelity entanglement distribution in multi-hop quantum networks. Unfortunately, it is still a mystery how they intertwine with each other to affect the fidelity and cost of end-to-end entanglements. Current scheduling algorithms consider this problem under relatively limited assumptions and a critical yet unjustified conjecture. In this work, we first consider more general assumptions with operation failures and, accordingly, extend a tree-based modeling for joint swapping and purification. Then, we analytically prove the previous conjecture that the optimal strategy under *Binary system* is always to purify the entanglements before any swapping. This sheds light on the protocol and device design for quantum networks. We then further propose a tree-based algorithm, which can efficiently schedule swapping and purification along a path for both *Binary* and *Werner systems*. Extensive simulations of the proposed method against state-of-the-art solutions show that our method uses fewer entanglements to establish qualified end-to-end entanglements, and thus achieves higher network throughput.**

***Index Terms*—Quantum Swapping, Quantum Purification, Fidelity, Entanglement Distribution, Quantum Network.**

I. Introduction

In a multi-hop quantum network [1]–[3], a pair of entangled particles (EPR pairs) needs to be distributed to the source and destination nodes (denoted as a Source-Destination (SD) pair) so that quantum bits/status can be transferred between them via teleportation. Such entanglements are the key resource in quantum networks and distributed quantum computing [4]–[9]. However, establishing long-distance (multi-hop) entanglements is a challenging task. Unlike in classic networks, we cannot simply copy the states of particles due to the no-cloning principle of quantum states. Simply sending photons (the usual particle used to carry quantum information) in optical fibers far away is also infeasible because photons decay exponentially as they travel in fibers.

Entanglements in quantum networks are initially generated over short quantum links (usually optical fibers). One link may contain multiple channels to generate entanglements on the two endpoints of the link. Such link-level entanglements can be later connected by the swapping operation to form long-distance connections. *Quantum swapping* takes two adjacent entanglements as inputs and generates a longer entanglement. This operation can be repeated over a path until we obtain an end-to-end (E2E) entanglement on the desired SD pair. However, while swapping establishes a longer entanglement, it also decreases the quality of the entanglement (measured by *fidelity*). Then, *quantum purification* is introduced in quantum networks: by sacrificing additional entanglements on the same two nodes, we can improve the fidelity of the entanglement. In short, swapping extends the distance of entanglements at the price of their fidelity; while purification improves fidelity by consuming additional entanglement(s). Combining these two basic operations, we can build long-distance (multi-hop) entanglements with high fidelity.

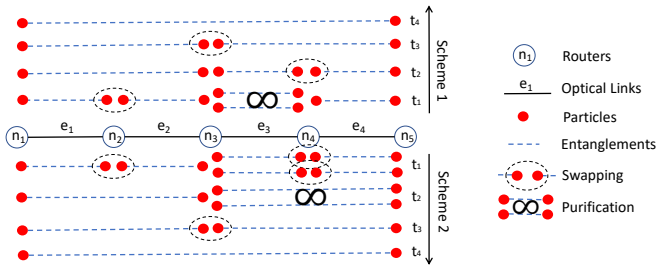


Fig. 1: Swapping and purification schemes (SPS): two example schemes along a path of four quantum links.

As shown in Fig. 1, we now consider an example of a quantum path from node n_1 to n_5 connected by four optical links, which generate entangled particle pairs as entanglements. Fig. 1 shows two possible sequences of swapping and purification operations (schemes) to establish the E2E entanglement between n_1 and n_5. In Scheme 1, at t_1, links e_1 and e_2 generate one entangled pair, separately. Because each pair has one particle on router n_2, swapping can be performed by n_2 to obtain an entanglement between n_1 and n_3 (at t_2). At the same time, link e_3 generates two entanglements, which share the same end nodes (n_3 and n_4), at t_1. Thus, a purification can be performed collaboratively by n_3 and n_4. If it succeeds, an entanglement of higher fidelity is established on link e_3. Via more swappings, Scheme 1 establishes an E2E entanglement between n_1 and n_5 at t_4. Scheme 2 gives an E2E entanglement on the same ends via a different swapping and purification order. Note that classic communications parts of those operations are ignored in this paper, and t_i's are not real-time stamps but only for marking the order of operations.

How swapping and purification interfere with each other, i.e., scheduling the swapping and purification operations to

J. Liu, X. Zhang, X. Liu, H, Gao and Y. Wang are with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19112. {jiyao.liu, ellenz, xzliu, hongchang.gao, wangyu}@temple.edu. X. Wei is with Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, 518055, China. Email: xl.wei@siat.ac.cn. Y. Chen is with the Department of Statistics at University of California, Riverside, Riverside, CA 92521. yuzhouc@ucr.edu. The work is partially supported by the US NSF (Grant No. CNS-2006604 and CNS-2128378).

reach a specific final E2E fidelity with the least number of entanglements consumed, is still one open challenge in quantum networks. As shown in the example of Fig. 1, to acquire one entanglement for an SD pair connected by a quantum path, different *Swapping and Purification Schemes* (SPS) [10], [11] can be applied. Two extreme types of SPS solutions are (i) *PS scheme:* purify the entanglements immediately after their generation on each link, then swap links to establish the E2E entanglement; (ii) *SP scheme:* firstly generate massive low-quality E2E entanglements by swapping and then perform purification over them. Obviously, you can also have a hybrid scheme that purifies intermediate entanglements after some swappings but before they are end-to-end. In Fig. 1, Scheme 1 follows the PS scheme, while Scheme 2 is a hybrid scheme.

A few studies have explored joint scheduling of swapping and purification but within a relatively limited scope. Some works [11]–[14] directly adopt one of the schemes without detailed justification, while others [15] provide only limited empirical study. Until recently, some analysis [16]–[18] has theoretically studied joint swapping and purification schemes but based on impractical assumptions, such as the absence of operation failures. To the best of our knowledge, no existing work has comprehensively analyzed the optimal SPS solutions (e.g., PS/SP/hybrid) while accounting for operation failures. This remains an important open problem as the optimal SPS is critical for device and protocol design for quantum networks. For example, if PS is indeed the best one, we may consider embedding purification into the link level (to allow fast purification over optical fibers) and design the repeaters accordingly. If not, we may need to consider collaboration between non-adjacent routers, which can be much more complicated.

In this paper, we fully explore the joint SPS problem. We first generalize the assumptions (by considering arbitrary operation orders and possible swapping/purification failures) and formulate an *Optimal Swapping and Purification Scheme* (OSPS) problem. By absorbing purification into the current tree-based modeling, we introduce the concept of *Swapping and Purification Tree* (SPT) to represent and analyze any possible solution of OSPS. We also propose a dynamic programming based SPS method, which enhances our understanding of the structure of SPS problem. These provide the foundation of our further analysis and solutions. By using the SPT model, we formally prove that the optimal solution of OSPS under *Binary system* follows the PS scheme. Furthermore, we propose a new tree-based algorithm, which can find feasible solutions under both *Binary* and *Werner systems*. We also prove that this tree-based method indeed follows a PS scheme in *Binary systems*. Via extensive simulations, we confirm our proposed method outperforms state-of-the-art solutions with both noisy and noiseless gates (with noisy channels). Overall, this paper makes four key contributions:

1) We present a new modeling method for joint SPS, by considering the impact of operational failures. We then formally introduce the optimal SPS problem which aims to minimize the expected cost while achieving required E2E fidelity. In addition, we present the network-wide scheduling optimization based on the results from OSPS solution as well.
2) We prove that the optimal SPS solution under Binary systems follows the PS scheme (i.e., performing purification before swapping). To the best of our knowledge, this is the first formal theoretical analysis with operation failures on the SPS problem.
3) We propose a new branching tree method for SPS problem, which is able to handle joint (hybrid) optimization of swapping and purification. We also show that it always follows the PS scheme in Binary systems.
4) We conduct experiments for both Binary and Werner systems under different noise levels, and simulation results confirm that the proposed tree-based method outperforms the state-of-the-art solutions.

The remainder of this paper is organized as follows. Section II reviews related work on entanglement distribution and quality improvement via purification. Section III introduces some basics on quantum networks and devices. Section IV first introduces our generalized SPS model and the formulated SPS problem, and then presents a general solution based on dynamic programming and an overall network-wide scheduling. Section V elaborates our analysis of the optimal SPS in Binary systems. Based on the analysis, we then propose an efficient SPS algorithm in Section VI that can find feasible scheduling in both Binary and Werner systems. Evaluations of the proposed methods are provided in Section VII. Section VIII concludes the paper with possible future directions. A preliminary version of this paper appears as [19].

II. Related Works

In this section, we briefly review the current works on entanglement distribution and modeling of swapping and purification in quantum networks. Table I summarizes the comparison between related models/methods reviewed in this section and our work.

A. Entanglement Distribution and Routing

Early works on entanglement distribution focus on maximizing network throughput or minimizing delay without considering fidelity and purification. For example, ORED [13] models throughput maximization as a linear programming problem, while [20] proposes an opportunistic routing to reduce the delay. [21] has studied stochastic behavior of entanglement swappings and proposed a swapping scheduling method for path-level latency minimization. [22] has more specific assumptions on entanglement generation on optical links and uses predetermined paths between SD pairs for routing. FENDI [23] adopts the ORED framework and considers fidelity but not purification, improving fidelity and delay by reducing path length. Yang *et al.* [24] propose an online routing algorithm to enhance efficiency and scalability. E2E entanglements established by these fidelity-agnostic algorithms typically have low fidelity: as the hop number increases, their fidelity drops quickly and may no longer be usable for applications or further purification.

B. Enhancement of Fidelity

Improving E2E entanglement fidelity can be attempted through purification over multiple E2E entanglements (as an SP scheme), though it is less effective for low-fidelity E2E

entanglements. To overcome this, Zhao *et al.* [11] and Li *et al.* [12] both design greedy algorithms to keep finding the critical link (with either the largest gradient or the most fidelity increase after purification) and purifying it until the E2E fidelity threshold is met. Such methods use the PS scheme, but their calculation of expected sacrificial entanglements is greatly simplified (and the results are limited, e.g., both their solutions do not work under the Werner system). Jia and Chen also study joint swapping and purification in their recent series [16]–[18]. Their latest work [18] is able to find near-optimal paths between SD pairs and performs path selection to maximize overall throughput. However, their optimality is defined on additive pseudo metrics (not the real fidelity and throughput) and they assume no operation failures. As a result, it is non-trivial to obtain the real cost of the proposed solution. Dur *et. al* [14] propose a hybrid method that purifies after a certain fixed number of swappings. Victoria *et. al* [25] consider a fixed hybrid SPS pattern like that in [14] in the network context. Jiang *et. al* [26] optimizes the total operation time via a dynamic programming based approach. All the above works ignore the swapping and/or purification failures. Instead, our tree-based solution is able to handle both types of failures and complicated hybrid schemes.

TABLE I: Models and schemes considered in related works.

	fidelity	purif.	noise model	noisy gates	failure	hybrid SPS
[13]	×	×	×	×	S	×
[20]	×	×	×	×	S	×
[22]	×	×	×	✓	S	×
[21]	×	×	×	✓	S	×
[23]	✓	×	W	✓	S	×
[24]	×	×	×	×	S	×
[27]	×	×	×	×	S	×
[28]	×	×	×	×	S	×
[11]	✓	✓	bit flip	×	×	×
[12]	✓	✓	bit flip alike	×	P	×
[14]	✓	✓	B & W	✓	P	✓
[16]–[18]	✓	✓	B & W	×	×	✓
[25]	✓	✓	W	✓	P	✓
[26]	✓	✓	W	✓	×	✓
this paper	✓	✓	B & W	✓	S & P	✓

×: not considered; ✓: considered;
B/W: Binary/Werner system; S/P: swap/purify.

C. Modeling and Analysis of Joint Swapping and Purification

Modeling E2E entanglement with hybrid swapping and purification in quantum networks is complex. Some works, like [13], [27], use tree structures for modeling swappings along quantum paths but do not include purification. Incorporating purification into the tree is easy as it also has two inputs and one output, but it is hard to perfectly absorb its properties. For example, the probability of purification failure is related to the fidelity of input pairs. When the tree is larger, how to compute the fidelity and expected entanglements used by this tree becomes challenging. Chang *et al.* [28] noticed that the associativity of operators helps to analyze the throughput on a single path, but does not consider fidelity. In this work, we use associativity to explain some properties of SPS. Jia and Chen [16]–[18] present the first closed-form fidelity of multiple purifications and optimal SPS for Werner systems. However, their analysis assumes no operation failure and narrowed hardware parameters, which limits its application. Instead, our analysis focuses on Binary system, considers operation failures, and allows more flexible hardware parameters.

III. Quantum Preliminary

In this section, we briefly review the preliminaries of quantum networks and the current status of quantum networking devices. We first introduce the noise models, which also explain why we use Binary and Werner systems in this paper. We then review the recent advancements in physical devices of quantum networks, and use the parameters of these devices to justify our parameter selection in our analysis and experiments.

A. Quantum States and Noise Models

A quantum bit can be in a superposition of multiple states, e.g., two states in a 2-state system. Such states can be expressed by $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers. This formula describes a state $|\psi\rangle$ that, upon measurement, turned to be $|0\rangle$ with probability $|\alpha|^2$, and to be $|1\rangle$ with probability $|\beta|^2$, where $|\alpha|^2 + |\beta|^2 = 1$. Similarly, when two qubits are entangled (called ebits), there are four possible states, so the state can be expressed by

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle. \quad (1)$$

The squared coefficients still indicate the probability of the results and they sum to 1, e.g., we may get $|00\rangle$ with probability $|\alpha_{00}|^2$. We are especially interested in four special states, which are called Bell states (also, interchangeably in this paper, EPR pairs): $|\Phi^{\pm}\rangle = \frac{1}{\sqrt{2}}(|00\rangle \pm |11\rangle)$ and $|\Psi^{\pm}\rangle = \frac{1}{\sqrt{2}}(|01\rangle \pm |10\rangle)$. We can express a pair of ebits not only by Eq. (1), but also by the Bell states. We are interested in expressing the state in the density matrix format:

$$\begin{aligned}\rho =& F_1 |\Phi^+\rangle\langle\Phi^+| + F_2 |\Psi^-\rangle\langle\Psi^-| \\ &+ F_3 |\Psi^+\rangle\langle\Psi^+| + F_4 |\Phi^-\rangle\langle\Phi^-|, \end{aligned} \quad (2)$$

where $\sum_i F_i = 1$. Note that ρ is not a vector like $|\psi\rangle$ in Eq. (1) but a matrix, and F_i's are the diagonal elements. Typically, we hope ρ to be a pure state of one of the four Bell states, e.g., without loss of generality, $|\Phi^+\rangle$. Then, the **fidelity** of an EPR pair is $F = F_1$, which means the probability that the EPR pair is in the desired state. In perfect systems, we have $F = 1$. In reality, however, the entanglements are always noisy, so we only have $F < 1$, and the other non-zero diagonals indicate the probability of being in the undesired states.

1) Quantum States: In this paper, we consider two formats of Eq. (2). For the *Binary state* [14], we define fidelity $F = F_1$ and allow only one of F_2, F_3, F_4 to be non-zero (i.e, $(1-F)$). E.g., in [14], $F_3 = 1 - F$. We are interested in such a state because it allows faster purification [14]. The other format is the *Werner state*, usually associated with the worst case noise: the entanglement is i) totally destroyed with probability p, or ii) left untouched with probability $1 - p$. Such noisy channels are also known as depolarizing channels. The result density matrix is $\rho = (1-p)|\Phi^+\rangle\langle\Phi^+| + p \cdot \frac{1}{4}I$, where I is the identity matrix and $\frac{1}{4}I$ represents the totally destroyed entanglement. Then, fidelity $F = F_1 = \frac{3}{4}p + \frac{1}{4}$ and $F_2 = F_3 = F_4 = \frac{1}{4}(1-p) = \frac{1}{3}(1-F)$. In the remaining, we are only concerned with the fidelity.

2) Noise Models: We consider noise introduced in two stages: entanglement generation and quantum operations (i.e., swapping and purification). We consider two settings of state representations and noise: (i) Binary states with noisy entanglement generation and noiseless quantum operations; (ii) Werner states with noisy entanglement generation and noisy quantum operations. We are interested in those two settings because the first setting allows faster purification protocols, while the second one represents the general (worst case) noise model. In the rest of this paper, we refer to these two settings by **Binary and Werner systems**. In other papers, the Binary/Werner states are sometimes called dephased/depolarized states, respectively.

B. Current Status of Network Devices

We now introduce the current status of quantum links, quantum memory, and quantum gates in quantum networks and the assumptions of them used in this work.

1) Quantum Links: To establish a pair of ebits between two parties (e.g., two nodes in a quantum network), optical links are usually used to distribute two entangled photons. In recent years, there have been numerous successful demonstrations of entangled photons distribution, such as [29]–[31]. The photon loss in fibers (the major reason of low entanglement rate) is exponentially negative to the length of the optical link,

$$q = e^{-\alpha L}, \tag{3}$$

where α is a constant determined by the medium (i.e., the optical fiber in this case), and L is the length of the optical link. When L is large, the successful probability can be excessively low. That is the reason why we should substitute a long link with multiple shorter ones. With many entanglements on shorter links, we can perform quantum swapping to connect them into a longer entanglement. Note that although the process of generating one single pair of entangled photons is probabilistic, our method operates in a time-slotted manner, where the number of successful entanglements can accumulate at each link, and we use its expectation to represent the available resource on each link. This assumption has been adopted in [10]–[12], [18].

To perform a swapping operation on a node to establish a longer entanglement, we have to have one ready pair of ebits at each side of this swapping node. If the attempt to distribute the pair of ebits at one side fails, we have to drop the successful one at the other side as we cannot store the photons. To tackle this problem, quantum memory [32] is needed. Equipped with quantum memory, the node can "store" (information carried by) a photon to wait for a successful attempt on the other side. In this way, we do not have to have two successful attempts at the same time: we can do asynchronous Bell state measurement (a core part of quantum swapping), resulting in much higher throughput [33].

2) Quantum Memory: Like its classic counterpart, quantum memory [32] can be used to store qubits. It can be achieved by many promising techniques, such as the nitrogen-vacancy (NV) center in diamond [34], trapped ion [35], and rare-earth doped solids [36].

Quantum memory is still under active development with many piratical issues need to be solved. For example, read and write operations may not be perfectly faithful: the results from readouts can be wrong. Reading the memory multiple times can reduce the probability of wrong results, but every readout makes the stored qubit more noisy, causing worse results. Despite these limitations, current works are already able to achieve readout fidelity as high as 0.9998 [33]. Therefore, in the near future, we can expect faithful memory in terms of read and write. Another critical problem of quantum memory is that the fidelity of stored qubits decreases as time elapses (due to noise). Thus, one may want to execute following operations as soon as possible. Such property makes the use of quantum memory even more complicated. Fortunately, various recent works have achieved considerably long-time coherence, from seconds [35] to minutes [37], and hours [38].

In this paper, we consider faithful memory and focus on the network-wide connection establishment.

3) Quantum Gates: For swapping and purification operations, quantum gates are used. The most important one is the Bell State Measurement (BSM) gate.

Linear optic-based BSM is a popular realization because of its efficiency. On the other hand, it suffers from low success probability. Classic linear optic-based BSM is not able to achieve a rate higher than 50% as it is only able to distinguish two of the four Bell states. This fundamental drawback makes it hard to improve its performance. Further improvements can achieve higher rates, such as 62.5% [39] and 75% [40], at the price of auxiliary single or entangled photons and lower efficiency. While more auxiliary resources lead to higher success probability (as high as 100% in theory), the gates are also less practical (more costly and less efficient). Solid state-based gates such as [33] can offer deterministic BSM (no failure). In this paper, we assume the success probability is in $(0.5, 1)$ for flexibility.

In addition to success rate, fidelity is another metric for quantum gates. Wang *et al.* [41] demonstrated a BSM that is potential to achieve fidelity as high as 99.2%. Besides, [42] reports their high-fidelity universal quantum gates. They achieved 0.999952 fidelity for 1-qubit gates and 0.992 for two-qubit gates. That is, we can expect that the BSM gates can be reliable enough in the future. In this paper, we assume that high-fidelity gates (e.g., fidelity between 0.99 and 0.99999), including BSM gates, will be available in the near future. We set the parameters in the simulation section accordingly.

IV. SPS Modeling and OSPS Problem

In this section, we first provide the modeling of swapping and purification operations and define the studied OSPS problem and SPT. Then we introduce a dynamic programming based solution to OSPS. Last, we also present a corresponding network-wide scheduling problem aim to maximize the throughput based on the solution of OSPS.

A. Swapping and Purification

Now, we look at the two fundamental building blocks [11]: *swapping* and *purification*.

1) Swapping: Swapping, or more generally, teleportation, involves sending a quantum state to a remote node with the help of an EPR pair. When the qubit to be teleported is already

entangled with another qubit, we call it a swapping: two entanglements are connected into a longer one. The density matrix of the output entanglement can be expressed by the diagonals of the two input states. A detailed calculation is given by formula (10.9) in [43]. In this paper, we are only interested in the output fidelity (the first diagonal element), $F_1^{out} = \sum_i F_i^1 F_i^2$, where F_i^1/F_i^2 are the diagonals of the first/second input pair. It is easy to see that the output fidelity is the probability that the two input pairs are in the same states, no matter in the desired one ($F_1^1 F_1^2$) or not ($\sum_{i=2,3,4} F_i^1 F_i^2$). The formula in [43] gives us the most general case as it allows all diagonals to be independent. When bringing the constraints of those terms in Binary/Werner states, we obtain their swapping functions S_b (in Binary systems) and S_w (in Werner systems). Suppose the fidelity of the two input entangled pairs is f_1 and f_2, separately; then the fidelity of the output pair can be modeled as follows

$$S_b(f_1, f_2) = f_1 f_2 + (1 - f_1)(1 - f_2), \tag{4}$$

$$S_w(f_1, f_2) = f_1 f_2 + \frac{1}{3}(1 - f_1)(1 - f_2). \tag{5}$$

As we have mentioned in Section II, the *success probability of swapping* does not rely on the input fidelity, but its implementation (e.g., 0.5 for linear optics, 0.625 [39] or 0.75 [40] for improved ones, and 1.0 for deterministic BSM [34]). We allow it in $(0.5, 1)$ in this paper.

2) Purification: We consider the two most popular and well studied purification protocols: DEJMPS [14], [44] and BBPSSW [45]. DEJMPS is efficient in terms of improving the output fidelity when two input pairs are in the Binary state, while BBPSSW requires the input states in Werner state (usually associated with the worst case noise). This is also the reason why we consider Binary/Werner systems. Note that the output of BBPSSW may not be in the Werner state, but we can easily bring it back by depolarization [45] [46].

Purification consists of two CNOT gates and a Bell state measurement (BSM). The two CNOT operations are conducted by using one of the input pairs as control qubits, the other pair as the target qubits. When the BSM result on the target pair agrees (i.e., the two qubits are in the same state), the purification succeeds and the other input pair is left with higher fidelity. Otherwise, it fails and both pairs are discarded without any outcome. Such inevitable failures are exactly one of the major reasons why it is not easy to find a good SPS. By the definition of the two purification protocols, the fidelity of their output pairs can be modeled as

$$P_b(f_1, f_2) = \frac{f_1 f_2}{f_1 f_2 + (1 - f_1)(1 - f_2)} = \frac{f_1 f_2}{f_1 f_2 + \bar{f}_1 \bar{f}_2}, \tag{6}$$

$$P_w(f_1, f_2) = \frac{f_1 f_2 + \frac{1}{9}\bar{f}_1 \bar{f}_2}{f_1 f_2 + \frac{1}{3}[f_1 \bar{f}_2 + \bar{f}_1 f_2] + \frac{5}{9}\bar{f}_1 \bar{f}_2}, \tag{7}$$

where $\bar{f}_i = 1 - f_i$. The denominators in these formulas are the probability that the result of BSM agrees, which is exactly the *success probability of purification*. The numerators are the first diagonals obtained from the protocols, respectively. For example, in Binary systems, the success probability is when both pairs are in the desired state ($f_1 f_2$) or both not ($\bar{f}_1 \bar{f}_2$), and only ($f_1 f_2$) contributes to the first diagonal according to [14], [44]. It is similar to that of [45].

3) Noisy Gates: The above assumes that the gates are noiseless. When they are noisy, the output fidelity of swapping and purification is affected by the accuracy of used gates. Typically, three types of gates are involved: 1-qubit gates, 2-qubit gates, and BSM gates. Formulas including the impact of these three types of gates can be found in Supplementary Material and [15], [44], [47]. We will use them in our evaluations for noisy Werner systems.

B. OSPS Problem and Tree-based Modeling

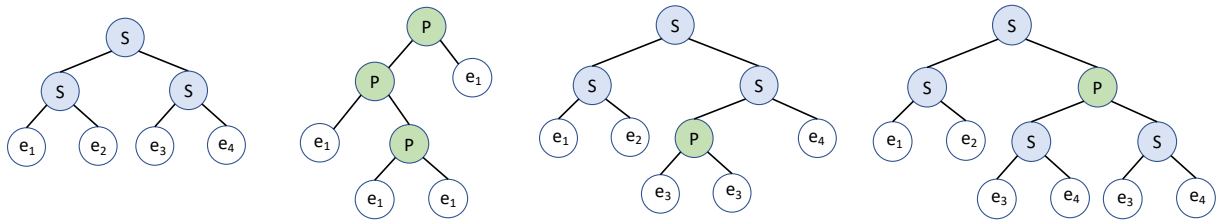
1) OSPS Problem: We now formally introduce the *Optimal Swapping and Purification Scheme* (OSPS) Problem on a quantum path between an SD pair.

Definition 1: **OSPS Problem**: Given a simple quantum path $n_1 \leftrightarrow n_N$ with $N-1$ edges/hops $\{e_1, e_2, ..., e_{N-1}\}$ where the edge e_i connects nodes n_i and n_{i+1} and generates a limited number of entanglements in a time period, how to use swapping and purification operations to create an E2E entanglement between the SD pair nodes of n_1 and n_N so that (1) the fidelity of generated entanglement is larger or equal to a fidelity threshold F^*; (2) the expected number of original link-level entanglements used is minimized.

2) Swapping and Purification Tree: Given a quantum path, a Swapping Scheme (SS) defines the order in which quantum swaps are performed to establish an end-to-end entanglement along the path. Similar to [27], we can model any SS using a tree structure, called a *Swapping Tree* (ST). By ignoring purification in the SPS shown in Fig. 1, we can model the swapping scheme along the path $n_1 \leftrightarrow n_5$ in both SPS schemes, whose corresponding ST is the same and shown in Fig. 2(a). Similarly, we can model purification schemes using a *Purification Tree* (PT), as shown in Fig. 2(b). It demonstrates a PT over the link e_1. A slight difference is that a PT only involves entanglements on the same two nodes. When both swapping and purification are used along the path, we can model the SPS using a *Swapping and Purification Tree* (SPT), as shown in Fig. 2(c) and Fig. 2(d) which depict Scheme 1 and Scheme 2 in Fig. 1, respectively. It is clear that both ST and PT are special SPTs.

3) Fidelity and Cost of SPT: For any of these types of trees, we can compute the fidelity and expected cost of any entanglement (i.e., the expected number of entanglements needed) generated on the corresponding node in the tree in a bottom-up manner. It is obvious that the fidelity can be calculated in a (bottom-up) layer-by-layer fashion using the equations defined in Section IV-A. We now show that the **expected cost** can also be calculated by a layer-by-layer method in the tree. Note that (1) if all operations never fail, then the cost is simply the number of leaves in the tree (i.e., the number of entanglements used in total); (2) if we consider the success probabilities of swapping or purification operations, we now use a simple example of three-level tree (as shown in Fig. 3) to illustrate that the expected cost at the root (node A), denoted by $\mathbb{E}[c_A]$, can be calculated as follows:

$$\mathbb{E}[c_A] = \frac{\mathbb{E}[c_B] + \mathbb{E}[c_C]}{p_A} = \frac{\frac{\mathbb{E}[c_D]+\mathbb{E}[c_E]}{p_B} + \frac{\mathbb{E}[c_F]+\mathbb{E}[c_G]}{p_C}}{p_A}, \tag{8}$$



(a) ST for Scheme 1/2 (b) PT for link e_1 (c) SPT for Scheme 1 (d) SPT for Scheme 2

Fig. 2: Modeling of different trees: (a) the swapping tree (ST) of Scheme 1 and 2 (ignoring purification); (b) an example purification tree (PT) over link e_1; (c)/(d) the swapping and purification tree (SPT) of Scheme 1/2.

where c_* and p_* represent the number of entanglements used by the operation on node $*$ and the success probability of the operation on node $*$. The expected cost of a parent node is the ratio between the summation of the cost of its two children (because they are both consumed in the operation) and the successful rate of the operation (i.e., swapping or purification).

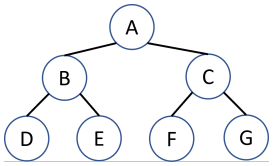


Fig. 3: A 3-level SPT: nodes A, B, C are either swapping or purification, while leaves can be quantum links or sub-SPTs.

A proof of Eq. (8) when the leaves (nodes D, E, F and G) are quantum links can be derived as

$$\begin{aligned}
&\mathbb{E}[c_A]\cdot p_A\\
=&\sum_{u,v=0}^{\infty}\big[(1-p_B)^u p_B(1-p_C)^v p_C\cdot((c_D+c_E)(u+1)\\
&+(c_F+c_G)(v+1))\big]\\
=&\sum_u\sum_v(1-p_B)^u p_B(1-p_C)^v p_C[(c_D+c_E)(u+1)]\\
&+\sum_u\sum_v(1-p_B)^u p_B(1-p_C)^v p_C[(c_F+c_G)(v+1)]\\
=_a&\frac{c_D+c_E}{p_B}+\frac{c_F+c_G}{p_C},
\end{aligned}$$

where u and v are the numbers of failed operations of the left and right branches, separately. This is a combination of two geometric distributions from the two children, separately: both of their operations may fail, and we need them to succeed at least once to proceed. Step $=_a$ is similar to calculating the expected cost of the geometric distribution.

By applying this three-layer tree recursively towards the root from the bottom, we can process the whole SPT and calculate the expected cost of this SPT in a layer-by-layer way.

C. Dynamic Programming Solution

Based on the tree modeling, we now introduce a dynamic programming (DP) solution for the dual problem of our OSPS problem where the cost C (the number of entanglements needed) is given and the optimization goal is to maximize the E2E fidelity. We call such a problem the *optimal fidelity SPS* (OFSPS). Note that the DP solution for OFSPS has the same issue as the early hybrid scheme [14]: the cost needs to be known as input, which makes them not useful when failure is considered. To use DP to solve our OSPS problem, we have to try excessive times (for various costs) to find a proper cost. Moreover, DP cannot solve the OFSPS with operation failures. However, the DP solution still helps us to understand what the optimal SPS should be like in our OSPS problem.

We now consider the optimal structure of OFSPS. Let $T(i,j,c)$ denote the optimal fidelity through link e_i to link e_j (inclusive), when the total *expected* budget is c EPR pairs. Then, we have:

$$T(i,j,c)=\max\begin{cases}\max\limits_{\substack{i<m<j\\ m-i\le k\le c-j+m}} S(T(i,m,k),T(m,j,c-k)),\\ \textit{[via swapping at node } n_m\textit{]}\\ \max\limits_{j-i\le k\le c-j+i} P(T(i,j,k),T(i,j,c-k)).\\ \textit{[via purification between } n_i \textit{ and } n_j\textit{]}\end{cases}$$

This considers the current path fraction as $n_i \leftrightarrow n_j$ with a budget of c. Whenever we get a new entanglement pair (except link-level ones), it is generated from either quantum purification (1st term in the eq.) or swapping (2nd term in the eq.) operations. By exploring all possible path fractions and budget splits, a simple bottom-up DP algorithm can compute the final optimal fidelity $T(1,N,C)$ for OFSPS.

Optimality. Under the assumptions of [11], [16]–[18] (no failure operation, actually impossible for purification), this DP algorithm is able to find the optimal solution, as DP algorithms exhaust all possible integer budget splits. However, if failure of operations is considered, the expected cost can be real numbers. As there are infinite possible splits for real numbers, the DP algorithm is not able to try all splits, and thus cannot find a feasible solution for OFSPS (and OSPS).

D. Network-wide Scheduling Problem

In our formulated OSPS problem, we only focus on optimization over a single quantum path between an SD pair. If we would like to consider a network-wide scheduling problem, we can define a network scheduler similar to the one in [11]. It is essentially an optimization problem to maximize the network throughput (served E2E entanglement) under resource constraints (link and memory capacity) in the whole network, where the path-level results of OSPS are used as inputs.

Assume that in the network $G(V,E)$, there cloud be an entanglement demand $d_{i,j}$ between any SD pair (i,j). We use $P_{i,j}$ to denote the path set of all quantum paths between nodes

i and j. Each node $v \in V$ has a memory limit m_v and each link $e \in E$ has a entanglement capacity k_e. Then the overall scheduling problem can be defined as follows:

$$\max_{x} \sum_{i,j \in V, p \in P_{i,j}} x_{i,j,p} \tag{9}$$

s.t. entanglement distribution constraints:

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

resource constraints:

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

$$\sum_{i,j \in V, p \in P_{i,j}, v \in p} \beta_{i,j,p,v} x_{i,j,p} \leq m_v, \qquad \forall v \in V \tag{12}$$

decision variables:

$$x_{i,j,p} \in \mathbb{Z}_0^+ \tag{13}$$

Here, the decision variable is $x_{i,j,p}$, denoting the number of entanglements served for the SD pair (i,j) via the path p. The objective (9) is maximizing the total number of entanglements distributed in the whole network. Constraint (10) ensures that we do not serve each SD pair with more than demanded $d_{i,j}$ entanglements. The served entanglements for each pair (i,j) may come from any path $p \in P_{i,j}$ between between them, so the total number of served entanglements is summarized over all paths. Constraint (11) limits the total number of used entanglements on each edge does not exceed its capacity k_e. Here, $\alpha_{i,j,p,e}$ is the entanglements consumed on edge e by the path p between pair (i,j). We can obtain $\alpha_{i,j,p,e}$ by the path level solution of each path p via our algorithm for OSPS before the network-wide scheduling process so it is an input. Constraint (12) limits the number of used memory slots on node v not to exceed the available m_v slots. Similar to $\alpha_{i,j,p,e}$, $\beta_{i,j,p,v}$ is the number of memory slots consumed on node v by the path p between pair (i,j), which is also obtained from the path level solution. We solve this network-wide optimization in our simulations to obtain the network throughput under fidelity requirements, presented in the evaluation section.

V. Analysis of Optimal SPS in Binary Systems

To facilitate better solutions for our OSPS problem, we first analyze the optimal SPS in Binary systems.

1) SPS Tree Patterns: We introduce the definitions of two tree patterns: *optimal SPT* (OSPT) and *restricted SPT* (RT).

Definition 2: The *optimal SPT* (OSPT) is an SPT with the least resource cost of the root node, while the root fidelity is no less than a given requirement F^*.

Obviously, OSPT describes the optimal solution of OSPS.

Definition 3: A *restricted SPT* (RT) is an SPT in which there is no swapping node below a purification node on any path from the root to a leaf.

In an RT, no purification should happen on any pairs generated after one or more swappings. All entanglements participating in any purification either come from (i) entanglement generation on optical links, or (ii) previous purification.

2) Main Theorem on Optimal Tree Pattern: We prove that any OSPT in Binary systems must be an RT.

Theorem 1: In Binary systems, when $F^* \in (0.5, 1)$, any OSPT of the OSPS problem must be an RT.

Proof: We prove this by contradiction. Assume that there exists an OSPT that is not an RT. Then we show that there exists a transformation that can transform this OSPT into an RT while increasing its root fidelity and decreasing its cost. This is done by recursively transforming subtrees of this OSPT to RTs in a bottom-up way. Finally, the whole OSPT is transformed into an RT which has higher fidelity and less cost than the OSPT, leading to the contradiction.

Now we show how to conduct such transformations. When the subtrees are limited to RT and PT, there are six distinct internal node patterns, as shown in Fig. 4. We use three letters (for 'the root'-'left subtree'-'right subtree') to name these patterns. These six patterns are universal because (1) there are $2^3 = 8$ trees in total, but exchanging left and right subtrees keeps the tree identical, so only 6 out of 8 are left; (2) RT includes PT, but we assume that any RT in those patterns are not PT. That is because, in those cases, the pattern degrades to one of the others. For example, if one of the RTs in SRR is a PT and the other is not, then the SRR is actually an SRP.

First, for four patterns (SRR, SRP, SPP, and PPP), the overall tree is already an RT, so we only need to handle the other two cases (PRP and PRR). Second, in PRP (Fig. 4(e)), the right child is a pure PT over one link. As the left RT child can perform the purification operation with this right PT, they must share the same end nodes (i.e., on the same two adjacent routers). That is, this RT must be either the leaf that the PT purifies or another PT over this leaf. Note that a potential counter example is when the RT forms a ring with the leaf: the RT is not a PT but still has the same ends as the leaf's. But this is impossible because SPT is defined on a non-cyclic simple path. Thus, this PRP is actually a PT without any swapping, thus it is a PPP, an RT too. Last. for PRR (Fig. 4(f)), we want to prove that it can be transformed into an SRR with better fidelity and cost. In a Binary system, we can formally prove this as Lemma 2. Generally, Lemma 2 tells us that we can lift the two 'S' nodes in PRR one level upwards, without violating its optimality.

Till now, we can transform the above 6 patterns into RT without decreasing the fidelity or increasing the cost. Now, we explain how to transform any SPT into an RT. For a non-RT SPT, we can process it in a bottom-up way: we find the shallowest 'S' node whose parent is 'P' (so a PRR). It must exist otherwise the SPT is an RT. And the children of this 'S' must be RT as it is the shallowest 'S'. We can lift the 'S' nodes to transform the subtree into an RT by applying Lemma 2. By repeating this transformation, we can concentrate all 'S' nodes towards the root. Finally, we can transform any SPT into an RT with increased fidelity and decreased cost. This contradicts the fact that the OSPT is optimal. ■

Lemma 2: In Binary systems, when all nodes' fidelity $f \in (0.5, 1)$, any PRR can be transformed into SRR with increased fidelity and reduced cost.

Proof: The detailed proof is provided in Supplementary

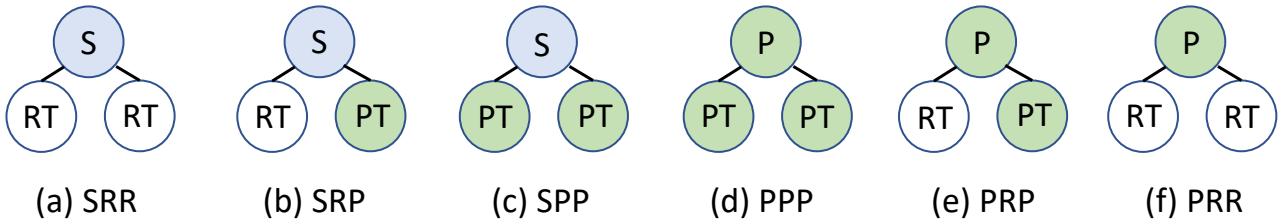


Fig. 4: Six possible SPS tree patterns. Here, the pattern is named by 'the root'-'left subtree'-'right subtree', where 'the root' can be swapping (S) or purification (P), and 'subtree' can be either a pure PT (P) or an RT (R).

Material, and we explain its basic idea here. We can quantify the fidelity and cost before and after the transformation via our layer-by-layer processing. Then, we get a set of two inequalities: one states that the cost is smaller after the transformation; the other states that the fidelity is higher after the transformation. We can find that the necessary and sufficient condition for both inequalities is that the fidelity of children in $(0.5, 1)$, which is reasonably true in Binary systems[1]. ■

3) Discussions: From Theorem 1, we can draw two main conclusions for Binary systems. First, to search for OSPS, we no longer need to consider all SPSes (i.e. all SPTs) but only focus on those of RTs. This greatly reduces the complexity of its search. Second, for the design of protocols and devices (e.g., routers), we may directly integrate the purification at the link level, which separates the swapping and purification. The protocol may send a requirement for link-level fidelity, then the routers of a link collaborate locally to generate such entanglements while the protocol can arrange the swapping. Such a design may greatly reduce the complexity of network-wide fidelity-aware protocols (compared with hybrid SPS).

Besides, because RT is an ST of PT (take the root nodes of PTs as leaves of the ST), we can propose an intuitive solution framework (for the OSPS problem) consisting of three stages: (i) PT shape search; (ii) ST shape search; (iii) sacrificial pair allocation. Existing works [11], [12], [23] do not do shape search because S_b, S_w, P_b (i.e., Eq. (4), Eq. (5), Eq. (6)) are all associative. That is, the orders of (i) swapping in Binary systems, (ii) purification in Binary systems, and (iii) swapping in Werner systems do not affect the final fidelity of an SPS. But *when the failure of operations occurs, the order now matters in both systems, as the expected cost is not associative anymore*. Theorem 1 also implies that combining the optimal solution for each stage can lead to the optimal solution overall. However, finding the optimal PT and ST shapes is still challenging. Existing works [11], [12] both somehow follow this 3-stage framework. However, since Theorem 1 is not true for Werner systems, such a framework may not even be able to generate a feasible solution. We will confirm that [11], [12] are indeed empirically infeasible in the evaluation section.

VI. Proposed Branching Tree Algorithm

Based on our previous analysis, we now propose a new tree-based SPS algorithm (TREE) that can efficiently find a feasible SPS solution over a quantum path in both *Binary systems* and *Werner systems*.

[1]The fidelity of any node in a SPT in Binary systems should always be in $(0.5, 1)$. Since if a node has $f \leq 0.5$, both swapping and purification will propagate this low fidelity towards the root to make the root's fidelity no greater than 0.5 which is useless in a Binary system.

Algorithm 1 Branching Tree SPS (TREE)

Input: Quantum path consisting of nodes $n_1 \leftrightarrow n_N$, and its swapping probability array $\mathbf{p} = \{p_i | i \in \{1, ..., N\}\}$ and edge fidelity array $\mathbf{f} = \{f_{(n_i, n_{i+1})} | i \in \{1, ..., N-1\}\}$, target E2E fidelity threshold F^*.
Output: final E2E fidelity f, (root of) SPT t.

$f, t =$ Build-ST($\mathbf{p}, \mathbf{f}$) ▷ initial ST t with root fidelity f
while $f < F^*$ **do**
  Grad(t) ▷ forward gradient from root to each node
  Calc-Efficiency(t)
  $n =$Find-Max-Efficiency(t) ▷ find the most eff. n
  $n =$ Purify(n) ▷ purify and update n
  $f, t =$ Backward(n) ▷ update nodes backwards
return f, t

1) Overall Framework: We first introduce the main workflow of our proposed algorithm. The basic idea is, with the help of the tree structure, we can repeat the following loop until the E2E fidelity is high enough: (i) find a proper node to purify; (ii) purify the node; (iii) update the tree. See Algorithm 1. First, Build-ST() constructs an ST of the router chain without any purification. Then, the loop starts, and Fig. 5 shows an example of flow within a single loop.

1) Grad() calculates the gradients of each node w.r.t. the root. The gradients consist of two parts: (i) gradient of fidelity G_f, (ii) gradient of cost G_c. They are used to estimate how any purification impacts the root node, in terms of fidelity and cost increase.
2) Calc-Efficiency() estimates the efficiency of purifying each node, i.e. (increased fidelity)/(increased cost).
3) We find the node with the highest efficiency and conduct purification at that node.
4) Finally, as the purified node (with its subtree) is replaced by a new node, we call Backward() to update this information to all its ancestors till root.

One of the advantages of this gradient-based design is its efficiency. By utilizing gradients, we can estimate the impact of purifying any node locally: we do not need to go backward (to the root) to know the impact on the root node. Because all functions inside the loop are $O(M)$ (where M is the number of nodes in the tree), we can do one purification in $O(M)$ time. Otherwise, without the gradients, we have to backtrack to root for every node, resulting in $O(M^2)$. Note that we do not provide an analytical relation between M and N or time

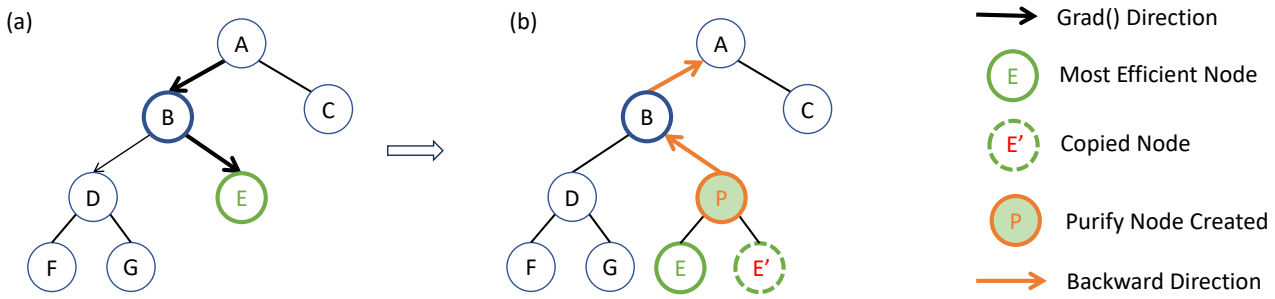


Fig. 5: Control flow of TREE in one single loop: (a) shows gradient calculation and node finding prior to purification; (b) shows the purification followed by ancestors updating.

complexity of the whole algorithm, which is left as our future work. Instead, we preform empirical study to compare running time of all methods in the evaluation section.

2) Key Steps: we now provide more details of each key step in our proposed algorithm.

BUILD-ST(). It simply creates an ST for the given links in the quantum path. The shape of ST is a factor in the performance of this algorithm, but again, we leave the shape search for ST as future work. Currently, we simply build a balanced tree, but any other swapping tree method can be used.

GRAD(). Note that all swapping/purification functions, both for fidelity and cost (probability), are (compositionally) differentiable. We can compute the gradient of any node w.r.t. to the root, according to the chain rule (for calculating derivatives). It is similar to the computational graph design (auto-grad) in machine learning frameworks (like PyTorch, TensorFlow, etc).

Each node should maintain two gradients: *the gradient of fidelity* ($G_f(*)$) and *the gradient of cost* ($G_c(*)$), where the $*$ is the corresponding node. $G_f(*)$ is relatively easy to compute because a node's fidelity is only determined by the fidelity of its children. However, $G_c(*)$ is determined by both the fidelity and the cost of its children for the purification operation. Therefore, $G_c(*)$ for a purification node (denoted by $G_c^P(*)$) is actually a vector of two partial gradients, i.e., $G_c^P(*) = (G_{c,f}^P(*), G_{c,c}^P(*))$. For the swapping operation, the cost gradient $G_c^S(*)$ is only determined by the cost of its children, so we set $G_{c,f}^S(*) = 0$. Thus, for either swapping or purification, we have three gradients to calculate, resulting in six gradients in total.

Suppose that we are calculating the gradients of node E in Fig. 5 (when the algorithm reaches its parent node B), we use

$$G_f^S(E) = \frac{\partial S(f_D, f_E)}{\partial f_E} \cdot G_f(B), \tag{14}$$

$$G_{c,c}^S(E) = \frac{\partial S(c_D, c_E)}{\partial c_E} \cdot G_{c,c}(B) = \frac{G_{c,c}(B)}{Pr\{S\}},\ G_{c,f}^S(E) = 0, \tag{15}$$

$$G_f^P(E) = \frac{\partial P(f_D, f_E)}{\partial f_E} \cdot G_f(B), \tag{16}$$

$$G_{c,c}^P(E) = \frac{\partial P(f_D, f_E, c_D, c_E)}{\partial c_E} \cdot G_{c,c}(B),$$

$$G_{c,f}^P(E) = \frac{\partial P(f_D, f_E, c_D, c_E)}{\partial f_E} \cdot G_{c,f}(B). \tag{17}$$

Based on the actual operation (swapping or purification) of E's parent node B, we use either Eq. (14, 15) or Eq. (16,17). Note $G_f(B)$ is the fidelity gradient of parent node B. When B is not the root, $G_f(B)$ is already calculated when the algorithm reaches B's parent node. If B is the root, its gradient is set to 1. For Eq. (15), the cost of swapping operations is calculated by Eq. (8), thus $\frac{\partial S(c_D,c_E)}{\partial c_E} = \frac{1}{p_B} = \frac{1}{Pr\{S\}}$.

When estimating the cost increase at the root A (caused by this purification at E), $\Delta_c(A) = (\Delta_f(E), \Delta_c(E)) \cdot (G_{c,f}(E), G_{c,c}(E))$, where $\Delta_f(E)$ and $\Delta_c(E)$ are the fidelity and cost increase after purifying node E, both of which can be locally computed. With the help of gradients, we are able to estimate the impact of an operation on the root, without tracing back to the root.

According to the chain rule, we can repeat such differentiation for both functions from the root to all leaves. Then, for CALC-EFFICIENCY(), the efficiency of node E is

$$\delta(E) = \frac{\Delta_f(A)}{\Delta_c(A)} = \frac{\Delta_f(E) \cdot G_f(E)}{(\Delta_f(E), \Delta_c(E)) \cdot (G_{c,f}(E), G_{c,c}(E))}. \tag{18}$$

PURIFY(). To purify a node, for example, E, we need to (i) make a copy of this node, say E', (ii) create a new purification node P whose children are E and E', (iii) replace the node E by P in the tree.

Finally, BACKWARD(). Suppose we have just performed PURIFY for a node (or a subtree), now we need to update all ancestors of the new node P. The red arrows in Fig. 5 show the trace of BACKWARD(). We backtrack from node P until the root. Whenever reaching a new ancestor, we re-calculate its fidelity and cost using the new information from its children.

In the next round, if we hope to do another purification, GRAD() and CALC-EFFICIENCY() will update the gradients and efficiencies of all nodes. The loop is repeated until the E2E fidelity reaches the threshold.

3) Analysis: An implicit property of the TREE method is that its output SPT is always an RT in Binary systems. This property is favorable because, though not all RTs are optimal, they include optimal trees in Binary systems (by Theorem 1).

Theorem 3: For Binary systems ($f \in (0.5, 1)$), TREE method never purifies a SWAP node and generates an RT.

Proof: Note that if swapping nodes are never purified, we cannot get a non-trivial PRR pattern in the tree so it is an RT; if the tree is not an RT, there must be at least one non-trivial

PRR, which implies that a swapping node is purified. Here, we briefly introduce how to prove that the swapping node cannot be more efficient than its two children at the same time, while the detailed proof of this step is provided in Supplementary Material. We can instantiate Eq. (18) for a swapping node and its two children. Then, we build a set of two inequalities: one states that the parent's efficiency is greater than the left child's; the other states that the parent's efficiency is greater than the right child's. Solving the set of inequalities, we can see that it requires $(f_l - \frac{1}{2})(f_r - \frac{1}{2}) > \frac{1}{4}$ or $(f_l - \frac{1}{2})(f_r - \frac{1}{2}) < -\frac{3}{4}$, where f_l/f_r is the fidelity of the left/right child. Since both f_l and f_r are in $(0.5, 1)$, the inequality set can never be satisfied. That is, the efficiency of any swapping node cannot be greater than its two children at the same time, so the TREE algorithm never purifies a swapping node (but its children or deeper descendants). ■

VII. Evaluations

To evaluate the proposed method and the baselines, we have conducted experiments for both Binary and Werner systems under different operations (gates) noise levels.

A. Experiment Setup

1) Default Network Settings: Each SD pair requires no more than 10 entanglements. Node memory and link capacity are as specified in Table III. For Binary systems, link fidelity is randomly drawn from $(0.7, 0.95)$ based on [11]. For Werner systems, we set it drawn from $(0.95, 1)$, since it is harder to maintain connections in Werner systems. The number of candidate paths between any SD pair is set to 5. The accuracy (or success probability) of quantum gates is specified in Table II, set based on current/near future achievable devices [34], [39]–[42]. In the table, α_1, α_2, η represent the fidelity of 1-qubit, 2-qubit, and BSM gates used for noisy Werner systems. For network topology, we use three types of network size summarized in Table III. $AT\&T$ is a real topology from [48], $G(n, p)$ and $P(n, m)$ are Erdos-Renyi graphs with parameters n/p and preferential attachment model with parameters n/m, respectively. The two random networks are generated by the *networkx* [49] library. These networks/models are used in [10].

2) Baselines: We compared five different SPS methods (three existing methods and two of our proposed methods), which can be grouped into two groups.

Group 1 includes methods designed for OSPS, i.e., accepting the E2E fidelity threshold as input and minimizing the used entanglements to reach the fidelity threshold.

1) **TREE**: our proposed branching tree method in Sec. VI.
2) **GRDY**: a greedy algorithm [12] that greedily selects the link with the highest final fidelity increase.
3) **EPP**: an entanglement path preparation method [11] that greedily finds the link with the largest fidelity gradient.

Group 2 contains methods for OFSPS, i.e., needs a predetermined budget and aims to find the highest achievable fidelity. Note that the real used cost may be higher than the budget (because these methods do not consider failures), and some may not be able to generate a feasible solution for OSPS.

4) **DP-x**: our proposed DP solution in Section IV-C with a input budget of x times the cost given by TREE method. The x's in all evaluations are selected to make DP have similar cost and resulting fidelity as TREE's for comparison.
5) **NESTED**: the nested (hybrid) purification method from [14], which is based on a fixed SPS and requires the total budget should be $(ly)^z$, where l is the fragment length, y is the budget for each link per level, and z is the nest level. Here, we decide the best l, y, and z according to the path length and budget from TREE method. The calculated y can be fractional, so we use both its floor and ceiling (denoted as NESTED-F and NESTED-C).

Note that we do not compare with the methods from [18] because they do not consider operation failures and it is non-trivial to calculate their methods' real expected cost under failures. The source code of our algorithm and the baselines is publicly available at https://github.com/Jiyao17/sptree.

3) Performance Metrics: To evaluate the proposed methods and baselines, we focus on their competency in providing high-fidelity entanglements. We compare their (i) cost (number of sacrificial entanglements) required to reach specific fidelity on a single path; (ii) impact on network-wide throughput under fidelity requirements, as defined in Section IV-D. For a fair comparison, this optimization problem is used for all baselines as their network scheduler.

TABLE II: Device parameters for Werner and Binary systems.

Setting	α_1	α_2	η	Pr$\{S_w\}$	Pr$\{S_b\}$
P	1	1	1	1	1
D	$1-10^{-5}$	$1-10^{-5}$	$1-10^{-4}$	$1-10^{-4}$	$1-10^{-4}$
H	$1-10^{-5}$	$1-10^{-5}$	$1-10^{-4}$	0.75	0.75
L	$1-10^{-3}$	$1-10^{-3}$	$1-10^{-2}$	0.5	0.5

P: perfect (noiseless); H/L: high/low reliability; D: w/ in-memory BSM.

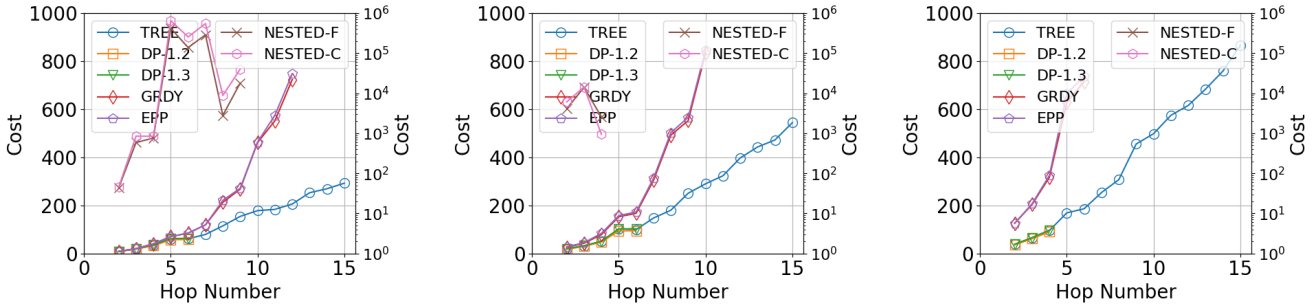
TABLE III: Network description.

Scale	SD #	Topology	$\lvert V\rvert$	$\lvert E\rvert$	m_v	k_e
Small	13	$AT\&T$	25	104	10	5
Medium	25	$G(50, 0.1)$	50	121	30	15
Large	50	$P(100, 2)$	100	196	50	25

B. Evaluation Results

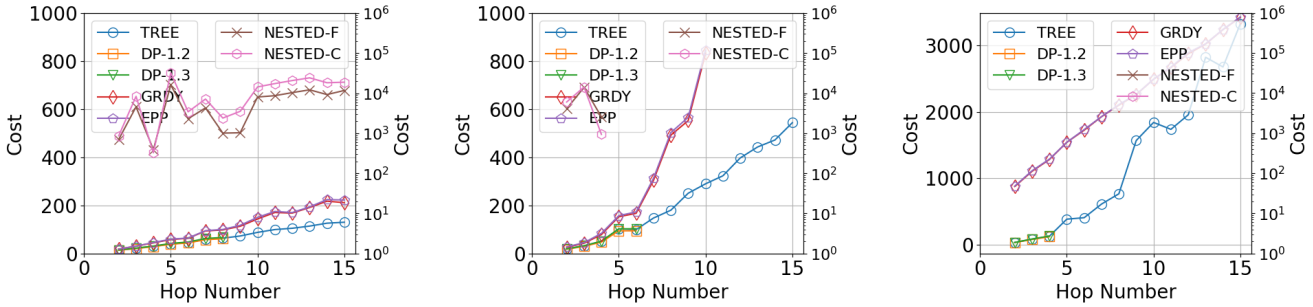
We first report the results on path efficiency in both Binary and Werner systems, then show their network throughput performances. Results are the average of 20 separate runs.

1) Path Efficiency in Binary System: We now investigate how well the algorithms can establish one single E2E entanglement on one single path. Fig. 6 provides the results of all methods in the Binary system under different fidelity thresholds. NESTED methods (and sometimes other methods) typically use significantly more entanglements compared to TREE and DP, so they are drawn in a logarithmic scale (right y-axis). For clarity, we limit the right y-axis to 10^6, so NESTED is sometimes cut-off as its cost is too large on a long path. We can observe the following. First, to generate E2E entanglement on a longer path, more entanglements (i.e., larger cost) are needed. Second, for higher fidelity requirement F^*, a larger cost is needed. Third, for all cases (different F^*), our method TREE uses the least number of sacrificial entanglements (except the tuned DP methods, explained later), which implies it can achieve higher network-wide throughput when employed by a network-wide scheduler. Last, when F^* is closer to 1, the more entanglement resources our method saves



(a) $F^* = 0.9$ (b) $F^* = 0.99$ (c) $F^* = 0.9999$

Fig. 6: Path efficiency (cost) of all methods with different F^* in Binary systems ($\Pr\{S_b\} = 0.75$).



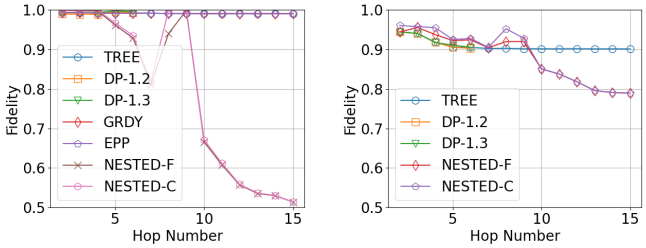
(a) $\Pr\{S_b\} = 1$ (b) $\Pr\{S_b\} = 0.75$ (c) $\Pr\{S_b\} = 0.5$

Fig. 7: Path efficiency (cost) of all methods under different device reliability levels in Binary systems ($F^* = 0.99$).

compared with GRDY, EPP and NESTED. Although GRDY and EPP follow the PS strategy, they do not perform well since they simply use sequential swapping and purification. While TREE is agnostic to the SPS schemes, it automatically follows the PS, and conducts shape search for PTs. The zigzags of the two NESTED methods are caused by their factorization strategy: they perform much better when the path length can be written in the powers. e.g., $4 = 2^2, 8 = 2^3$, etc. In addition, the two NESTED methods are not shown in Fig. 6(c) because they need more than 10^6 entanglements for all path lengths.

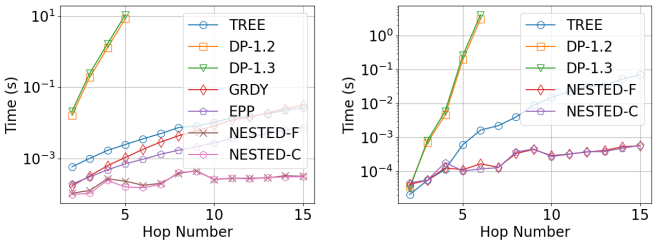
Fig. 7 shows the results with different reliability levels when $F^* = 0.99$. We have three reliability ($\Pr\{S_b\}$) levels (P, H, L) for Binary systems as shown in Table II. Obviously, lower reliability leads to larger costs. Fig. 8(a) also reports the achieved E2E fidelity when $F^* = 0.99$ and device level is H. As shown in Figs. 6,7,8(a), NESTED methods consume much more entanglements than TREE does, while still cannot generate comparable E2E fidelity in most cases.

Fig. 9(a) reports the running time of all methods. Obviously, a longer path or higher F^* leads to longer running time. NESTED is always the most efficient one as it uses a predetermined SPS. TREE, GRDY, and EPP all use greedy strategies, so they have similar running time performance. We pick DP-1.2/1.3 to represent the DP method because (i) they cost slightly fewer or more entanglements compared with TREE (Fig. 6 and Fig. 7); (ii) their achieved E2E fidelity is slightly lower or higher than TREE's (Fig. 8(a)). Not surprisingly, DP is inefficient as it has to try all budget splits, which can be a very large amount when the path is long. The running time



(a) Binary ($F^* = 0.99$) (b) Werner ($F^* = 0.9$)

Fig. 8: Achieved E2E fidelity of different methods in Binary and Werner systems when the device level is H.

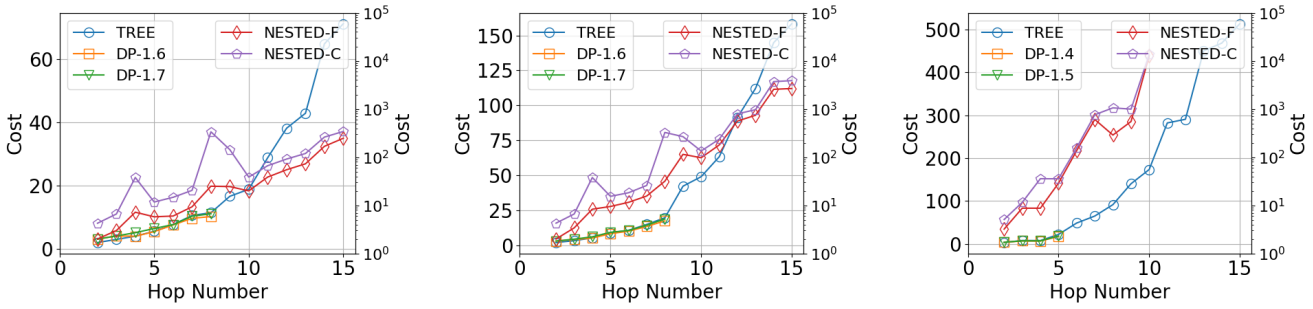


(a) Binary ($F^* = 0.99$) (b) Werner ($F^* = 0.9$)

Fig. 9: Running time of different methods in Binary and Werner systems when the device level is H.

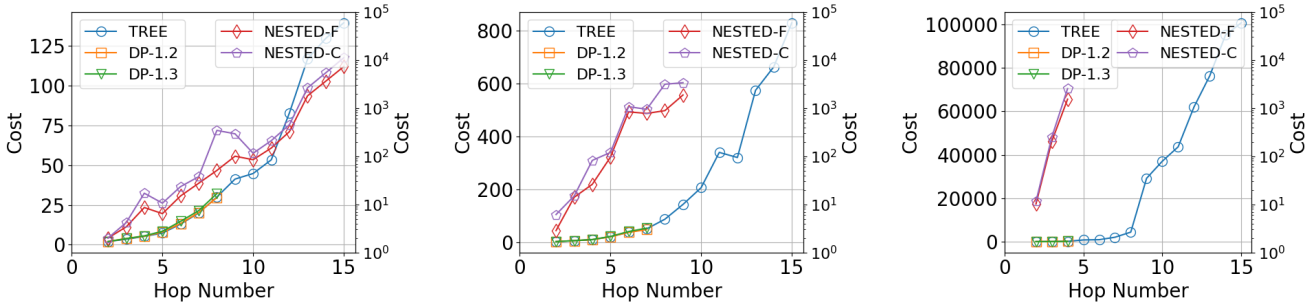
of DP increases drastically as the path length increases, so we do not test it on long paths here.

In summary, in Binary systems, NESTED performs the worst in terms of fidelity and cost. Recall that Theorem 1 shows that the OSPT in Binary systems should arrange purifi-



(a) $F^* = 0.85$ (b) $F^* = 0.9$ (c) $F^* = 0.95$

Fig. 10: Path efficiency of Group 2 methods under different F^* in Werner systems when the device level is D.



(a) device level of P (b) device level of H (c) L

Fig. 11: Path efficiency of Group 2 methods under different device reliability levels in Werner systems ($F^* = 0.9$).

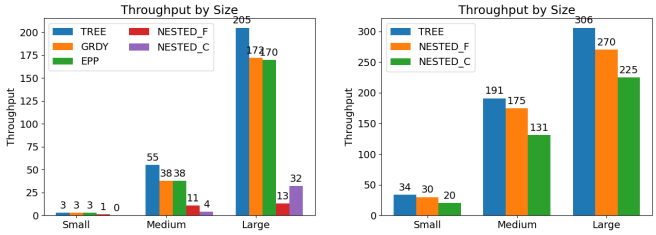
cation before swapping. But NESTED clearly disobeys such a structure. The other methods follow the PS structure, resulting in better efficiency compared to NESTED.

2) Path Efficiency in Werner System: We also compare TREE with Group 2 methods (DP and NESTED) in terms of fidelity, cost, and running time in Werner systems. GRDY and EPP are not consider here, since both do not work for Werner systems. For example, both cannot purify a simple 2-hop path with link fidelity at 0.95 to reach an E2E fidelity threshold $F^* = 0.95$, due to their purification order.

Figs. 10, Fig. 8(b) and Fig. 9(b) show the results in Werner systems with highly reliable devices. We can draw similar conclusions regarding all methods similar to that in Binary systems. DP-1.3 and DP-1.4 consume similar costs and achieve similar fidelity to TREE's, but still become extremely inefficient in terms of running time when the path is long. NESTED behaves similarly in Werner systems compared with Binary ones: (i) it typically consumes much more entanglements while does not achieve comparable E2E fidelity compared with TREE; (ii) it performs better when the path length is an exponent; (iii) it usually finishes quickly. Overall, our proposed method TREE still performs best in Werner systems with highly reliable devices.

We also consider Werner systems with devices of different reliability levels in Fig. 11 when $F^* = 0.9$. The difference is that all methods need more entanglements to achieve the same fidelity when the system is noisy.

3) Network Throughput in Binary/Werner Systems: Finally, we evaluate the network-wide throughput when the proposed TREE and baselines (if work in the corresponding scenarios)



(a) Binary, $F^* = 0.99$ (b) Werner, $F^* = 0.9$

Fig. 12: Network throughput of all methods for networks with different sizes when the device level is P/D for Binary/Werner.

are used for solving the candidate paths in a network scheduler (presented in Section IV-D) in both Binary and Werner systems. We consider three sizes of networks, defined in Table III. We do not include DP in this evaluation because it requires too long time for long paths even in a small network and if such paths are selected the scheduler will be stuck by DP. GRDY and EPP are not tested for Werner systems as they do not work there due to the reason mentioned above.

Results are reported in Fig. 12. Obviously, in a smaller network (so less demands), the throughput of all methods is smaller. More importantly, in all cases, our method TREE achieves the highest throughput among all methods. We also perform tests over different F^* and different noise levels (P, H and L), but due to space limitations, we ignore reporting the results here. Among the different noise levels, a lower noise level always leads to higher throughput.

VIII. Conclusion and Future Works

We studied the joint swapping and purification scheme to generate the required E2E fidelity for an SD pair while the consumed entanglements are minimized. We considered more general assumptions (both Binary and Werner systems with possible failures of swapping/purification operations). By leveraging tree-based modeling, we proved that the optimal SPS in Binary systems is a PS strategy where purification is before swapping. We then proposed a tree-based solution that can efficiently schedule swapping and purification for the SPS problem in both Binary and Werner systems. We also show that such solution outputs a PS strategy. We confirmed nice performances of the proposed method compared with existing solutions in both Binary and Werner systems via extensive simulations.

We leave several open problems as possible future works. (i) Theorem 1 is only true for Binary systems. It is still open whether it holds under practical assumptions in Werner systems. (ii) How to optimally solve the OSPS problem (to find the exact solution) in both systems is still open. One of the major difficulties is the complexity of the fidelity formulas (for swapping and purification). When they stack up, the final fidelity and cost have high order, which makes the problem prohibiting for current linear/quadratic solvers. One possible direction is using quantum algorithms [6], [50], [51] or other advanced optimization techniques to tackle this complex optimization problem. (iii) Extending existing results to more accurate system models, such as the stochastic behavior of operations (entanglement generation, swapping [21], and purification) and noisy memory, would also be of interest but cast new technical challenges.

References

[1] R. Van Meter, *Quantum networking*. John Wiley & Sons, 2014.

[2] A. Dahlberg, M. Skrzypczyk, T. Coopmans, L. Wubben, F. Rozpędek, *et al.*, "A link layer protocol for quantum networks," in *Proc. of the ACM special interest group on data communication*, 2019, pp. 159–173.

[3] Y. Wang, A. N. Craddock, R. Sekelsky, M. Flament, and M. Namazi, "Field-deployable quantum memory for quantum networking," *Physical Review Applied*, vol. 18, no. 4, p. 044058, 2022.

[4] X. Wei, J. Liu, L. Fan, Y. Guo, Z. Han, and Y. Wang, "Optimal entanglement distribution problem in satellite-based quantum networks," *IEEE Network*, vol. 39, no. 1, pp. 97–103, 2025.

[5] J. Liu, X. Liu, X. Wei, and Y. Wang, "Topology design with resource allocation and entanglement distribution for quantum networks," in *Proc. of IEEE SECON*, 2024.

[6] X. Wei, L. Fan, Y. Guo, Z. Han, and Y. Wang, "Entanglement from sky: Optimizing satellite-based entanglement distribution for quantum networks," *IEEE/ACM Transactions on Networking*, vol. 32, no. 6, pp. 5295–5309, 2024.

[7] J. Liu, L. Fan, Y. Guo, Z. Han, and Y. Wang, "Co-design of network topology and qubit allocation for distributed quantum computing," in *Proc. of IEEE QCNC*, 2025.

[8] X. Wei, L. Fan, Y. Guo, Z. Han, and Y. Wang, "Optimizing satellite-based entanglement distribution in quantum networks via quantum-assisted approaches," in *Proc. of IEEE QCNC*, 2024.

[9] Y. Zhang, Y. Gong, L. Fan, Y. Wang, Z. Han, and Y. Guo, "Efficient entanglement routing for satellite-aerial-terrestrial quantum networks," in *Proc. of IEEE ICCCN*, 2025.

[10] S. Pouryousef, N. K. Panigrahy, and D. Towsley, "A quantum overlay network for efficient entanglement distribution," *arXiv preprint arXiv:2212.01694*, 2022.

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

[12] 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*, vol. 70, no. 10, pp. 6748–6763, 2022.

[13] W. Dai, T. Peng, and M. Z. Win, "Optimal remote entanglement distribution," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 3, pp. 540–556, 2020.

[14] W. Dür, H.-J. Briegel, J. I. Cirac, and P. Zoller, "Quantum repeaters based on entanglement purification," *Physical Review A*, vol. 59, no. 1, p. 169, 1999.

[15] N. K. Panigrahy, T. Vasantam, D. Towsley, and L. Tassiulas, "On the capacity region of a quantum switch with entanglement purification," *arXiv preprint arXiv:2212.01463*, 2022.

[16] L. Chen and Z. Jia, "On optimum entanglement purification scheduling in quantum networks," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 7, pp. 1779–1792, 2024.

[17] Z. Jia and L. Chen, "From entanglement purification scheduling to fidelity-constrained entanglement routing," in *Proc. of IEEE ICNP*, 2024.

[18] Z. Jia and L. Chen, "From entanglement purification scheduling to fidelity-constrained multi-flow routing," *arXiv preprint arXiv:2408.08243*, 2024.

[19] J. Liu, X. Zhang, X. Wei, X. Liu, Y. Chen, H. Gao, and Y. Wang, "Joint swapping and purification with failures for entanglement distribution in quantum networks," in *Proc. of IEEE/ACM IWQoS*, 2025.

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

[21] J. Liu and Y. Wang, "Statistical modeling and latency optimization for entanglement routing in quantum networks," in *Proc. of IEEE QCE*, 2025.

[22] Y. Zeng, J. Zhang, J. Liu, Z. Liu, and Y. Yang, "Multi-entanglement routing design over quantum networks," in *Proc. of IEEE INFOCOM*, 2022, pp. 510–519.

[23] 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. on Networking*, vol. 32, no. 6, pp. 5033–5048, 2024.

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

[25] M. Victora, S. Tserkis, S. Krastanov, A. S. de la Cerda, S. Willis, and P. Narang, "Entanglement purification on quantum networks," *Phys. Rev. Res.*, vol. 5, p. 033171, Sep 2023.

[26] L. Jiang, J. M. Taylor, N. Khaneja, and M. D. Lukin, "Optimal approach to quantum communication using dynamic programming," *Proceedings of the National Academy of Sciences*, vol. 104, no. 44, pp. 17 291–17 296, 2007.

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

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

[29] S. Wengerowsky, S. K. Joshi, F. Steinlechner, J. R. Zichi, S. M. Dobrovolskiy, R. Van der Molen, J. W. Los, V. Zwiller, M. A. Versteegh, A. Mura *et al.*, "Entanglement distribution over a 96-km-long submarine optical fiber," *Proceedings of the National Academy of Sciences*, vol. 116, no. 14, pp. 6684–6688, 2019.

[30] S. Hermans, M. Pompili, H. Beukers, S. Baier, J. Borregaard, and R. Hanson, "Qubit teleportation between non-neighbouring nodes in a quantum network," *Nature*, vol. 605, no. 7911, pp. 663–668, 2022.

[31] F. Nosrati, A. Castellini, G. Compagno, and R. Lo Franco, "Robust entanglement preparation against noise by controlling spatial indistinguishability," *npj Quantum Information*, vol. 6, no. 1, p. 39, 2020.

[32] K. Heshami, D. G. England, P. C. Humphreys, P. J. Bustard, V. M. Acosta, J. Nunn, and B. J. Sussman, "Quantum memories: emerging applications and recent advances," *Journal of modern optics*, vol. 63, no. 20, pp. 2005–2028, 2016.

[33] M. K. Bhaskar, R. Riedinger, B. Machielse, D. S. Levonian, C. T. Nguyen, E. N. Knall, H. Park, D. Englund, M. Lončar, D. D. Sukachev *et al.*, "Experimental demonstration of memory-enhanced quantum communication," *Nature*, vol. 580, no. 7801, pp. 60–64, 2020.

[34] A. Kamimaki, K. Wakamatsu, K. Mikata, Y. Sekiguchi, and H. Kosaka, "Deterministic bell state measurement with a single quantum memory," *npj Quantum Information*, vol. 9, no. 1, p. 101, 2023.

[35] P. Drmota, D. Main, D. Nadlinger, B. Nichol, M. Weber, E. Ainley, A. Agrawal, R. Srinivas, G. Araneda, C. Ballance *et al.*, "Robust quantum memory in a trapped-ion quantum network node," *Physical Review Letters*, vol. 130, no. 9, p. 090803, 2023.

[36] D. Lago-Rivera, S. Grandi, J. V. Rakonjac, A. Seri, and H. de Riedmatten, "Telecom-heralded entanglement between multimode solid-state quantum memories," *Nature*, vol. 594, no. 7861, pp. 37–40, 2021.
[37] H. Bartling, M. Abobeih, B. Pingault, M. Degen, S. Loenen, C. Bradley, J. Randall, M. Markham, D. Twitchen, and T. Taminiau, "Entanglement of spin-pair qubits with intrinsic dephasing times exceeding a minute," *Physical Review X*, vol. 12, no. 1, p. 011048, 2022.
[38] Y. Ma, Y.-Z. Ma, Z.-Q. Zhou, C.-F. Li, and G.-C. Guo, "One-hour coherent optical storage in an atomic frequency comb memory," *Nature communications*, vol. 12, no. 1, p. 2381, 2021.
[39] 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*, vol. 9, no. 32, p. eadf4080, 2023.
[40] F. Ewert and P. van Loock, "3/4-efficient bell measurement with passive linear optics and unentangled ancillae," *Physical review letters*, vol. 113, no. 14, p. 140403, 2014.
[41] R. Reyes, T. Nakazato, N. Imaike, K. Matsuda, K. Tsurumoto, Y. Sekiguchi, and H. Kosaka, "Complete bell state measurement of diamond nuclear spins under a complete spatial symmetry at zero magnetic field," *Applied Physics Letters*, vol. 120, no. 19, 2022.
[42] X. Rong, J. Geng, F. Shi, Y. Liu, K. Xu, W. Ma, F. Kong, Z. Jiang, Y. Wu, and J. Du, "Experimental fault-tolerant universal quantum gates with solid-state spins under ambient conditions," *Nature communications*, vol. 6, no. 1, p. 8748, 2015.
[43] R. Van Meter, *Quantum networking*. John Wiley & Sons, 2014.
[44] 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*, vol. 77, no. 13, p. 2818, 1996.
[45] 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*, vol. 76, no. 5, p. 722, 1996.
[46] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, "Mixed-state entanglement and quantum error correction," *Physical Review A*, vol. 54, no. 5, p. 3824, 1996.
[47] A. Zang, X. Chen, A. Kolar, J. Chung, M. Suchara, et al., "Entanglement distribution in quantum repeater with purification and optimized buffer time," *arXiv preprint arXiv:2305.14573*, 2023.
[48] Traffic Engineering Applying Value at Risk, https://github.com/manyaghobadi/teavar.
[49] A. Hagberg, P. Swart, and D. S Chult, "Exploring network structure, dynamics, and function using networkx," Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 2008.
[50] T. Hogg and D. Portnov, "Quantum optimization," *Information Sciences*, vol. 128, no. 3-4, pp. 181–197, 2000.
[51] X. Wei, L. Fan, Y. Guo, Y. Gong, Z. Han, and Y. Wang, "Hybrid quantum-classical benders' decomposition for federated learning scheduling in distributed networks," *IEEE Trans. on Network Science and Engineering*, vol. 11, no. 6, pp. 6038–6051, 2024.

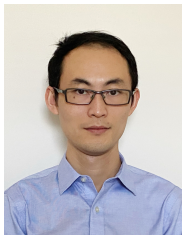
Jiyao Liu (S'24) is currently pursuing the Ph.D. degree in Computer and Information Sciences at Temple University, Philadelphia, PA, USA. His research interests span distributed quantum computing, quantum networking, and machine learning. He received the Outstanding Research Assistant Award from both College of Science and Technology and Department of Computer and Information Sciences at Temple University, respectively.

Xinwen Zhang is a Ph.D. student in the Department of Computer and Information Sciences at Temple University. Her research primarily focuses on large-scale otpimiation, particularly stochastic optimization beyond minimization and distributed algorithms for modern deep learning models. She has published multiple pioneering works in top machine learning venues, such as ICDM, ICML and NeurIPS. She is a recipient of KDD 2023 Student Travel Award, NeurIPS 2023 Scholar Award, WiML 2023 Travel Award, and ICML 2024 Travel Award.

Xinliang Wei (S'21-M'23) is an Assistant Professor in Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences. He holds a Ph.D. in Computer and Information Sciences from Temple University, Philadelphia, USA in 2023. He received his M.S. and B.E. degrees both in Software Engineering from SUN Yat-sen University, Guangzhou, China in 2016 and 2014, respectively. His research interests include edge computing, federated learning, reinforcement learning, and Internet of Things. He is a recipient of *Outstanding Research Assistant Award* from College of Science and Technology (2022) and *Scott Hibbs Future of Computing Award* from Department of Computer & Information Sciences (2023) at Temple University.

Xuanzhang Liu received the master's degree in computer science from University of Delaware in 2020. He is currently pursuing his PhD degree at the Department of Computer and Information Science, Temple University. His research interests include edge computing, blockchain, and federated learning.

Yuzhou Chen is an Assistant Professor in Department of Statistics at University of California, Riverside. He is also an Adjunct Assistant Professor in Department of Computer and Information Sciences at Temple University and a Visiting Research Collaborator in Department of Electrical and Computer Engineering at Princeton University. He received his Ph.D. in Statistics from Southern Methodist University. His research interests are machine learning, deep learning, graph mining, knowledge reasoning, knowledge graph, topological data analysis, reliability theory, nonparametric statistics, and their applications.



Hongchang Gao received the B.S. degree from the Ocean University of China, Qingdao, China, in 2011, the M.S. degree from Beihang University, Beijing, China, in 2014, and the Ph.D. degree from the University of Pittsburgh, Pittsburgh, PA, USA, in 2020. He is currently an Assistant Professor in the Department of Computer and Information Sciences at Temple University. His research interests include machine learning, optimization, and data mining.

Yu Wang (S'02-M'04-SM'10-F'18) is a Professor and Chair of the Department of Computer and Information Sciences at Temple University. He holds a Ph.D. from Illinois Institute of Technology, an MEng and a BEng from Tsinghua University, all in Computer Science. His research interest includes wireless networks, smart sensing, and distributed computing. He has published over 300 papers in peer-reviewed journals and conferences. He is a recipient of *Ralph E. Powe Junior Faculty Enhancement Awards* from Oak Ridge Associated Universities (2006), *Outstanding Faculty Research Award* from College of Computing and Informatics at the University of North Carolina at Charlotte (2008), *Fellow of IEEE* (2018), *ACM Distinguished Member* (2020), and *IEEE Benjamin Franklin Key Award* (2024). He has served as Associate Editor for *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Cloud Computing*, among others.