https://doi.org/10.1038/s42005-024-01796-2

Purification scheduling control for throughput maximization in quantum networks

Check for updates

Zirui Xiao[1], Jian Li [1], Kaiping Xue [1] ✉, Nenghai Yu[1], Ruidong Li[2], Qibin Sun[1] & Jun Lu[1]

Quantum networks can establish End-to-End (E2E) entanglement connections between two arbitrary nodes with desired entanglement fidelity by performing entanglement purification to support quantum applications reliably. The existing works mainly focus on link-level purification scheduling and lack consideration of purifications at network-level, which fails to offer an effective solution for concurrent requests, resulting in low throughput. However, efficiently allocating scarce resources to purify entanglement for concurrent requests remains a critical but unsolved problem. To address this problem, we explore the purification resource scheduling problem from a network-level perspective. We analyze the cost of purification, design the E2E fidelity calculation method in detail, and propose an approach called Purification Scheduling Control (PSC). The basic idea of PSC is to determine the appropriate purification through jointly optimizing purification and resource allocation processes based on conflict avoidance. We conduct extensive experiments that show that PSC can maximize throughput under the fidelity requirement.

In recent decades, along with the proof-of-concept validation of quantum repeaters and long-distance quantum communications, quantum networks have been gradually developed from theory to practice[1–4]. As the cornerstone of quantum networks, entanglement distribution can be achieved over quantum channels (e.g., optical fiber or a free-space link[4,5]) through single/multiple quantum repeaters[6]. Once the end-to-end (E2E) entanglement connections are established between two arbitrary quantum nodes, quantum networks can support various quantum applications via teleportation, such as quantum key distribution[7], quantum clock synchronization[8], blind quantum computation[9], and distributed computation[10].

However, considering scarce entanglement resources and probabilistic quantum operations on quantum repeaters, how to efficiently establish long-distance entanglement connections in a quantum network remains an open challenge, which is the so-called “remote entanglement distribution problem”[11]. To address this problem, some pioneering studies are proposed from the perspective of routing design, such as studying efficient path-finding and resource allocation algorithm[12–15] in the entanglement distribution process. However, most of these studies consider the inherent property of the probabilistic failure of quantum operations and propose the entanglement recovery strategy[16] to improve the performance in terms of robustness and throughput ref.[17] and ref. 18 focus on the design of entanglement resource allocation scheme and quantum network hardware allocation scheme (i.e., quantum channels and quantum storage) among multiple requests, respectively, so that each request can be allocated resources more efficiently and fairly. However, one critical metric, i.e., E2E entanglement fidelity, that evaluates the quality of entanglement connections is rarely considered in the existing designs. In practice, entanglement fidelity describes how well the current system maintains entanglement with an ideal system. The higher the fidelity, the higher the probability of successful quantum operations. Besides, the effects of environment and quantum gate noise on the fidelity of entangled particles degrade the performance of entanglement routing algorithms that do not take fidelity metrics into account during real-world applications. Therefore, to support upper-layer quantum applications reliably, it is necessary to give new consideration to fidelity in traditional quantum entanglement routing problems and ensure the E2E fidelity of entanglement connections satisfies certain fidelity constraints[19]. For example, quantum cryptography protocols (e.g., E91) require the fidelity of entanglement to exceed the quantum bit error rate to ensure the security of key distribution[17].

To improve the fidelity of entanglement connections, a physically effective method is introducing entanglement purification, which has been widely used to prevent fidelity degradation. With entanglement purification, shared lower-fidelity entangled pairs between adjacent quantum repeaters are consumed to obtain one higher-fidelity entangled pair[19]. Theoretically, an arbitrary high-fidelity entanglement connection can be established if there are sufficient entanglement resources. However, although introducing

[1]School of Cyber Science and Technology, University of Science and Technology of China, Hefei, Anhui, 230027, China. [2]School of Science and Engineering, Kanazawa University, Kakuma, Kanazawa, 920-1192, Japan. ✉e-mail: kpxue@ustc.edu.cn

purification can improve the fidelity of entanglement connection, it also brings out additional entanglement resource consumption. Considering the scarcity of available entanglement resources, the purification resource scheduling problem should be further addressed to create as many entanglement connections with a desired E2E fidelity as possible in a quantum network. Namely, an efficient purification scheme needs to be designed in an environment where entanglement resources are limited so that additional resources can be spent as little as possible to obtain an entanglement connection under the fidelity requirement.

Although a few existing studies are concentrating on the purification resource scheduling problem, all of them only consider link-level purification scheduling[17,20] rather than network-level purification scheduling, which can't provide an effective solution for concurrent requests along with various routes and further leads to low throughput. Thus, it is a challenge to design an efficient scheduling scheme from the network-level perspective, since it should be a tradeoff between the resources consumed by purification against those used for entanglement swapping and the maximization of network throughput under the premise of fidelity requirement.

To meet this challenge, we focus on the purification resource scheduling problem at the network-level. To the best of our knowledge, this work is the first complete exploration of the purification resource scheduling problem from a network-level perspective, and we design a network-level purification scheduling algorithm (i.e., the PSC algorithm). First, to provide a purification model for E2E entanglement connection, we formulate a simple but effective model[21,22] to quantify the effect of link-level purification on the fidelity of E2E entanglement connections. Second, we analyze the impact of entanglement purification on entanglement resource allocation for different links. Third, based on the impact of the intensity of the conflict link on network performance, we propose an effective Purification Scheduling Control (PSC) algorithm to schedule resources and further achieve efficient purifications at the network-level. PSC determines the critical links for purification and jointly optimizes the purification and resource allocation processes based on conflict avoidance. It then maximizes throughput under the constraint of fidelity requirement. Besides, PSC can satisfy the various fidelity requirements of different quantum applications to distribute long-distance entangled pairs by adjusting input parameters and can improve the performance of quantum applications. For example, PSC can improve the success probability of key distribution in QKD scenarios and help teleport qubits fast in distributed quantum computing scenarios. Finally, we conduct extensive simulations on an open-source quantum network simulation platform, i.e., SimQN[23], for large-scale system-level simulation verification. Results show that our proposed scheduling scheme can adapt to various network scenarios and outperform the existing schemes in terms of throughput and resource consumption.

Results and discussion

We use a discrete-event-based quantum network simulation platform SimQN[23], which can support system-level simulations of large-scale quantum networks. The LP solver used in our simulator is the GEKKO Optimization suite package. Regarding network topology construction in the experiments, we use a random topology generation method based on a minimum spanning tree to generate an arbitrary quantum network topology with N quantum nodes, which are connected randomly with $1.5N$ quantum links. By default, the network has 100 quantum nodes and 20 quantum requests. The initial fidelity of the entangled pairs generated using quantum links, F_0, is between [0.90,0.95], and the capacity of the quantum links is 100 (we can generate up to 100 entanglement resources between adjacent nodes). We also set the threshold F^* of the required fidelity of each request to 0.8. In addition, we use two schemes as the comparison scheme. The first is the Propagatory Update (PU) algorithm which has performed better for resource allocation, and it uses the threshold-based (F_{th}) purification strategy in[17]. The second is a greedy strategy, where the fidelity of the entanglement links is improved by purifying each quantum link to obtain more entanglement connections satisfying the requirements. We use the PU resource allocation algorithm in the greedy strategy to prevent the effect of different resource allocation methods. We run each set of parameters 200 times to obtain statistical expectations for each performance metric and reduce randomness.

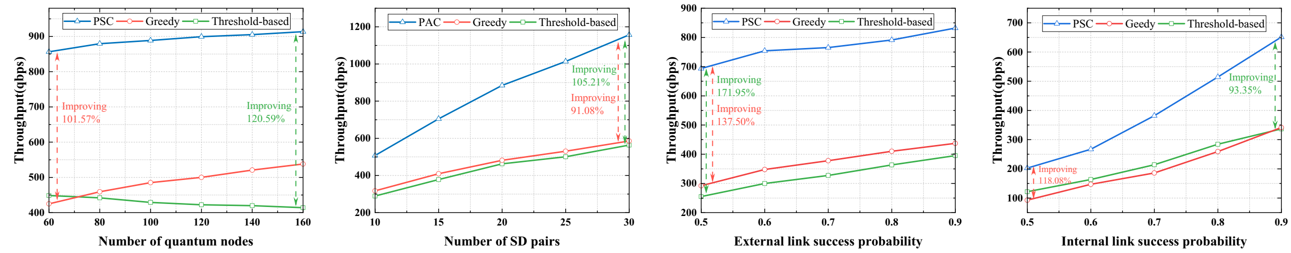
In terms of the performance metrics, differing from the traditional definition of throughput, we define throughput as the number of entanglement connections that satisfy the fidelity requirement (qubit per slot, qbps). In addition, we also define the purification resource consumption ratio for each entanglement connection that satisfies the fidelity requirement, as shown in Eq. (1), which is the ratio of additional resources consumed by purification to the entanglement connections established in the whole network. It represents the additional resources consumed by purification needed to establish an E2E entanglement connection that satisfies the fidelity requirement. The smaller the value of this metric is, the better it is.

$$ratio = \sum_{i=1}^{|\mathbf{R}|} \frac{puri\_cost_i}{x_i}. \tag{1}$$

At first, we vary the number of quantum nodes in the network to evaluate the performance of each algorithm at different network scales. As shown in Fig. 1a, as the network size increases, the throughput of PSC and Greedy algorithms initially increases to some extent, while the throughput of the Threshold-based algorithm continuously decreases. This phenomenon can be explained as follows. With the increasing network size, the probability of choosing overlapping entanglement paths between different S-D pairs decreases, thereby enhancing resource utilization. At the same time, the increase in network size imposes higher entanglement link fidelity requirements to meet the end-to-end fidelity requirement. Since both the PSC and Greedy algorithms ensure that the established entanglement connections meet the fidelity requirements, leading to a significant improvement in terms of throughput. However, the Threshold-based algorithm does not quantitatively consider the fidelity of entanglement connections, and thus, established connections do not meet the required fidelity, resulting in a reduction in throughput. We can see that regardless of whether the network size is large or small, the PSC algorithm works better than the Greedy algorithm and the Threshold-based algorithm. PSC obtains the highest throughput across the network, outperforming the Threshold-based and Greedy algorithms by up to 120.59 and 101.57%, respectively. The purification resource consumption ratio of the three algorithms increases as the network size increases, as shown in Fig. 2a. The Threshold-based algorithm has a lower purification resource consumption ratio than the Greedy algorithm because it is selectively purified. But its ratio is still much higher than the PSC purification resource consumption ratio. So PSC consumes the fewest resources to establish an entanglement connection that meets the E2E fidelity requirement.

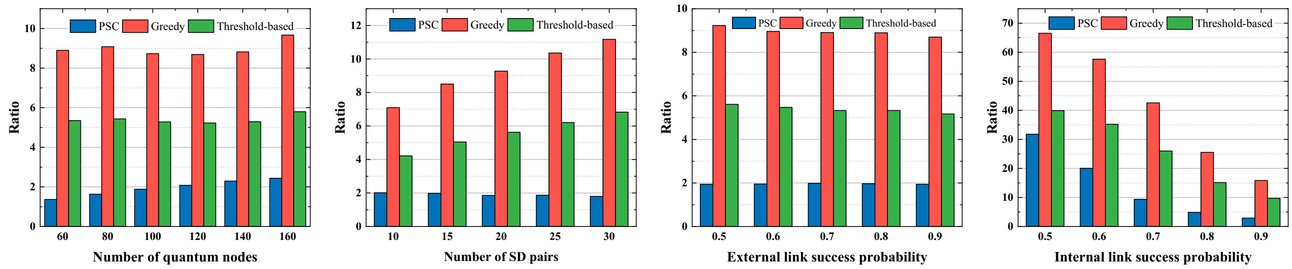
Second, we vary the number of requests in the network to explore the algorithm's performance in a real-world scenario with high concurrency. As shown in Fig. 1b, we can see that the throughput increases with the number of requests increasing, and the PSC shows advantages in both low-load and high-load scenarios. The purification resource consumption ratio fluctuates slightly, because there is no exact positive or negative relationship between the purification resource consumption ratio and the number of requests. However, as shown in Fig. 2b, we can still observe that the Greedy algorithm has the highest purification resource consumption ratio, and PSC has the lowest purification resource consumption ratio.

Third, we investigate the effect of entanglement link success probability on the algorithms. As shown in Fig. 1c, we can see that the PSC consumes the least amount of resources. And the network-wide throughput obtained by all algorithms decreases to some extent as the probability of successfully establishing an entanglement link decreases because fewer entanglement resources are available. However, PSC still has a significant advantage over the Greedy algorithm and Threshold-based algorithm in terms of throughput. As shown in Fig. 2c, the purification resource consumption ratio of the three algorithms do not change significantly as the entanglement



(a) (b) (c) (d)

Fig. 1 | Throughput variations of different algorithms in different scenarios. **a** Throughput vs. network scale. Purification scheduling control (PSC) obtains the highest throughput for the whole network. When the network size is small, the throughput obtained by the threshold-based algorithm is higher than that of the Greedy algorithm because the number of hops between the source and destination nodes is relatively small. The fidelity requirement of the request may be achieved without purification, and the resources will be wasted if each link is purified. However, as the network size increases, the Threshold-based algorithm can obtain a lower throughput than the Greedy algorithm. **b** Throughput vs. S-D pairs. As the number of requests increases, more resources in the network can be used, resulting in an increase in the throughput obtained by all three algorithms. **c** Throughput vs. external link success probability. Regardless of whether the entanglement establishment probability is low or high, the PSC obtains the highest throughput, Greedy the second highest, and threshold-based obtains the lowest throughput. **d** Throughput vs. internal link success probability. As the entanglement swapping probability decreases, the throughput obtained by the PSC algorithm decreases faster than the other two algorithms. Because PSC can obtain more throughput than the other two algorithms when the entanglement swapping probability is not considered. However, when the entanglement swapping probability is considered, the PSC algorithm fails to establish a large number of entangled connections due to the failure of "internal connection" establishment. Nevertheless, PSC still achieves higher throughput than the other two algorithms. When the entanglement swapping probability is low, the threshold-based algorithm can achieve higher throughput than Greedy. Because the threshold-based algorithm spends fewer entanglement resources on entanglement purification than Greedy, it can establish more entanglement connections (most of which do not meet the fidelity requirement, and a few do). Still, the Greedy algorithm establishes entanglement connections that satisfy the fidelity requirement, so when there is an effect of entanglement swapping probability, Greedy is more likely to fail to establish entanglement connections that satisfy the fidelity requirement than threshold-based. When the entanglement swapping probability is high enough, the Greedy achieves higher throughput than threshold-based.



(a) (b) (c) (d)

Fig. 2 | Purification resource consumption ratio variations of different algorithms in different scenarios. **a** Ratio vs. network scale. The purification scheduling control (PSC) algorithm has the lowest purification resource consumption ratio because it can be considered a better purification strategy in terms of E2E entanglement fidelity. The Greedy algorithm has the highest purification resource consumption ratio because it chooses to perform entanglement purification on every link. The Threshold-based algorithm has a lower purification resource consumption ratio than Greedy because it is a threshold-based purification strategy that will decide whether to purify or not based on the link fidelity. **b** Ratio vs. S-D pairs. As the number of requests increases, the purification resource consumption ratio of both Threshold-based and Greedy algorithms increases to a certain extent. Because the increase in the number of requests causes more links to be purified, which leads to more resources consumed for purification. However, the purification resource consumption ratio of the PSC algorithm only fluctuates slightly because the PSC algorithm makes purification and resource allocation decisions based on the entanglement path, the link fidelity, and the number of entanglement resources on the link to make purification and resource allocation decisions. It can better weigh the relationship between the resources consumed by purification and the resources allocated to the request. **c** Ratio vs. external link success probability. The purification resource consumption ratio does not change significantly for the three algorithms because the increase in the entanglement establishment probability only changes the entanglement resources that can be utilized in the network. The number of available resources in the network affects both the number of resources consumed by purification and the number of resources allocated to the request to establish an entanglement connection. **d** Ratio vs. internal link success probability. The Greedy algorithm has the highest purification resource consumption ratio, followed by the Threshold-based, and the PSC algorithm has the lowest purification resource consumption ratio.

establishment probability changes. The PSC still has the lowest purification resource consumption ratio, so it has the advantage of establishing entanglement connections with fewer resources to meet the fidelity requirements.

Finally, we investigate the effect of the entanglement swapping success probability on different experimental schemes. Figure 1d indicates that the higher the probability of successful entanglement swapping, the higher the throughput. When the probability of successful entanglement swapping is small, the number of entanglement connections we can establish will not be particularly high. Because even if the optimal purification and resource allocation strategy are used, it may still fail to improve the throughput due to the inability to establish the "internal link"[13]. It results in a slightly better performance of Threshold-based than the Greedy algorithm, and the gap decreases as the probability of successful entanglement swapping increases. As shown in Fig. 2d, when the success probability of entanglement swapping decreases, the resource utilization of each scheme increases due to the difficulty of establishing an entanglement connection, which requires more resources to establish an entanglement connection. However, the PSC still has an advantage over the other two schemes.

Based on the results of large-scale system-level simulation experiments, the superiority of PSC is verified. In addition to this, we also qualitatively consider link failures and node failures to verify the robustness of the PSC algorithm against random network failures (see Supplementary Note 3 for the figure and specific numerical simulation results are given in Supplementary Data 1). The results demonstrate that PSC is robust and can be adapted to different application scenarios. Regarding performance metrics, PSC achieves higher network throughput and more efficient resource utilization than the existing Greedy algorithm and Threshold-based algorithm.

Conclusions

We examined the different solutions available today for the purification resource scheduling problem under different scenarios of different network sizes and the number of concurrent requests. Performance results show that PSC has significant advantages. In terms of network throughput and purification resource consumption ratio, the PSC algorithm stands out by a large margin, and the superiority is largely maintained under all network conditions, even in the face of network failures. It suggests that PSC utilizes network resources in the most economical way and is more robust against network failures.

In conclusion, we proposed an efficient PSC algorithm that tackles the purification resource scheduling problem on quantum networks. The PSC algorithm jointly optimizes the purification and resource allocation process to avoid the impact of bottleneck links on network performance caused by entanglement purification and to ensure that each established entanglement connection meets the E2E fidelity requirements.

We now present the scope of use, limitations, and potential future extensions of the routing scheme. In this paper, we study the remote entanglement distribution problem by introducing entanglement purification and consider the problem from the perspective of purification resource scheduling. Many papers have studied the path selection problem, but purification resource scheduling is a new problem, and we have designed the PSC algorithm for it. The PSC algorithm is highly flexible and extensible, and we can directly couple with the existing (or future) path selection algorithms with the PSC algorithm.

In the future, we will study how to use the limited local information to perform remote entanglement distribution for requests in the network and determine an efficient purification timing when we consider performing purification operations between multi-hop entanglement. We will also explore more accurate end-to-end fidelity quantification formulas and corresponding purification resource scheduling problems in scenarios where the fidelity of each entanglement link is different.

Methods

Network Model: An arbitrary quantum network is denoted as the graph $G = (V, E, C)$, where V is the set of $|V|$ quantum nodes, E is the set of $|E|$ edges, and C is the set of the capacity of each edge. Two adjacent nodes can share more than one quantum link. The Source-Destination (S-D) pair of a routing request is represented as $< s_i, d_i >$. If two quantum nodes have an edge (u, v), then there are one or more quantum channels between these two nodes, and let W denote the number of channels on the edge. To create the desired Bell pair (e.g., $|\beta_{00}\rangle$) between a pair of neighboring nodes, two nodes (u and v) simultaneously attempt to create Bell pairs on the quantum channel that connects them. The capacity $c(u, v)$ determines the maximum number of Bell pairs that u and v can create in a time slot.

Fidelity Model: Although the Bell state may not be in the maximally entangled state (i.e., the ideal state) for many reasons, we use the fidelity to evaluate the quality of the Bell state. In this paper, we only consider bit-flip error, which may be experienced when each of two qubits is measured. Because the bit-flip error model is the most fundamental and common physical model for studying fidelity[24], our study based on the bit-flip error model helps us to establish a clear understanding of the nature of the research problem. For example, the existing work[19] explores the entanglement routing problem in bit-flip error scenarios and can select paths for requests that satisfy the fidelity requirements. The initial fidelity of Bell pairs between adjacent nodes is denoted by F_0. Here, each request $<s_i, d_i, F_i^*>$ has F_i^* as the E2E fidelity requirement. To ensure that the E2E fidelity is above a certain threshold F_i^*, we introduce a purification operation to improve the E2E fidelity. Accordingly, when we perform a purification operation, the probability of true positive is F_0^2, and the probability of getting a positive measurement result is $F_0^2 + (1 - F_0)^2$. Thus, the fidelity of the Bell pair after 1-round purification will be $\frac{F_0^2}{F_0^2+(1-F_0)^2}$[19]. It is worth noting that an additional Bell pair is consumed during each round of purification. We call this Bell pair the sacrificial pair.

Operation Model: All quantum nodes are connected through a classical network and controlled by a centralized controller through this classical network. Besides, the quantum network is synchronized to a clock, where the time slot is a constant related to the decoherence time of entangled pairs. Since the lifetime of the quantum memory can be maintained for several seconds or even hours[25–27] and random access quantum memory emerged[28], we can store entangled particles generated over a period in the quantum memory and use them according to the decisions of a central controller.

In the scenario illustrated above, we define the purification resource scheduling problem as the following: Given a quantum network with arbitrary topology, imperfect quantum operations, limited quantum memory resources, and the same fidelity of entanglement links, how to provide a purification and resource allocation solution for each request to maximize the throughput (i.e., the number of entanglement connections that satisfy the fidelity requirement) under the premise of fidelity constraint (i.e., the fidelity of the established entanglement connections exceeds a threshold).

To solve the purification resource scheduling problem, first, we analyze the E2E fidelity calculation method for a single request when considering link-level purification. Then, we further consider multiple requests based on the conclusion of a single request to jointly optimize the purification resource scheduling and resource allocation for the whole network by modeling.

To measure the fidelity enhancement due to purification, we expand the formula for calculating the fidelity after multiple rounds of purification and give the formula for calculating the E2E fidelity for a single request. Under the premise that the two qubits of a Bell pair will flip with the same probability, the fidelity $F^{(T)}$ after T rounds of purification can be calculated iteratively by[19]:

$$F^{(T)} = \frac{FF^{(T-1)}}{FF^{(T-1)} + (1 - F)(1 - F^{(T-1)})}, \tag{2}$$

where $F^{(0)} = F_0$ represents the initial fidelity. Eq. (2) is difficult to derive a close form of the E2E fidelity equation for all purification methods because it considers the possibility of using entangled pairs with different fidelity in each round of purification. Similar to the analysis of final entanglement fidelity proposed in the existing study[20], all sacrificial pairs are generated and consumed simultaneously without considering the generation of new sacrificial pairs in the middle, and the fidelity of all sacrificial pairs is identical. Based on the above considerations, Eq. (2) can be extended in a particular case so that we can obtain the fidelity of the entangled pairs after T rounds of purification as Eq. ((3)), and it can be proved using mathematical induction (see Supplementary Note 1).

$$F^{(T)} = \frac{F^{T+1}}{F^{T+1} + (1 - F)^{T+1}}. \tag{3}$$

Multiple rounds of purifications result in more fidelity improvement but consume more resources. We use the fidelity enhancement efficiency, i.e., the ratio of fidelity improvement (i.e., the difference between the entanglement fidelity before purifications and the entanglement fidelity after purifications) to resource consumption (i.e., the number of Bell pairs as sacrificial pairs), to measure the cost-effectiveness of multiple rounds of purification, as shown in Fig. 3.

Considering the fidelity degradation during the entanglement swapping[20,29], the cumulative multiplication result of E2E entanglement connections obtained after single/multiple entanglement swapping operations can be calculated by Eq. (4), where $r_i(u, v)$ is a binary variable representing whether purification operation is performed on edge (u, v) for the ith request, and $\mathbf{P}_i = \{(v_1, v_2), (v_2, v_3), \ldots, (v_{n-1}, v_n)\}$ is the set of a series of edges representing the path chosen by the ith request:

$$F_{r_i} = \prod \frac{F^{r_i(u,v)+1}}{F^{r_i(u,v)+1} + (1-F)^{r_i(u,v)+1}}, \quad \forall (u, v) \in \mathbf{P}_i. \tag{4}$$

In a quantum network with multiple concurrent requests, a link may be used by more than one request. The more entanglement resources on the link are consumed by purification, the more likely it is to become a bottleneck link for these requests, thus affecting the throughput in the network. According to Fig. 3 and the corresponding legend illustration, it can be proved that one-round purification consumes fewer resources and brings more remarkable fidelity improvement than multiple-round purification (see Supplementary Note 2). Thus, we consider using 1-round purification on the link to minimize the impact of purification resource consumption in the network. Consequently, we can obtain Eq. (5), the particular expression form of Eq. (4) with one-round purification, where *len* is the length of the entanglement path obtained by the routing algorithm for request *i*:

$$F_{r_i} = \frac{F^{\sum r_i(u,v)+len}}{\left[F^2 + (1-F)^2\right]^{\sum_{i=1}^{len} r_i(u,v)}}, \quad \forall (u, v) \in \mathbf{P}_i. \tag{5}$$

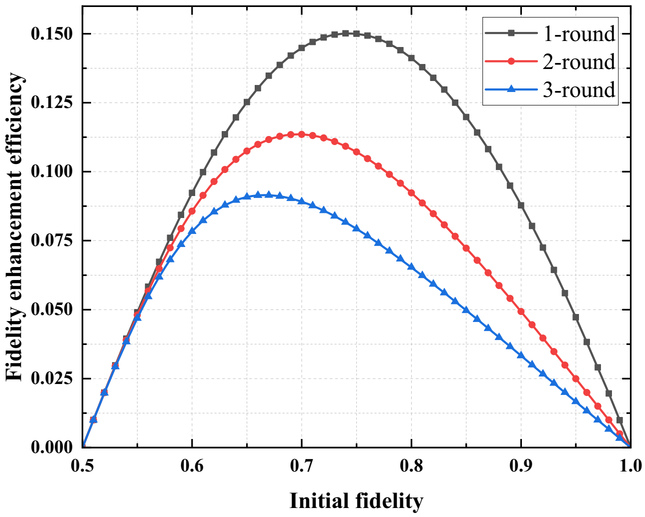


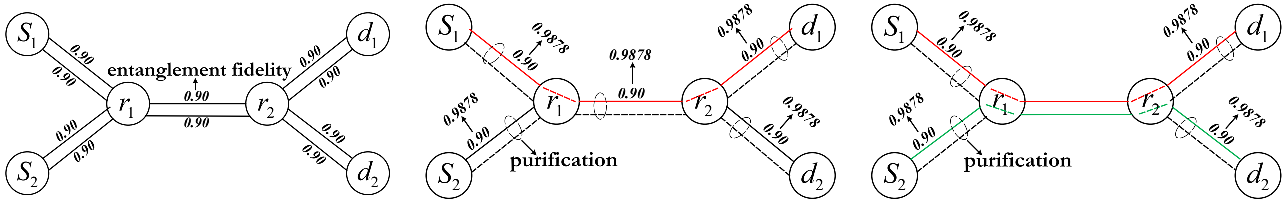
Fig. 3 | Illustration of performance for multi-round entanglement purification. It can be seen that regardless of the initial fidelity, the fidelity enhancement efficiency, i.e., the ratio of fidelity improvement to resource consumption, is the highest one for the one-round purification scheme, i.e., the fidelity improvement from the one-round purification scheme is the largest for the same resource consumption.

So far, we have Eq. (5) for quantifying the fidelity of E2E entanglement connections in combination with the entanglement purification. However, in a quantum network with multiple concurrent requests, the purification strategy will be determined by various factors. We need to consider the relationship between the resources consumed by purification, the bottleneck links, and the purification strategy's impact on the resource allocation process. Then, we need to model the purification resource scheduling problem for a multi-request scenario and trade these factors to design a more effective purification strategy, as shown in Fig. 4.

Combined with the derived Eq. (5) for E2E fidelity, we model the purification resource scheduling problem for a multi-request scenario, aiming to establish as many entanglement connections beyond fidelity threshold as possible for each request. All the symbols used in this section and their descriptions are provided in Table 1.

Table 1 | Symbol description

Parameters	Description
c_{uv}	Maximum number of entanglement links that can be created on (u, v)
e_{uv}	Number of successfully created entanglement links on (u, v)
p_{in}	The probability of success of the entanglement swapping
p_{out}	The probability of success in creating an entanglement link
s_i	The source node of the i^{th} request
d_i	The destination node of the i^{th} request
F_i^*	The fidelity requirement of the i^{th} requested entanglement connection
F_0	The initial fidelity of the created entanglement link
$\mathbf{P}_i(s_i, d_i)$	The set of a series of edges representing the path chosen by the i^{th} request
R	The set of quantum requests that need to establish long-distance entanglement
Variables	**Description**
x_i	Number of entanglement connections that should be established for the i^{th} request
$r_i(u, v)$	Number of rounds purified on the (u, v) edge for the i^{th} request
$f_i(u, v)$	Number of entanglement links on edge (u, v) for use by the i^{th} request



(a) **Network topology and network resources.** (b) **Li et at. [17] algorithm, throughput = 1.** (c) **PSC algorithm, throughput = 2.**

Fig. 4 | A comparison example between the threshold-based algorithm and the proposed purification scheduling control (PSC) algorithm. a The fidelity of the initial entanglement links established between each pair of neighboring nodes in the figure is 0.9, where each quantum link has a capacity of 2. The fidelity threshold $F_1^* = F_2^* = 0.80$. The solid lines represent the entanglement links, and the dashed lines represent the entangled pairs used as sacrificial pairs in the purification process. For the request (s_1, d_1), there is a path $s_1 \rightarrow r_1 \rightarrow r_2 \rightarrow d_1$, and for the request (s_2, d_2), there is a path $s_2 \rightarrow r_1 \rightarrow r_2 \rightarrow d_2$. We must select (s_1, d_1) and (s_2, d_2) to perform a round of purification on at least one quantum link to obtain an entanglement connection that satisfies the fidelity threshold. **b** Li et al.[17] did not quantify the fidelity of the E2E entanglement connection. They use a threshold-based purification strategy (links are purified if their fidelity is below a certain threshold). So, Li's algorithm either purifies all of them and eventually reduces the throughput or does not. **c** PSC algorithm chooses to perform entanglement purification between r_2 and d_2, and between r_2 and d_1, at which point there are still two entanglement links between r_1 and r_2, which can serve both two requests.

Based on the previously illustrated network model, fidelity model, and operation model, the purification resource scheduling problem for a multi-request scenario can be expressed in the following:

$$\text{Maximize} \quad \sum_{i=1}^{|\mathbf{R}|} x_i, \tag{6}$$

subject to:

$$\sum_{v} f_i(s_i, v) - \sum_{v} f_i(v, s_i) = x_i, \forall s_i, \tag{6a}$$

$$\sum_{v} f_i(d_i, v) - \sum_{v} f_i(v, d_i) = -x_i, \forall d_i, \tag{6b}$$

$$\sum_{v} f_i(u, v) - \sum_{v} f_i(v, u) = 0, \forall u \neq s_i, d_i, \tag{6c}$$

$$\sum_{i} \left[r_i(u, v) + 1 \right] x_i \leq e_{uv}, \forall (u, v) \in \mathbf{P}_i, \tag{6d}$$

$$\sum_{len} r_i(u, v) \cdot F' \leq \log_F F_i^* - len, \forall (u, v) \in \mathbf{P}_i, \tag{6e}$$

$$x_i \in \mathbb{Z}^+, \forall i \leq |\mathbf{R}|, \tag{6f}$$

$$r_i(u, v) \in \{0, 1\}, \forall i \leq |\mathbf{R}|. \tag{6g}$$

The objective of the problem in Eq. (6) is to maximize the number of entanglement connections that satisfy the fidelity requirement. The first three constraints, i.e., (6a–c), are the flow conservation constraints that should be held in all routing-related problems. Constraint (6d) indicates that the number of entanglement links used by all requests on edge (u, v) cannot exceed the number of successfully created entanglement links on this edge. Constraint (6e) indicates that the fidelity of the entanglement links created after purification and entanglement swapping should be no less than the threshold of the fidelity required by request, where $F' = \left[1 - \log_F(F^2 + (1 - F)^2)\right]$. We can obtain it by a mathematical variation of Eq. (5). When the fidelity of each entanglement link in the requested entanglement path is not the same, we can use the lowest fidelity among the entanglement links to quantify the lower bound of the end-to-end entanglement connection fidelity. Constraint (6f) restricts the number of entanglement connections for each request to an integer. Constraint (6g) limits the number of purification rounds for each link.

The problem modeled by network flow theory is an Integer non-linear programming (INLP) one that doesn't fit the particular convex quadratic programming model without considering the integer constraints. The watershed between the practical solvable and intractable optimization problem is convexity[30]. Therefore, it is tough to use the branch-and-bound method to derive the optimal solution directly to this INLP problem in a large-scale network system. We propose the heuristic PSC algorithm to solve the problem by considering the relationship between multiple concurrent requests competition and purification resource consumption. It can successfully solve the problem in polynomial time and establish as many entanglement connections in the network as possible that satisfy the fidelity requirements. The proposed PSC solves this problem in two steps iteratively and returns the best result among all solutions.

Algorithm 1. **Purification Scheduling Control**

```
Input: $G = (V, E, C)$, $\mathbf{R} = \{r_i\} = \{< s_i, d_i, F_i^* >\}$,
  $\mathbf{P}_i$, $F^*$, $F$;
Output: $x_i$, $r_i(u, v)$;
Initialization:
Find an entangled path $\xi = \{\{\mathbf{P}_i\}\}$ for each request
  using Dijkstra's algorithm;
Relax the integer variable in problem 6 into a
  continuous variable;
Solve the relaxed problem 6 and derive the purification
  decision on each link $\hat{r}_i(u, v) \in [0, 1]$;
Take $\hat{r}_i(u, v)$ as the purification probability for each
  link, i.e., the value of $w_i(u, v)$;
Compute the link conflict metric $C^{flic}_{(u,v)}$ for each
  entangled link on the selected entanglement path $\xi$
  based on $w_i(u, v)$;
repeat
    Using $C^{flic}_{(u,v)}$ as the probability and thus
      probabilistically setting the purification strategy
      $\zeta = \{\{r_i(u, v)\}\}$;
    Relax the integer variable $x_i$ into a continuous
      variable $x_i \geq 0$;
    Solve the corresponding LP problem in Eq. (8)
      and obtain the corresponding result $\hat{f}_i$;
    repeat
        $\hat{f}_i \leftarrow \lfloor \hat{f}_i \rfloor$;
        for $\mathbf{P}_i$ in $\xi$ do
            $f_i \leftarrow f_i + \hat{f}_i$;
            Update the remaining resources in the
              network;
        end
    until No entanglement resources;
    if $\sum f_i$ better than $\sum x_i$ then
        $x_i \leftarrow f_i$;
        Update $r_i(u, v)$ based on $\zeta$;
    end
until num < iteration;
return $x_i$, $r_i(u, v)$;
```

We find that the choice of the purification strategy affects the subsequent resource allocation. If a link is used simultaneously in multiple request entanglement paths, when we choose to purify that link, the entanglement resources consumed by the purification on that link will result in fewer entanglement resources allocated to each request for entanglement swapping. Then, it will likely become a bottleneck link and affect the whole network's subsequent resource allocation process and throughput. Based on this observation, we define the link conflict metric, $C^{flic}_{(u,v)}$, for each link, as shown in Eq. (7), where $w_i(u, v)$ is obtained by solving the relaxed problem (6) denoting the probability that request i selects the current link (u, v) for purification and Q_i is a binary variable representing whether the entanglement path of the i^{th} request includes the current link. We use $C^{flic}_{(u,v)}$ as the probability of choosing to perform purification on link (u, v). When the value of $C^{flic}_{(u,v)}$ is small, the number of requests passing through the link (u, v) is high, which means that the entanglement resources on this link are allocated for multiple requests. If we choose to purify this link, it may impact the resource allocation of multiple requests, thus affecting the throughput of the whole network. Therefore, it is necessary to avoid purification on this link as much as possible. The specific details of the PSC algorithm are described in Algorithm 1. The PSC algorithm uses probabilistic selection based on the link conflict metric to determine the purification strategy and ensure the fidelity of E2E entanglement connections meets the requirements.

$$C^{flic}_{(u,v)} = \frac{w_i(u,v)}{\sum_i Q_i}, \quad \forall (u,v) \in \mathbf{P}_i. \tag{7}$$

Suppose $r_i(u, v)$ is the (integer) solution derived by probabilistic selection based on the link conflict metric to determine the purification strategy. Then the purification resource scheduling problem for a multi-request scenario, aiming to maximize the network throughput with desired entanglement fidelity, can be formulated as follows:

$$\text{Maximize} \quad \sum_{i=1}^{|\mathbf{R}|} x_i, \tag{8}$$

subject to:

$$\sum_i \left[r_i(u,v) + 1\right] x_i \le e_{uv}, \forall (u,v) \in \mathbf{P}_i, \tag{8a}$$

$$x_i \in \mathbb{Z}^+, \forall i \le |\mathbf{R}|. \tag{8b}$$

The problem becomes an integer linear programming problem. We first relax the integer constraint (8b) to continuous constraint and solve the derived linear programming model to obtain the solution $\hat{f}_i$ of the relaxation problem. Subsequently, the integer solution f_i is obtained by rounding the continuous solution $\hat{f}_i$, and then f_i is used to update the remaining entanglement resources on the entanglement path of each request. After the update, we try to allocate the remaining resources to the requests to increase the network throughput as much as possible. The algorithm ends when there are insufficient resources to support more entanglement connections. The algorithm will converge in a finite number of iterations. Notably, our designed PSC algorithm is also compatible with the phase-flip error. Because the phase-flip error can be converted into the bit-flip error with the unitary operation[31,32], only constraint (6e) is modified to consider the effect of additional introduced operations on entanglement fidelity. Besides, the PSC algorithm runs on a centralized controller, and the network clocks are synchronized and divided into different time slots. The centralized controller periodically collects link state information in the network and uses it as input to the PSC algorithm. Subsequently, the centralized controller informs each node on the entanglement path about the execution results of the PSC algorithm (i.e., purification and resource allocation strategies). Finally, each node performs the corresponding entanglement swapping based on the obtained purification and resource allocation strategies. When the link entanglement is suddenly unavailable in the network, it does not affect the normal execution of the PSC algorithm. Still, it may only lead to the failure to establish some entanglement connections because unavailable entanglement cannot be used to perform the corresponding purification and entanglement swapping.

Theorem 1. The PSC algorithm has a total time complexity of $O(NE + K(MN)^{2.373})$, where V is the number of nodes, E is the number of edges, N is the number of requests, M is the number of entangled paths contained in total by N requests, and K is the iterative run times of the algorithm.

Proof. Suppose the quantum network has V nodes, E edges, and N requests. The entanglement path chosen by the routing algorithm for N requests contains M edges ($M \le E$). Specifically, we will analyze the time complexity of the designed PSC algorithm in three parts: (1) The PSC uses Dijkstra's algorithm to find an entanglement path for each request and compute the link conflict metric value for each link on the path with a time complexity of $O(NElogV + NM)$. (2) The PSC makes randomized purification strategies based on the link conflict metrics and solves the corresponding relaxed linear programming problem with complexity $O(M + (NM)^{2.373})$ (when[33] solves relaxed linear programming, the linear programming solution with lowest time complexity is $O(NM)^{2.373}$ as far as we know). (3) The PSC tries to allocate the remaining resources on the link for each request with complexity $O(MN)$. It is worth noting that the second and third parts are run iteratively K times based on the designed algorithm parameters. For this reason, the total time complexity of the PSC algorithm is $O(NElogV + NM + K(M + (NM)^{2.373} + MN)) = O(NE + K(MN)^{2.373})$.

Data availability

The data sets generated during and or analyzed during the current study are available from the corresponding author on reasonable request.

Code availability

Methods, algorithms, and the value of parameters are fully described in the main text. The code has been publicly released at https://github.com/QNLab-USTC/PSC.

Received: 12 June 2023; Accepted: 6 September 2024;
Published online: 18 September 2024

References

1. Elliott, C., Pearson, D. & Troxel, G. Quantum cryptography in practice. In *Proc. 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* 227–238 (ACM, 2003).
2. Peev, M. et al. The SECOQC quantum key distribution network in vienna. *N. J. Phys.* **11**, 075001 (2009).
3. Sasaki, M. et al. Field test of quantum key distribution in the Tokyo QKD network. *Opt. Express* **19**, 10387–10409 (2011).
4. Yin, J. et al. Satellite-based entanglement distribution over 1200 kilometers. *Science* **356**, 1140–1144 (2017).
5. Wengerowsky, S. et al. Entanglement distribution over a 96-km-long submarine optical fiber. *Proc. Natl Acad. Sci. USA* **116**, 6684–6688 (2019).
6. Duan, L.-M., Lukin, M. D., Cirac, J. I. & Zoller, P. Long-distance quantum communication with atomic ensembles and linear optics. *Nature* **414**, 413–418 (2001).
7. Ekert, A. K. Cryptography and Bell's Theorem. In *Quantum Measurements in Optics* 413–418 (Springer, 1992).
8. Komar, P. et al. A quantum network of clocks. *Nat. Phys.* **10**, 582–587 (2014).
9. Broadbent, A., Fitzsimons, J. & Kashefi, E. Universal blind quantum computation. In *Proc. 2009 Annual IEEE Symposium on Foundations of Computer Science* 517–526 (IEEE, 2009).
10. Denchev, V. S. & Pandurangan, G. Distributed quantum computing: a new frontier in distributed systems or science fiction? *ACM SIGACT News* **39**, 77–95 (2008).
11. Dai, W., Peng, T. & Win, M. Z. Optimal remote entanglement distribution. *IEEE J. Select. Areas Commun.* **38**, 540–556 (2020).
12. Van Meter, R., Satoh, T., Ladd, T. D., Munro, W. J. & Nemoto, K. Path selection for quantum repeater networks. *Network. Sci.* **3**, 82–95 (2013).
13. Pant, M. et al. Routing entanglement in the quantum internet. *npj Quantum Inform*. **5**, 25 (2019).
14. Shi, S. & Qian, C. Concurrent entanglement routing for quantum networks: model and designs. In *Proc. 2020 ACM Special Interest Group on Data Communication* 62–75 (ACM, 2020).
15. Zhao, Y. & Qiao, C. Redundant entanglement provisioning and selection for throughput maximization in quantum networks. In *Proc. 2021 IEEE Conference on Computer Communications* 1–10 (IEEE, 2021).
16. Zhang, S., Shi, S., Qian, C. & Yeung, K. L. Fragmentation-aware entanglement routing for quantum networks. *J. Lightwave Technol.* **39**, 4584–4591 (2021).
17. Li, C., Li, T., Liu, Y.-X. & Cappellaro, P. Effective routing design for remote entanglement generation on quantum networks. *npj Quantum Inform.* **7**, 10 (2021).
18. Aparicio, L. & Meter, R. V. Multiplexing schemes for quantum repeater networks. In *SPIE Quantum Communications and Quantum Imaging IX*, **8163**, 59–70 (2011).

19. Li, J. et al. Fidelity-guaranteed entanglement routing in quantum networks. *IEEE Trans. Commun.* **70**, 6748–6763 (2022).
20. Van Meter, R., Ladd, T. D., Munro, W. J. & Nemoto, K. System design for a long-line quantum repeater. *IEEE/ACM Trans. Network.* **17**, 1002–1013 (2008).
21. Chen, Z. et al. Exponential suppression of bit or phase errors with cyclic error correction. *Nature* **595**, 383–387 (2021).
22. Riste, D. et al. Detecting bit-flip errors in a logical qubit using stabilizer measurements. *Nat. Commun.* **6**, 6983 (2015).
23. Chen, L. et al. SimQN: a network-layer simulator for the quantum network investigation. *IEEE Network* **37**, 182–189 (2023).
24. Van Meter, R. *Quantum networking* (John Wiley & Sons, 2014).
25. van Loock, P. et al. Extending quantum links: Modules for fiber-and memory-based quantum repeaters. *Adv. Quantum Technol.* **3**, 1900141 (2020).
26. Steger, M. et al. Quantum information storage for over 180 s using donor spins in a 28si “semiconductor vacuum”. *Science* **336**, 1280–1283 (2012).
27. Zhong, M. et al. Optically addressable nuclear spins in a solid with a six-hour coherence time. *Nature* **517**, 177–180 (2015).
28. Jiang, N. et al. Experimental realization of 105-qubit random access quantum memory. *npj Quantum Inform.* **5**, 28 (2019).
29. Chakraborty, K., Elkouss, D., Rijsman, B. & Wehner, S. Entanglement distribution in a quantum network: a multicommodity flow-based approach. *IEEE Trans. Quantum Eng.* **1**, 4101321 (2020).
30. Chiang, M., Low, S. H., Calderbank, A. R. & Doyle, J. C. Layering as optimization decomposition: a mathematical theory of network architectures. *Proc. IEEE* **95**, 255–312 (2007).
31. Sheng, Y.-B., Zhou, L. & Long, G.-L. Hybrid entanglement purification for quantum repeaters. *Phys. Rev. A* **88**, 022302 (2013).
32. Wang, G.-Y. et al. Faithful entanglement purification for high-capacity quantum communication with two-photon four-qubit systems. *Phys. Rev. Appl.* **10**, 054058 (2018).
33. Cohen, M. B., Lee, Y. T. & Song, Z. Solving linear programs in the current matrix multiplication time. *J. ACM* **68**, 1–39 (2021).

Acknowledgements

We thank Dr. Zhonghui Li and Dr. Lutong Chen for the fruitful discussions. This work is supported in part by the Innovation Program for Quantum Science and Technology under Grant No. 2021ZD0301301, Anhui Initiative in Quantum Information Technologies under Grant No. AHY150400, National Natural Science Foundation of China under Grant No. 62402466, and Youth Innovation Promotion Association of the Chinese Academy of Sciences (CAS) under Grant No. Y202093.

Author contributions

K.X. and N.Y. planned and supervised the project. Z.X. conceived the study and wrote the first draft of the manuscript. Z.X. and J.L. performed numerical analysis and conducted extensive experiments. J.L., K.X., N.Y., R.L., Q.S., and Jun Lu contributed to the discussion and the manuscript writing.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42005-024-01796-2.

Correspondence and requests for materials should be addressed to Kaiping Xue.

Peer review information *Communications Physics* thanks Changhao Li and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. A peer review file is available.

Reprints and permissions information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2024