

# REDP: Reliable Entanglement Distribution Protocol Design for Large-Scale Quantum Networks

Lutong Chen<sup>1</sup>, Graduate Student Member, IEEE, Kaiping Xue<sup>2</sup>, Senior Member, IEEE,  
Jian Li<sup>3</sup>, Senior Member, IEEE, Zhonghui Li<sup>4</sup>, Member, IEEE, Ruidong Li<sup>5</sup>, Senior Member, IEEE,  
Nenghai Yu<sup>6</sup>, Qibin Sun<sup>7</sup>, Fellow, IEEE, and Jun Lu<sup>8</sup>

**Abstract**—Remote entanglement distribution in an efficient and reliable manner, especially in the context of a large-scale quantum network with multiple requests, remains an unsolved challenge. The key difficulties lie in achieving spontaneous and precise control over the entanglement distribution procedure, as multiple nodes need to reach a consensus on how to perform it. From the network aspect, allocating link-layer entangled pairs as resources to achieve high efficiency is also challenging. To address these issues, we propose a decentralized Reliable Entanglement Distribution Protocol (REDP) for large-scale networks. The protocol operates in a Forward-Backward Propagation (FBP) manner, where consensus is reached hop-by-hop and disseminated to all nodes on the path. We further use probabilistic analysis and quasi-static modeling to seek the fairness and efficiency of the network based on the above transmission model. Accordingly, we introduce a Source Window Strategy (SWS) and an Entanglement Allocation Strategy (EAS) to assign sending windows and allocate resources for multiple requests, ensuring a high level of fairness and efficiency from a network perspective. Through systematic simulations involving both classical and quantum communication protocols, we demonstrate that REDP outperforms existing approaches in terms of fairness, throughput, and fidelity performance.

**Index Terms**—Entanglement distribution, protocol designing, quantum networks, resource allocation.

## I. INTRODUCTION

QUANTUM networks [1], [2] represent a cutting-edge network technology that enables the development of

Manuscript received 2 November 2023; revised 30 November 2023; accepted 20 December 2023. Date of publication 21 March 2024; date of current version 19 June 2024. The work of Lutong Chen, Kaiping Xue, Jian Li, Zhonghui Li, Nenghai Yu, Qibin Sun, and Jun Lu was supported in part by Anhui Initiative in Quantum Information Technologies under Grant AHY150300, in part by the National Natural Science Foundation of China (NSFC) under Grant 61972371 and Grant U19B2023, and in part by the Youth Innovation Promotion Association Chinese Academy of Sciences (CAS) under Grant Y202093. The work of Ruidong Li was supported in part by Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant 23H03380. (Corresponding author: Kaiping Xue.)

Lutong Chen, Kaiping Xue, Jian Li, Zhonghui Li, Nenghai Yu, and Qibin Sun are with the School of Cyber Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China (e-mail: kpxue@ustc.edu.cn).

Ruidong Li is with the Institute of Science and Engineering, Kanazawa University, Kakuma Campus, Kanazawa 920-1192, Japan.

Jun Lu is with the School of Cyber Science and Technology and the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, Anhui 230027, China.

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/JSAC.2024.3380101>.

Digital Object Identifier 10.1109/JSAC.2024.3380101

0733-8716 © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.  
See <https://www.ieee.org/publications/rights/index.html> for more information.

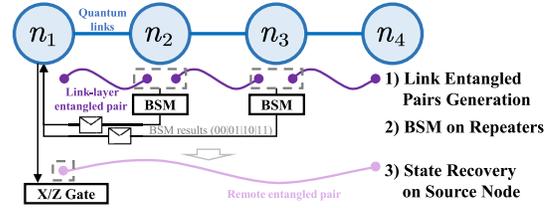


Fig. 1. The remote entanglement distribution procedure in a linear path.

novel quantum applications, including security key distribution with unconditional security [3], [4]. These networks can also establish connections between quantum computers, facilitating multi-party quantum computations [5], [6], which have the potential to solve mathematically challenging problems that classical computers cannot efficiently handle [7]. Additionally, quantum sensor networks [8] and clock synchronization [9] are also significant applications of quantum networks.

To enable these applications in quantum networks, a crucial function is the distribution of remote entangled pairs between non-neighboring nodes [10]. This process involves creating a specific quantum state known as a Bell-state entangled pair, where one qubit (usually a photon) resides in the source node, and the other qubit is located in the destination node. When two nodes are adjacent, it is relatively easier to achieve the goal by using a quantum link (such as optical fibers [11] or free space [12]). However, distributing entangled pairs between remote nodes presents challenges due to quantum imperfection [13]. To address this, quantum repeaters are introduced [14]. These repeaters sit between the source node and the destination node to perform entanglement swapping [15], which aggregates link-layer entangled pairs into end-to-end entangled pairs between the source and destination nodes. The procedure is illustrated in Fig. 1. The entanglement distribution process comprises three steps: Firstly, adjacent nodes generate link-layer entangled pairs continuously, which serve as link resources. Secondly, repeaters use Bell State Measurement (BSM) to combine link-layer entangled pairs into end-to-end entangled pairs between non-neighboring nodes. The BSM results are then sent back to the source node. Finally, the source node applies Pauli X or Z gates based on the BSM results to correct the remote entangled pair's state to the desired  $|\Phi^+\rangle$  Bell-state.

However, with many nodes cooperating in distributing entangled pairs in the network, it becomes crucial to have

a control mechanism that facilitates the coordinated usage of network resources among all the requests, particularly in large-scale networks with multiple concurrent requests [16]. This mechanism should allocate link resources properly to serve multiple simultaneous requests and instruct repeaters to perform the appropriate quantum operations on specific qubits. Ultimately, it should enable high network performance, fairness, and fidelity. Previous approaches [17], [18], [19], [20], [21] have mainly employed a centralized approach, which necessitates global information and a centralized controller. However, these approaches are susceptible to single-point failures and introduce heavy communication overhead, especially in large-scale networks. Therefore, this paper proposes a decentralized Reliable Entanglement Distribution Protocol (REDP) to schedule entanglement distribution spontaneously among the nodes.

Designing a decentralized mechanism is challenging, mainly due to the absence of a global view. Firstly, repeaters need to collaborate to reach a consensus on the distribution of entangled pairs. In other words, repeaters must determine the precise qubit on which the operations should be performed. This consensus must be reached without the involvement of a centralized controller. If a node exhibits unintended behavior, the remote entangled pair may not be distributed correctly. Secondly, each node independently decides on the allocation of resources, i.e., the number of link resources assigned to multiple requests, even though the network should achieve high throughput and fairness. Currently, link-layer entangled pairs are scarce due to physical device limitations and can serve only one request before vanishing. An inappropriate resource allocation may lead to massive costs in critical scenarios, such as decentralized quantum computation. Therefore, it is crucial to allocate resources accurately to achieve the best overall network performance. Thirdly, entangled pairs are susceptible to decoherence over time. Hence, this paper provides a decentralized protocol that can effectively and reliably complete remote entanglement distribution with a minor delay. This constraint is essential since upper-layer applications may require a high quality of distributed entangled pairs.

In this paper, we address the aforementioned challenges by presenting our REDP protocol. We first consider a scenario with a single request and introduce the transmission model utilized in this paper. It uses a Forward-Backward Propagation (FBP) protocol to achieve consensus on reliable entanglement distribution decisions among all nodes on the path while minimizing communication delays. Subsequently, we concentrate on providing network fairness and efficiency properties for the REDP protocol. The efficiency property necessitates that the network offers optimized overall throughput in the face of limited network resources. The fairness property guarantees that no requests are left starving. To ensure both fairness and efficiency, the quantum nodes must determine the optimal number of resources to allocate for multiple requests. Consequently, we design an Entanglement Allocation Strategy (EAS) for this purpose. We also propose a Source Window Strategy (SWS), which estimates the expected bandwidth, i.e., the total number of entangled pairs distributed in one FBP round, based on probabilistic-based modeling. The repeaters can utilize the

expected bandwidth as global information to provide better EAS results. All these algorithms are executed autonomously by the nodes. However, we prove that the SWS and EAS can be fair and efficient in theory.

We conduct extensive experiments in the systematic quantum network simulation platform named SimQN [22]. We construct full-stack evaluations for both quantum and classic networks and compare REDP with cutting-edge mechanisms. The results show that REDP provides better throughput, fairness, and fidelity, especially with multiple requests and an enormous network scale. The contributions of this paper are summarized as follows:

- We propose the REDP protocol for addressing the challenge of reliable quantum entanglement distribution in quantum networks. The protocol employs forward-backward propagation and is decentralized, allowing nodes to function autonomously. The protocol incurs negligible computation overhead and reduces fidelity degradation.
- We present a decentralized resource allocation algorithm that is proven to achieve fair property for multiple requests through probabilistic and quasi-static modeling. Furthermore, the resource allocation can be further optimized by utilizing an evaluated bandwidth to improve overall throughput efficiency.
- We conduct extensive full-stack network simulations that compare our proposal with baselines and other cutting-edge approaches. The results illustrate the advantages of our decentralized protocols in multiple requests large-scale situations.

This paper is organized into the following sections. First, we brief the related work in Sec. II. Next, in Sec. III, we present the system model and the problem statement. Then, in Sec. IV, we introduce the design of the REDP protocol and the corresponding EAS and SWS algorithms to guarantee efficiency and fairness. In Sec. V, we conduct solid experiments with the SimQN platform with several typologies, and finally, we conclude our work in Sec. VI.

## II. RELATED WORK

Entanglement-based quantum networks are novel network structures providing several significant applications, including distributed cryptography functions, distributed computation, and sensor networks [23]. As a critical function, the remote entanglement distribution problem plays a vital role in these networks [10], [23], [24]. Currently, several physical experiments have been conducted to illustrate the feasibility of entanglement distribution [25], [26], [27]. Therefore, cutting-edge research is beginning to consider the effectiveness of remote entanglement distribution.

Several previous works focused on the efficient entanglement distribution for one request [21], [28], [29], [30] or scheduling in a specific topology [31], [32]. They mainly consider the detailed procedure of the entanglement distribution to achieve a better entanglement distribution rate.

Furthermore, since the network typically serves multiple requests, other works have studied the scenario of multiple

requests. These works can be classified into two categories, namely, centralized mechanisms and decentralized protocol approaches. The first category has received more attention, and several contributions have been made [17], [18], [19], [20]. These works have proposed centralized mechanisms to manage the allocation of entanglement resources. For instance, Li et al. [19] proposed three algorithms, namely Proportional Share (PS), Progressive Filling (PF), and Propagatory Update (PU), for handling resource allocation. PS algorithm assigns link resources locally and in proportion among requests on the links. PF algorithm adopts the Bertsekas' algorithm [33] to guarantee fairness, while the PU algorithm utilizes global scheduling and assigns allocation in a backward manner to achieve a higher throughput. Additionally, Li et al. designed an entanglement distribution algorithm that guarantees fidelity [34]. However, centralized mechanisms may suffer from single-point failure and longer signal delay.

Thus, this paper focuses on the second category, designing a decentralized protocol that can allocate resources effectively and spontaneously. However, due to their complexity, fewer mechanisms have been proposed in this category. Yu et al. [35] proposed a protocol that mainly focuses on transmitting qubits rather than distributing remote entangled pairs. Similarly, Kozłowski et al. [36] considered distributing entanglement for one request. On the other hand, Zhao and Qiao [30] proposed Tele-DTP for remote entanglement distribution, which is the closest work to this paper as it also addresses the issues of allocating resources to achieve high performance and fairness. It assigns memories among all nodes on the path and generates link-layer entangled pairs on demand. However, both [30] and [36] use a link model that generates link-layer entangled pairs on demand, which may result in longer waiting times (in link resources generation) and less fidelity. Therefore, REDP uses the same link-layer model in [19] that continuously generates entangled pairs on the links and fine-grained manages these resources.

### III. SYSTEM MODEL

In this section, we describe the network models, the request model, and the problem statement.

#### A. Network Models

We represent a quantum network using an undirected graph  $G = \{V, E, C\}$ , where  $V = \{n_i | i \leq N\}$  denotes the set of quantum nodes. Here,  $n_i$  is  $i$ -th node, and  $N$  is the total number of nodes. The set  $E = \{l_{i,j}\}$  denotes a set of quantum links, where  $l_{i,j}$  represents a quantum link (usually an optical fiber) between  $n_i$  and  $n_j$  that can transmit qubits. Additionally, we consider the limited quantum memory capacity, where quantum memories can store a qubit for a while. Two neighboring nodes  $n_i$  and  $n_j$  assign  $C_{i,j} \in C$  memories to the link  $l_{i,j}$  so that at most  $C_{i,j}$  link-layer entangled pairs can exist simultaneously. In this paper, we do not require the topology  $G$  to be a global knowledge to all nodes. Also, it can change over time. However, since quantum networks are typically connected by optical fibers, the topology will not undergo sudden and drastic changes.

It is important to note that the quantum network is not intended to replace the classical network but utilizes it to transmit control messages. In this paper, the classical topology is identical to the quantum topology. That is, if two nodes are connected by a quantum channel, they can also operate classical communication. Note that it is not a prerequisite, as the proposed protocol does not require the classical topology to be identical. Furthermore, we notice that classical communication can affect the quality of entanglement distribution. For example, a longer propagation delay or congestion may reduce the number of distributed entangled pairs. To evaluate the impact of classical communication, we introduced a variable single-hop delay  $t$  (in a normal distribution) in our experiments, which measures the completion time of control message transmission between two neighbor nodes. It evaluates the transmission delay and congestion in the classical network.

In this paper, the quantum network should serve multiple requests concurrently to be efficient. A request refers to a pair of two quantum nodes that aim to distribute entangled pairs between them. The set of requests, denoted by  $R = \{r^{(k)} | k \leq K\}$ , contains all requests, where  $r^{(k)}$  denotes the  $k$ -th request, and  $K$  represents the total number of requests. Specifically, a request  $r^{(k)}$  can be represented as  $r^{(k)} = (s^{(k)}, d^{(k)}, \mathcal{P}^{(k)})$ , where  $s^{(k)}$  denotes the source node,  $d^{(k)}$  denotes the destination node, and  $\mathcal{P}^{(k)}$  denotes a specific routing path that connects  $s^{(k)}$  and  $d^{(k)}$ . In this paper, we assume a routing algorithm exists to find the path  $\mathcal{P}^{(k)}$  for the two nodes. However, we do not specify a particular routing algorithm due to decoupling considerations. For instance, the OSPF [37], adaptive routing algorithm [38], or the Yen's multiple-path algorithms [39] can be utilized. Furthermore, the request set  $R$  can be local information, and only the nodes in  $\mathcal{P}^{(k)}$  are aware of the existence of the request.

#### B. Link-Layer Entangled Pair Generation Models

In this paper, the entangled pairs between neighbor nodes are referred to as link-layer entangled pairs and are regarded as valuable network resources for distributing end-to-end entangled pairs. The generation of link-layer entangled pairs is controlled by a link-layer entanglement generation protocol [40] that runs on each node. These entangled pairs are generated continuously by an SPDC technology [41], and the generation rate on link  $l_{i,j}$  is denoted by  $v_{i,j}$ . It represents the link can generate at most  $v_{i,j}$  entangled pairs per second. To account for the SPDC inefficiency [41] and signal attenuation that occurs during generation, we use  $p_{i,j}$  to represent the generation success rate:

$$p_{i,j} = (1 - p_{\text{init}}) \cdot 10^{-\mu D_{i,j}/10}, \quad (1)$$

where  $p_{\text{init}}$  is the initial loss rate from the SPDC inefficiency,  $D_{i,j}$  is the link length, and  $\mu$  is the attenuation factor in  $\text{db/km}$ . Consequently, during  $T$  seconds, the number of the generated entangled pairs follows a binomial distribution  $B(v_{i,j}T, p_{i,j})$ , and the expected number is  $v_{i,j}p_{i,j}T$ . Considering that the link can hold at most  $C_{i,j}$  entangled pairs, the link-layer protocol continuously generates entangled pairs until the memory is full.

Fidelity (a value between 0 and 1) is the purity of entangled pairs compared to the ideal entangled state  $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ . Fidelity drops due to transmissions over quantum links, storage at memories, and operation noise. To mitigate the fidelity downgrade, entanglement distillation protocols [42] have been introduced to improve fidelity by combining multiple entangled pairs into one entangled pair with a success probability. Similar to [19], we use a layered network structure and view the link-layer protocol as a black box to provide high-fidelity link-layer entangled pairs. Therefore, we refer  $v_{i,j}$  and  $p_{i,j}$  to its capacity to provide high-fidelity entangled pairs in the following parts of the paper.

### C. Problem Statement

The fundamental purpose of quantum networks in this paper is to distribute entangled pairs upon multiple requests. Therefore, we design a REDP protocol to control the entanglement distribution procedure precisely and efficiently.

To overcome the attenuation of the long-distance transmission, a relay technique named entanglement swapping is introduced [15]. As mentioned in [43], this procedure has three steps, as shown in Fig. 1.

Step 1, link-layer entangled pairs are generated by all nodes on a path  $\mathcal{P}^{(k)}$  continuously with adjacent nodes, following the instructions of a link-layer protocol [40]. Each link-layer entangled pair contains two qubits, one stored on the predecessor node and the other sent to the successor node. These link-layer entangled pairs are supposed to be assigned to serve multiple requests as the resource for distributing remote entangled pairs.

Step 2, Bell State Measurement (BSM) on repeaters. The repeaters perform a BSM to aggregate link-layer entangled pairs into one end-to-end entangled pair. After that, repeaters send the measurement result to  $s^{(k)}$ . To ensure minimal delay and achieve high fidelity, all repeaters perform BSMs concurrently. It is worth noting that BSM operations may occasionally fail. However, due to the heavy communication overhead in the decentralized network, handling such failures is challenging until the next step.

Step 3, state recovery on the source node. If repeaters perform BSMs successfully, the entangled pair is expected to collapse into one of the four Bell states, namely,  $|\Phi^+\rangle$ ,  $|\Phi^-\rangle$ ,  $|\Psi^+\rangle$ , and  $|\Psi^-\rangle$ . The source node will then proceed to recover the entanglement state into the targeted  $|\Phi^+\rangle$  state. Specifically, the source node collects all measurement results and applies either a Pauli X or Z gate, depending on the measurement outcome.

The protocol we design aims to facilitate the reliable distribution of remote entangled pairs by controlling the aforementioned procedures. To be more specific, the protocol must efficiently allocate link-layer entangled pairs to serve multiple requests in Step 1. It should then instruct the repeaters to perform the BSMs correctly. Finally, the protocol should guide the source node to recover the entanglement state. We have identified two remaining challenges, which can be summarized as follows.

1) *Consistent Consensus*: all nodes on a path  $\mathcal{P}^{(k)}$  must reach a consensus on entanglement distribution. Since multiple

qubits exist on the nodes, each node should know the specific operations to be conducted on individual qubits to ensure the correct distribution of remote entangled pairs and prevent unexpected quantum states. The main challenge lies in establishing this consensus among quantum nodes without the involvement of a controller.

2) *Resource Allocation*: The link-layer entangled pairs are used as network resources to serve multiple requests. The REDP protocol should allocate the resources for multiple requests properly. On the one hand, the resources should be used efficiently to achieve high throughput. On the other hand, the resources should be allocated fairly to all requests to avoid the massive costs of starvation in some scenarios like distributed quantum computation.

## IV. REDP PROTOCOL DESIGN

### A. Overview

REDP is a reliable entanglement distribution protocol to control the entanglement distribution procedure for multiple concurrent requests precisely. Initially, we focused on single-request scenarios and devised a Forward Backward Protocol (FBP) to establish a consensus on entanglement distribution among all nodes along the entire path. Briefly, the forward phase reserves link-layer resources, and it ensures that all links reserve the same number of resources. No entanglement swapping happens in this phase. The backward phase sends the path-level consensus to all the nodes on the path, and the repeaters enter the entanglement distribution phase to operate the actual entanglement distribution operations. This protocol is discussed in detail in Sec. IV-B.

Next, we present our approach to servicing multiple requests, which aims at enhancing network efficiency. Therefore, we design two resource allocation strategies (i.e., EAS and SWS) to guarantee the fairness and efficiency properties in Sec. IV-C and Sec. IV-D, respectively. SWS runs on the source node to evaluate the expected bandwidth  $w_{\text{src}}^{(k)}$  as the global information based on our probabilistic modeling. EAS, on the other hand, is the core resource allocation strategy, which decides the resource allocation for each request using the information provided by SWS. EAS guarantees fairness and efficiency from the network's aspect. Here, fairness means that multiple requests should use link-layer entangled pairs equally, while efficiency requires complete resource utilization to achieve the best throughput.

### B. REDP-FIFO: Entanglement Distribution Protocol on the Multiple-Hop Path

This section demonstrates how REDP operates using a Forward-Backward Protocol (FBP) in a single-request scenario. It is imperative for the repeaters to identify the two qubits to perform BSMs, and for the source node to understand how to recover the entangled pair's state. The FBP is devised to achieve the consensus on the entire path to establish a reliable entanglement swapping process. It runs in loop rounds, and each round cycles in three phases: forward phase, backward phase, and entanglement distribution phase. The forward and backward phases are intended for assigning link-layer

entangled pairs and achieving consensus on entanglement distribution among all nodes on the path. In the entanglement distribution phase, the actual entanglement swapping takes place, with the repeaters performing BSMs and sending the measurement results to the source node.

First, we use the example of a one-hop link to introduce the consensus and explain how FBP achieves it. Consider a one-hop link  $l_{i,j}$  that connects node  $n_i$  and  $n_j$ . Link-layer entangled pairs are generated continuously by a link-layer protocol whenever the quantum memories are not full. Let us assume the link contains  $C_{i,j}$  link entangled pairs. Note that the entangled pair contains two qubits stored on the two adjacent nodes. As a result, the two nodes must agree on resource allocation. The consensus mainly contains two things: 1) Considering that multiple entangled pairs exist on the link, the consensus should assign which entangled pairs to use. 2) Considering multiple requests pass through the link, the consensus should assign the link resources to one specific request. A basic FBP can reach this agreement. In the Forward phase,  $n_i$  selects  $w^{(k)}$  number of link entangled pairs ( $w^{(k)} < C_{i,j}$ ). It then sends the  $w^{(k)}$  and the IDs of the selected entangled pairs to  $n_j$ . Let the applying set  $a_{i,j}^a$  be the set of the IDs of the selected entangled pairs. Second,  $n_j$  checks the set  $a_{i,j}^a$ . If an entangled pair in  $a_{i,j}^a$  has not been assigned before and  $n_j$  agrees to use it,  $n_j$  will put the ID of the entangled pair into a permitted set  $a_{i,j}^p$ . We denote  $a_{i,j}^p$  as the permitted set that contains IDs of the entangled pairs that both  $n_i$  and  $n_j$  agree to use. Otherwise,  $n_j$  cannot put the ID in  $a_{i,j}^p$ . Finally,  $n_j$  sends  $a_{i,j}^p$  back to  $n_i$  in the backward phase, and now  $n_i$  and  $n_j$  agree that entangled pairs in  $a_{i,j}^p$  can be used for the request  $r^{(k)}$ .

In this way, a consensus can be reached, and no conflict will exist. The two nodes cannot use different entangled pairs, and an entangled pair cannot be assigned to multiple requests. This is because both nodes are permitted to use the entangled pair. Specifically,  $n_i$  is permitted to use entangled pairs in  $a_{i,j}^p \subseteq (a_{i,j}^a \in a_{i,j}^a)$  in the forward phase, and  $n_j$  is permitted to use entangled pairs in  $a_{i,j}^p$  during the backward phase.

Now, let us present how FBP works when the path consists of more than one hop and how to control the entanglement distribution reliably. Similarly, two adjacent nodes decide the resource allocation on the connected link. During the forward phase, packets are transmitted hop-by-hop from the source node to the destination. When the predecessor sends a forward packet to the successor node, it selects some link-layer entangled pairs to use based on the EAS strategy mentioned in the next section. Upon receiving the packet, the successor node verifies the applying set and approves some entangled pairs. Now, the link-layer agreement is reached, and the successor node then allocates resources on the subsequent link and transmits a new applying set to the following node. The backward phase starts when the destination node receives the message. All links have reached a link-layer consensus about resource assignment by this point. Therefore, during the backward phase, all the consensus is transmitted back to all nodes on the path to achieve path-level consensus. In the entanglement distribution phase, the repeaters execute BSM based on the consensus and transmit the measurement result to

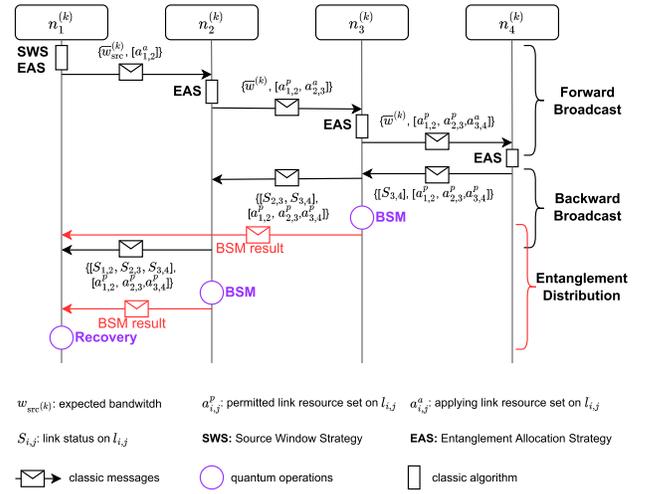


Fig. 2. REDP protocol on a 4-node path.

the source node. The source node then recovers the entangled pair's state. Once the remote entangled pairs are distributed, a new round of FBP can start.

Fig. 2 shows the FBP protocol on an example with a 4-node path. Here, we present the details of each phase. Consider a request  $r^{(k)} = (s^{(k)}, d^{(k)}, \mathcal{P}^{(k)})$ , it requires to distribute entangled pairs between the source node  $s^{(k)}$  and  $d^{(k)}$ . Moreover, a routing algorithm selects a  $L$ -hop path  $\mathcal{P}^{(k)}$ . Without loss of generality, the  $i$ -th node in the path  $\mathcal{P}^{(k)}$  is named  $n_i$  for convenience, i.e.,  $s^{(k)} = n_1$  and  $d^{(k)} = n_{L+1}$ , where  $L$  is the number of the links in  $\mathcal{P}^{(k)}$ . The link  $l_{i,i+1}$  connects  $n_i$  and  $n_{i+1}$ . Furthermore, we use the notation  $LR_{i,i+1}$  to denote the link resources (link-layer entangled pairs) on the link  $l_{i,i+1}$ .

In REDP, the link resources have two states: 1) “allocated,” indicating that the entangled pair has been assigned to a specific request, and 2) “free,” suggesting that the entangled pair has not yet been assigned. As a decentralized protocol, the states are maintained locally by the nodes. Two nodes may view a single entangled pair in different states. For instance, if node  $n_i$  assigns to use an entangled pair, it labels the pair as “allocated,” but the other node may still label the same entangled pair as “free” before a consensus is reached.

The forward phase starts from the source node  $n_1$ , which utilizes the entanglement allocation strategies (EAS) to select entangled pairs from the available link resources  $LR_{1,2}$  which will be discussed in Sec. IV-C and Sec. IV-D. Let the applying set  $a_{1,2}^a$  represent the selected entangled pairs, and  $w_1 = |a_{1,2}^a|$  denote the number of applying resources. Subsequently,  $n_1$  marks the entangled pairs in  $a_{1,2}^a$  as “allocated” to prevent other requests from using the same entangled pairs. Finally,  $n_1$  sends the forward packet containing  $a_{1,2}^a$  to the next-hop  $n_2$ . The operations on the source node in the forward phase are described in Algorithm 1.

When node  $n_2$  receives the forward packet from  $n_1$ , it handles two tasks. Firstly, it verifies the applying set  $a_{1,2}^a$  received from  $n_1$ . Specifically,  $n_2$  checks if the link-layer entangled pair is in a “free” state. If the entangled pair is free,  $n_2$  permits the assignment of this entangled pair. However, if the entangled pair is already “allocated,” a conflict arises,

---

**Algorithm 1** The Source Node's Action in Forward Phase
 

---

**Input:** The source node  $n_1$ ; The next-hop node  $n_2$ ;  
The path  $\mathcal{P}^{(k)} = \{n_j\}$ ;

- 1  $LR_{1,2} \leftarrow l_{1,2}.get\_link\_layer\_resources()$ ;
- 2  $a_{1,2}^a \leftarrow EAS(LR_{1,2})$ ;  $w_1 \leftarrow |a_{1,2}^a|$ ;
- 3  $msg \leftarrow \{a_{1,2}^a\}$ ;
- 4  $n_1.send(n_2, msg)$ ;

---

and  $n_2$  cannot allow the use of this entangled pair. As a result, some entangled pairs in  $a_{1,2}^a$  are agreed upon to be used, and the final permitted set is denoted as  $a_{1,2}^p \subset a_{1,2}^a$ . Then, node  $n_2$  applies resources on the next link  $l_{2,3}$  and generates a new applying set  $a_{2,3}^a$ . Finally,  $n_2$  sends both  $a_{1,2}^p$  and  $a_{2,3}^a$  to the next node  $n_3$ .

Generally, when a repeater (denoted as  $n_i$ ) receives a forward packet, it contains two parts. The first part is the applied resource set  $a_{i-1,i}^a$  from the last-hop node,  $n_{i-1}$ , and the second part is the permitted set  $\{a_{j,j+1}^p | 1 \leq j < i-1\}$  on the link from  $l_{1,2}$  to  $l_{i-2,i-1}$ , where  $i > 2$ . Similar to  $n_2$ ,  $n_i$  performs two tasks: it checks the resource allocation in  $a_{i-1,i}^a$  to produce a  $a_{i-1,i}^p$  and applies resources on the next link as  $a_{i,i+1}^a$ . Algorithm 2 presents a formal description of the repeaters' actions in the forward phase, which consists of four steps:

- **Step 1**,  $n_i$  checks whether the entangled pairs in  $a_{i-1,i}^a$  are in a “free” state and allows the free ones to be utilized by the current request  $r^{(k)}$ , thereby forming the permitted set  $a_{i-1,i}^p$ . Consequently,  $n_{i-1}$  and  $n_i$  both acknowledge that the entangled pairs within  $a_{i-1,i}^p$  are assigned for the current request.
- **Step 2**, it applies link resources to the next link by utilizing EAS to choose “free” entangled pairs on link  $l_{i,i+1}$  and placing the IDs of the selected entangled pairs into a new applying set  $a_{i,i+1}^a$ .
- **Step 3**, the node guarantees that the newly formed applying set for the next link and the permitted sets for the previous links have the same size. Considering a bottleneck link allocates  $w$  link resources, at most  $w$  remote entangled pairs can be distributed, and allocating more resources than  $w$  on other non-bottleneck links will result in unused and wasted. Therefore, the repeaters must ensure that all permitted sets  $a_{j,j+1}^p$  ( $j < i-1$ ) for the previous links and the applying set  $a_{i,i+1}^a$  for the next link are truncated to the same sizes. Let  $w_{\min} = \min\{|a_{i,i+1}^a|, |a_{1,2}^p|, \dots, |a_{i-1,i}^p|\}$  be the minimized size (resource allocated on the bottleneck link). Once a set has more elements than  $w_{\min}$ ,  $n_i$  will drop some elements to shrink the size to  $w_{\min}$ . Specifically, for all permitted sets on the previous links in  $\{a_{j,j+1}^p | j \leq i-1\}$ , if  $|a_{j,j+1}^p| > w_{\min}$ ,  $n_i$  truncates  $a_{j,j+1}^p$  to  $a_{j,j+1}^p[:w_{\min}]$ , where  $[-x:]$  denotes selecting the first (last)  $x$  entangled pair from a list, respectively. This step ensures that all links allocate the same number of resources. Meanwhile, as we will explain in the backward phase, the truncated

---

**Algorithm 2** The Repeater Handling in Forward Phase
 

---

**Input:** The current node  $n_i$ ; The next-hop node  $n_{i+1}$ ;  
The path  $\mathcal{P}^{(k)} = \{n_j | 1 \leq j \leq L+1\}$ ;  
The received forward packet  $\{a_{j,j+1}^p | j < i-1\} \cup \{a_{i-1,i}^a\}$ ;

// Step 1: Check allocation on  $l_{i-1,i}$

- 1  $a_{i-1,i}^p = \{\}$ ;
- 2 **for**  $qubit \in a_{i-1,i}^a$  **do**
- 3     **if**  $qubit$  is “free” **then**
- 4          $a_{i-1,i}^p = a_{i-1,i}^p \cup \{qubit\}$ ;
- 5     **end**
- 6 **end**

// Step 2: Apply allocation on  $l_{i,i+1}$

- 7  $LR_{i,i+1} \leftarrow l_{i,i+1}.get\_link\_layer\_resources()$ ;
- 8  $a_{i,i+1}^a \leftarrow EAS(LR_{i,i+1})$ ;

// Step 3: Drop the extra entangled pairs

- 9  $w_{\min} \leftarrow \min\{|a_{i,i+1}^a|, |a_{1,2}^p|, \dots, |a_{i-1,i}^p|\}$ ;
- 10 **if**  $w_{\min} < |a_{i,i+1}^a|$  **then**
- 11      $a_{i,i+1}^a \leftarrow a_{i,i+1}^a[:w_{\min}]$
- 12 **end**
- 13 **for**  $j \in [1, i-1]$  **do**
- 14     **if**  $w_{\min} < |a_{j,j+1}^p|$  **then**
- 15          $a_{j,j+1}^p \leftarrow a_{j,j+1}^p[:w_{\min}]$ ;
- 16     **end**
- 17 **end**

// Step 4: State update and send forward message

- 18 **for**  $qubit \in \{a_{i,i+1}^a \cup a_{i-1,i}^p\}$  **do**
- 19     mark  $qubit$  as “allocated”;
- 20 **end**
- 21  $msg \leftarrow \{a_{j,j+1}^p | j < i\} \cup \{a_{i,i+1}^a\}$ ;
- 22  $n_i.send(n_{i+1}, msg)$ ;

---

resources will be noticed by the repeaters and thus can be used for further requests.

- **Step 4**, the repeater marks the entangled pairs in the new applying set  $a_{i,i+1}^a$  as “allocated” and sends the new forward message  $\{a_{i,i+1}^a\} \cup \{a_{j,j+1}^p | j < i\}$  to the next-hop node  $n_{i+1}$ .

The backward phase starts immediately after the destination node  $n_{L+1}$  receives the forward packet. Similar to its predecessor repeaters,  $n_{L+1}$  also inspects  $a_{L,L+1}^a$  and permits to use of “free” entangled pairs in  $a_{L,L+1}^a$ . Additionally, the destination node ensures that the size  $a_{j,j+1}^p$  ( $j \leq L$ ) remains identical, similar to Step 3 of Algorithm 2. At this point, the path has produced a resource allocation decision in the form of the permitted sets for all the links on the path  $\{a_{j,j+1}^p | j \leq L\}$ . During the backward phase,  $n_{L+1}$  sends the entanglement distribution decision  $\{a_{j,j+1}^p | j \leq L\}$  hop-by-hop until the source node  $n_1$ , and all nodes on the path are aware of the link resource allocation, thus forming a path-level resource allocation consensus. Note that REDP allows the applying set and the permitted set to be empty, indicating that there are no resources available at the current time. However, since REDP runs in multiple rounds and will not finish until it fulfills the

requests, it is possible for the future to have enough resources to serve them.

The entanglement distribution phase is the last stage of one FBP round. When a repeater, denoted as  $n_i$ , receives the consensus  $\{a_{j,j+1}^p | j \leq L\}$  in the backward packet, it enters this phase. It selects one entangled pair from the previous link's permitted set  $a_{i-1,i}^p$  and another from the next link's permitted set  $a_{i,i+1}^p$ . The repeater then performs BSM using the selected two entangled pairs. It is noteworthy that all permitted sets have the same size, guaranteed by Step 3 of Algorithm 2. Thus, the repeater can use the two entangled pairs with the same position in  $a_{i-1,i}^p$  and  $a_{i,i+1}^p$ . Let  $a_{i,i+1}^p[x]$  be the  $x$ -th entangled pair in the permitted set  $a_{i,i+1}^p$ , the repeater performs BSM using the entangled pairs  $a_{i-1,i}^p[x]$  and  $a_{i,i+1}^p[x]$  ( $\forall x \leq w_{\min}$ ). Afterward, the repeater modifies the state of the unused entangled pairs marked as "free" again. These unused entangled pairs result from two aspects. Firstly, the previous node applies to use an entangled pair, but the following node does not permit it. Secondly, some entangled pairs are revoked for being used in Step 3 of Algorithm 2 to maintain all permitted sets' size consistency. Subsequently, the repeaters send measurement results to the source node. When a source node collects all measurement results from all repeaters, it can perform Pauli Z or X gate on the qubit it owns and forms the targeted  $|\Phi^+\rangle$  entangled state between  $n_1$  and  $n_{L+1}$ .

1) *The Correctness and Reliability of FBP Protocol:* We prove that the FBP protocol is reliable in three aspects: consistent results, conflict avoidance, and guaranteed fidelity. (1) consistent results, the FBP protocol guarantees consistent results by ensuring that all nodes on the path reach the same consensus. It is because the destination node generates the final path-level entanglement distribution consensus and notifies all nodes on the path in the backward phase. (2) conflict avoidance, the FBP protocol ensures conflict avoidance by preventing one entangled pair from being used by multiple requests. Only when both adjacent nodes allow an entangled pair to be used for the current request and be marked "allocated" immediately. (3) guaranteed fidelity, the FBP protocol guarantees fidelity by minimizing time delay and the number of quantum operations. When the path's length is fixed, the fidelity of remote entangled pairs is influenced by time delay. Since a repeater enters the entanglement distribution phase once it receives a backward message, the entanglement distribution and the backward phases overlap, as shown in Fig. 2. It ensures the REDP protocol achieves the optimal time delay of  $2L \cdot t$ , where at least  $L \cdot t$  is required to form a consensus on an  $L$ -hop path and another  $L \cdot t$  is needed for entanglement distribution.

2) *Achieve Optimal Performance in Single-Request Scenarios Using FBP:* In this scenario, we present the first version of Entanglement Allocation Strategy (EAS) and prove that the maximum bandwidth (number of remote entangled pairs distributed in one round) can be achieved in single-request scenarios. This version of EAS assigns all available "free" entangled pair resources on the next-hop link, i.e.,  $a_{i,i+1}^a = LR_{i,i+1}$ . When the bottleneck link is  $l_{i,i+1}$ , i.e., the link generating the least amount of entangled pairs, the final

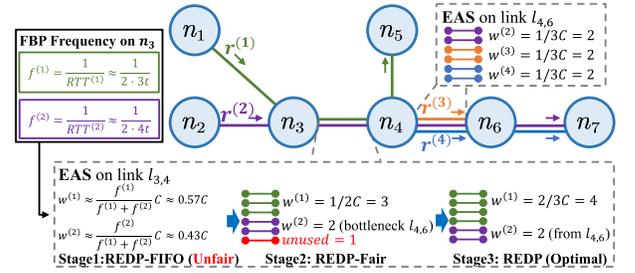


Fig. 3. Examples of fair and efficient link resource allocation ( $C = 6$ ).

throughput is  $w = w_i = |LR_{i,i+1}|$ . As  $LR_{i,i+1}$  represents all entangled pair resources on link  $l_{i,i+1}$ , the bandwidth upper bound is  $\min_{0 \leq j \leq L} \{|LR_{j,j+1}|\} = |LR_{i,i+1}|$ , which is achievable using the EAS. In the scenario with multiple requests, the first coming request will apply all resources and requests will assign resources in a FIFO queue manner. Therefore, we named it REDP-FIFO.

### C. REDP-Fair: Fairness Guarantee in Multiple Request Scenario

In Sec. IV-B, we design the first version of the REDP protocol and prove its correctness and performance in a single-request scenario. However, it cannot provide fairness and optimal performance when multiple requests exist due to the lack of a proper resource allocation strategy. Therefore, in this section, we propose an improvement to provide the fairness property with a probabilistic quasi-static model. In Sec. IV-D, we further enhance the resource utilization and throughput performance.

The fairness property requires that if multiple requests pass through the same bottleneck link [44], [45], they should equally use the link-layer entangled pair resources, no matter its path length. In FBP, the repeater assigns resources whenever it receives a forward packet. Obviously, REDP-FIFO cannot guarantee fairness, for the first-come request uses all link-layer entangled pair resources, and the later requests may starve. We observe that the resource allocation is affected by the length of the path. Considering an example of two requests  $r^{(1)}$  and  $r^{(2)}$  in Fig. 3. Request  $r^{(1)}$  distributes entangled pairs between  $n_1$  and  $n_5$ , while  $r^{(2)}$  distributes entangled pairs between  $n_2$  and  $n_7$ . Two requests share the same bottleneck link  $l_{3,4}$ .  $r^{(1)}$  takes  $2 \times 3t$  times on average to finish one FBP round, but  $r^{(2)}$  takes roughly  $2 \times 4t$  times to complete one FBP round. Note that resource allocation happens whenever the node receives a forward packet. As a result,  $r^{(1)}$  has a faster frequency of applying link resources on link  $l_{3,4}$ , which leads to unfair resource allocation.

One way to address this issue is to allocate fewer link resources for requests with fewer path hops. However, this is challenging in a decentralized network for two primary reasons. First, nodes have limited information about when they will receive a forward message and from which request the message is originating. Second, nodes cannot predict the number of link-layer entangled pair resources that will be generated in the future, as it is a stochastic process mentioned in Sec. III-B.

Despite these challenges, we design a new version of the REDP protocol, REDP-Fair, and prove its fairness in the long-term time perspective. The critical task is to achieve fairness. To this end, a proper EAS is responsible for deciding how many new resources should be assigned to a request when a repeater receives a forward packet. In REDP-Fair, the EAS strategy is to select  $\alpha_{\text{fair}}^{(k)} \cdot |LR_{i,j}|$  resources, where the fair factor  $\alpha_{\text{fair}}^{(k)}$  is

$$\alpha_{\text{fair}}^{(k)} = \frac{L^{(k)}}{\sum_{r^{(q)} \in R_{i,j}} L^{(q)}}, \quad (2)$$

where  $L^{(k)}$  is the number of hops for the request  $r^{(k)}$ , and  $R_{i,j}$  is the set of all requests that pass through the next-hop link  $l_{i,j}$ .

Now, we prove that REDP-Fair ensures the fairness property in a long-term time perspective by utilizing a quasi-static model. The fundamental concept of the quasi-static model analysis is to model both the producer aspect (which generates the link-layer entangled pairs) and the consumer aspect (which allocates resources). Once the system reaches equilibrium, the rate of link resource generation equals consumption, enabling us to evaluate the average resource allocation for each request in the long term. Subsequently, we prove that when the fair factor follows Eq. (2), it guarantees the fair utilization of link resources for all requests.

For a request  $r^{(q)}$  that passes the link  $l_{i,j}$ , it takes approximately  $2L^{(q)}t$  time to complete one FBP round, as demonstrated in Sec. IV-B, and its frequency is  $f^{(q)} = \frac{1}{2L^{(q)}t}$ . Considering that the resource allocations happen when the node receives a forward packet, the frequency of node  $n_i$  and  $n_j$  allocating link resources for all requests is  $\sum_{r^{(q)} \in R_{i,j}} f^{(q)} = \sum_{r^{(q)} \in R_{i,j}} \frac{1}{2L^{(q)}t}$ .

From the link-layer entangled pair generation aspect, during the time between each resource allocation, the link  $l_{i,j}$  generates

$$\overline{a_{i,j}^{\text{prod}}} = \frac{v_{i,j}p_{i,j}}{\sum_{r^{(q)} \in R_{i,j}} f^{(q)}} = \frac{2tv_{i,j}p_{i,j}}{\sum_{r^{(q)} \in R_{i,j}} 1/L^{(q)}} \quad (3)$$

link-layer entangled pair resources on average. Remember that  $v_{i,j}$  represents the rate of entangled pair generation, and  $p_{i,j}$  denotes the possibility of successful entanglement generation between adjacent nodes.

From the resource consumption perspective, we model the average number of assigned entangled pairs. Assuming that the link has an average of  $|LR_{i,j}|$  entangled pair resources when the system is stable, the repeater will assign  $a_{i,j}^{(k)} = \alpha_{\text{fair}}^{(k)} |LR_{i,j}|$  entangled pairs for the  $k$ -th request  $r^{(k)}$ . Therefore, the expected number of assigned resources is

$$\overline{a_{i,j}^{\text{alloc}}} = \sum_{r^{(k)} \in R_{i,j}} p^{(k)} a_{i,j}^{(k)} \quad (4)$$

$$= \sum_{r^{(k)} \in R_{i,j}} \frac{f^{(k)}}{\sum_{r^{(q)} \in R_{i,j}} f^{(q)}} \cdot a_{i,j}^{(k)} \quad (5)$$

$$= \sum_{r^{(k)} \in R_{i,j}} \frac{1/L^{(k)}}{\sum_{r^{(q)} \in R_{i,j}} 1/L^{(q)}} \cdot \frac{L^{(k)}}{\sum_{r^{(q)} \in R_{i,j}} L^{(q)}} |LR_{i,j}| \quad (6)$$

$$= \frac{|R_{i,j}| |LR_{i,j}|}{\sum_{r^{(q)} \in R_{i,j}} 1/L^{(q)} \cdot \sum_{r^{(q)} \in R_{i,j}} L^{(q)}}, \quad (7)$$

where  $p^{(k)}$  denotes the possibility that the current request is  $r^{(k)}$ .

From a quasi-static model, we assume that the number of link resources reaches equilibrium, and the consuming rate equals the producing rate (i.e.,  $a_{i,j}^{\text{alloc}} = \overline{a_{i,j}^{\text{prod}}}$ ). Thus, we can get a stable amount of link-layer entangled pairs from Eq. (3) and Eq. (7):

$$|LR_{i,j}| = \frac{2tv_{i,j}p_{i,j} \sum_{r^{(q)} \in R_{i,j}} L^{(q)}}{|R_{i,j}|}. \quad (8)$$

By substituting the result of Eq. (8) into Eq. (7), we obtain the expected number of entangled pairs allocated to request  $r^{(k)}$  per second  $R^{(k)}$  can be expressed as:

$$R^{(k)} = f^{(k)} \cdot a_{i,j}^{(k)} = \frac{1}{2L^{(k)}t} \frac{L^{(k)}}{\sum_{r^{(q)} \in R_{i,j}} L^{(q)}} \quad (9)$$

$$\cdot \frac{2tv_{i,j}p_{i,j} \sum_{r^{(q)} \in R_{i,j}} L^{(q)}}{|R_{i,j}|} \quad (10)$$

$$= \frac{v_{i,j}p_{i,j}}{|R_{i,j}|}. \quad (11)$$

This analysis proves that, on average, all requests are allocated the same number of link-layer entangled pairs per second, regardless of the path length. Therefore, it can be concluded that resources are allocated fairly and independently among all requests.

For a given request  $r^{(k)}$ , EAS allocates  $\alpha_{\text{fair}}^{(k)} |LR_{i,j}|$  link resources. To further reduce resource conflicts during the allocation of entangled pairs between two adjacent nodes, we add an extra rule. The rule requires that the two nodes apply the entangled pair from two directions with the resource's ID sort. Consider two requests, one has a sub-path from  $n_i$  to  $n_j$ , and the other has a sub-path from  $n_j$  to  $n_i$ . To avoid bi-directional conflicts, the extra rule is: if  $i < j$ ,  $n_i$  allocates resources in the forward order in  $LR_{i,j}$  (using #1, #2, and so on, entangled pairs), and  $n_j$  uses resources in the reverse order (#n, #n-1, and so on, entangled pairs). Note that here, we do not choose entangled pairs based on their fidelity, since it is still an open problem on how to select the entangled pairs. One may prefer to use entangled pairs with higher-fidelity, but this leads to the low-fidelity entangled pairs being wasted. Or, if we select low-fidelity entangled pairs, the distributed end-to-end entangled pairs' fidelity will also be influenced. In our REDP protocol, we intend to reduce the fidelity downgrading by optimizing their waiting time, which will be further discussed in Section IV-E and proved by the experiments.

Consequently, the EAS algorithm in REDP-Fair is shown in Algorithm 3. Furthermore, line 2 of Algorithm 3 demonstrates the extra rule to reduce conflicts.

#### D. Final Version of REDP: Further Efficiency Improvement

So far, while REDP-Fair guarantees the fairness property, it does not achieve optimal efficiency. An example is shown in Fig. 3.  $r^{(3)}$  is distributing entanglements between  $n_4$  and

**Algorithm 3** EAS Algorithm in REDP-Fair

---

**Input:** The current node  $n_i$ ; The request  $r^{(k)}$ ;  
The next-hop link  $l_{i,j}$ ; Resources on next-hop link  $LR_{i,j}$ ; Requests on next-hop link  $R_{i,j}$ ;  
**Output:** The applying entanglement resources set  $a_{i,j}^a$ ;

```

1  $w_i \leftarrow \frac{L^{(k)}}{\sum_{r^{(q)} \in R_{i,j}} L^{(q)}} |LR_{i,j}|$ ;
2 if  $i < j$  then
3    $a_{i,j}^a \leftarrow LR_{i,j}[w_i]$ ; // apply  $w_i$  resources
   in ascending order
4 else
5    $a_{i,j}^a \leftarrow LR_{i,j}[w_i]$ ; // apply  $w_i$  resources
   in descending order
6 end
7 return  $a_{i,j}^a$ ;

```

---

$n_6$ , while  $r^{(4)}$  is between  $n_4$  and  $n_7$ . Without loss of generality, we assume that all links generate the same number of link resources per second, denoted as  $C$ . Since REDP-Fair guarantees fairness,  $r^{(2)}$ ,  $r^{(3)}$ , and  $r^{(4)}$  applies  $C/3$  resources, as they share the same bottleneck link  $l_{4,6}$ . For  $r^{(1)}$  and  $r^{(2)}$ , they share the same link,  $l_{3,4}$ . Based on REDP-Fair, both  $r^{(1)}$  and  $r^{(2)}$  should use no more than  $C/2$  resources. However, since  $r^{(2)}$  has a bottleneck on  $l_{4,6}$  and cannot use more than  $C/3$  resources,  $r^{(1)}$  can actually use up to  $2C/3$  resources (larger than the fairness bound of  $C/2$ ). Consequently, the unused resources are  $2C/3 - C/2 = 1/6C$ . This example demonstrates that the network is not optimal, as repeaters have no information about other requests. In this case,  $r^{(1)}$  does not know the expected bandwidth of  $r^{(2)}$ .

Thus, we propose the final version of REDP to investigate and exploit the unused entangled pair resources. The basic idea is to evaluate the expected bandwidth for each request in each FBP round and use this global information for better decision-making. As a result, optimal throughput performance can be achieved while ensuring fairness. To this end, we design an algorithm called the Source Window Strategy (SWS), which operates on the source node of each request. The SWS calculates the expected bandwidth  $w_{\text{src}}^{(k)}$ , which represents the expected number of distributed end-to-end entangled pairs in the current FBP round. The expected bandwidth is also sent along with the forward packet. Thus, all repeaters on the path can receive and utilize this global information to make better resource allocation decisions to balance fairness and throughput.

To evaluate the expected bandwidth, we adopt a probe-based approach rather than a statistical one. The statistical approach, which involves collecting bandwidth data from previous rounds, is known to suffer from a long convergence time because the number of existing link resources varies significantly due to the qubit loss on links, as demonstrated in the evaluations later in the experiments in Sec. V.

Therefore, we adopt a probe-based approach that comprises two stages: **fair bandwidth evaluation** and **actual bandwidth probing**. The SWS enters the **fair bandwidth evaluation** stage when a new request begins to distribute end-to-end

entangled pairs or when the link status changes. The proof in the previous section shows that the resources are allocated fairly, but not all resources are used. Therefore, the evaluated bandwidth when resources are fairly assigned represents only the lower bound as it only guarantees fairness but does not fully utilize the network resources. Subsequently, in the next FBP round, the SWS enters the **actual bandwidth probing** stage. It updates the expected bandwidth based on the actual bandwidth in the last round. For instance, if the actual bandwidth reaches the evaluated bandwidth, it means that the network has enough resources, or there may be unused resources. Therefore, the expected bandwidth  $w_{\text{src}}^{(k)}$  can be increased in the next FBP round to probe unused resources. Eventually, we propose an Additive-Increase Reset-Fair (AIRF) algorithm to probe and leverage the free resources in the network that other requests cannot use.

We first present the **fair bandwidth evaluation** stage, which involves two significant improvements to the FBP protocol for evaluating the expected bandwidth. Firstly, during the backward phase, all repeaters report the network states  $S_{i,j} = (v_{i,j}, p_{i,j}, r_{i,j})$  to the source node, which includes  $v_{i,j}$ ,  $p_{i,j}$ , and the number of requests currently passing through the link  $r_{i,j} = |R_{i,j}|$ . Secondly, when the new forward phase begins, the source node calculates the evaluated fair bandwidth for the current round based on the link status. Let  $w_{\text{src}}^{(k)}$  be the source node's evaluated bandwidth, and  $w_{\text{fair}}^{(k)}$  be the bandwidth when resources are allocated fairly. Thus, We use  $w_{\text{fair}}^{(k)}$  as the lower bound and the initial value of the expected bandwidth  $w_{\text{src}}^{(k)}$ . The source node sends  $w_{\text{fair}}^{(k)}$  along with the forward packet. This process introduces no additional messages or delays compared to the original FBP protocol.

To calculate the fair bandwidth  $w_{\text{fair}}^{(k)}$ , we need to determine the distribution of available entangled pair resources for a given request  $r^{(k)}$  with path  $\mathcal{P}^{(k)} = \{l_{i,i+1} | i \leq L\}$  when resources are fairly assigned. Specifically, the source node computes the distribution of available resources fairly assigned to  $r^{(k)}$  for each link. Then, we use the first-order statistics of these distributions as the expected fair bandwidth of the whole path, corresponding to the number of distributions of the bottleneck link.

First, we model the number of entangled pairs to be fairly assigned to each link for the request  $r^{(k)}$ . The number of entangled pairs generated in  $T$  seconds follows a binomial distribution  $|LR_{i,i+1}| \sim B(v_{i,i+1}T, p_{i,i+1})$ . However, computing the binomial distribution can be computationally expensive when  $v_{i,i+1}T$  is large. Thus, we use a normal distribution to approximate the distribution [46], where  $|LR_{i,i+1}| \sim N(v_{i,i+1}Tp_{i,i+1}, v_{i,i+1}Tp_{i,i+1}(1-p_{i,i+1}))$ , and  $N(\cdot)$  is the normal distribution. Additionally, we consider that the limited number of quantum memories restricts the number of link resources. Therefore, we modify the distribution as  $Pr[|LR_{i,i+1}| = y] =$

$$\begin{cases} \Phi(0.5), & y = 0, \\ \Phi(y + 0.5) - \Phi(y - 0.5), & y \in [1, C_{i,i+1}), \\ 1 - \Phi(y - 0.5), & y = C_{i,i+1}, \end{cases} \quad (12)$$

where  $C_{i,i+1}$  represents the number of quantum memories on the link  $l_{i,i+1}$ , and  $\Phi$  represents the Cumulative Distribution Function (CDF) of  $N(v_{i,i+1}Tp_{i,i+1}, v_{i,i+1}Tp_{i,i+1}(1 - p_{i,i+1}))$ . We truncate the distribution when  $y < 0$  or  $y > C_{i,i+1}$ . Eq. (12) provides an approximation of the total number of entangled pairs on the link  $l_{i,i+1}$ . We then calculate the number of entangled pairs that can be fairly assigned to  $r^{(k)}$ , which follows the constraints  $Pr[w_{i,i+1}^{(k)} = y] = Pr[LR_{i,i+1} = |R_{i,j}| \cdot y]$ . We denote the new distribution of the fair bandwidth of one request as  $\mathbf{W}_{(k)}^i$ .

Finally, we can predict the expected path-level fair bandwidth, which is equivalent to the bottleneck link's resource. The path-level fair bandwidth equals the first-order statistic of all links  $\mathbf{W}_{(k)}^i$ :

$$w_{\text{fair}}^{(k)} = E[\mathbf{W}_{(k)}^{(0)}] = \sum_{y=0}^{C_{\min}^{(k)}} y \sum_{i=1}^L Pr[\mathbf{W}_{(k)}^i = y] \quad (13)$$

$$\times \prod_{j=1, j \neq i}^L Pr[\mathbf{W}_{(k)}^j \geq y]. \quad (14)$$

The computational overhead is  $\mathcal{O}(C_{\min}^{(k)} \cdot L^{(k)} + C_{\min}^{(k)} \cdot L^{(k)^2})$ , where  $C_{\min}^{(k)} = \min\{C_{i,i+1} | 1 \leq i \leq L\}$  represents the size of the smallest memory on the links. The first part involves pre-computing  $W_{i,i+1}^{(k)}$  for each link, and the second part comes from calculating Eq. (14).

We now discuss the **actual bandwidth probing** phase, which describes how the expected bandwidth is updated in the following FBP rounds. As previously mentioned, we evaluate  $w_{\text{fair}}^{(k)}$  as the lower bound and initial value of  $w_{\text{src}}^{(k)}$ . Then, in **actual bandwidth probing**, we use AIRF to probe the unused network bandwidth. Unlike TCP and Tele-DTP [30], we do not adopt a Slow Start period because  $w_{\text{fair}}^{(k)}$  can already be calculated as bandwidth lower bound when all resources are allocated fairly. In the following rounds, if the actual bandwidth  $w^{(k)} \geq \beta w_{\text{src}}^{(k)}$ , there is no network resource congestion, and there may be free resources that can be used. Consequently, the expected bandwidth increases by 1 in the next round. Otherwise, if  $w^{(k)} < \beta w_{\text{src}}^{(k)}$  or the network state changes (e.g., when a new request arrives), we recalculate  $w_{\text{fair}}^{(k)}$  and reset the expected bandwidth  $w_{\text{src}}^{(k)}$  to it. The SWS algorithm is shown in Algorithm 4.

We enhance the EAS algorithm by incorporating the expected bandwidth evaluated from SWS to improve the efficiency property. The modified EAS algorithm has two factors. The first factor is the fairness factor  $\alpha_{\text{fair}}^{(k)}$ , introduced in Eq. (2) to ensure fairness property. The second factor is the efficiency factor  $\alpha_{\text{eff}}^{(k)}$ , which is in proportion to  $w_{\text{src}}^{(k)}$  for all requests:

$$\alpha_{\text{eff}}^{(k)} = \frac{w_{\text{src}}^{(k)}}{\sum_{r^{(q)} \in R_{i,j}} w_{\text{src}}^{(q)}}. \quad (15)$$

Since the source node sends  $w_{\text{src}}^{(k)}$  in the forward packet, the repeaters can calculate  $\alpha_{\text{eff}}^{(k)}$ . With the two factors, repeaters can make a better decision to be both fair and efficient by using the larger value of  $\alpha_{\text{eff}}^{(k)}$  and  $\alpha_{\text{fair}}^{(k)}$  as the enabling factor. If  $\alpha_{\text{fair}}^{(k)}$

---

#### Algorithm 4 AIRF in Source Window Strategy (SWS)

---

**Input:** The current request  $r^{(k)}$ ; The actual bandwidth in the last round  $w^{(k)}$ ; Network states collected from backward packet  $\{(v_{i,i+1}, p_{i,i+1}, r_{i,i+1}) | 1 \leq i \leq L\}$ ;  
**Output:** The expected bandwidth  $w_{\text{src}}^{(k)}$ ;

- 1 **if**  $w^{(k)} \geq \beta w_{\text{src}}^{(k)}$  *and network states does not change* **then**
- // actual bandwidth probing
- $w_{\text{src}}^{(k)} \leftarrow w_{\text{src}}^{(k)} + 1$ ;
- 3 **else**
- // fair bandwidth evaluation
- 4 **for**  $i \in [1, L^{(k)}]$  **do**
- calculate the distribution of usable link resources on link  $l_{i,i+1}$ ,  $W_{(k)}^i$ ;
- 6 **end**
- 7  $w_{\text{fair}}^{(k)} \leftarrow E[\mathbf{W}_{(k)}^{(0)}]$ , calculated from Eq. (14) ;
- 8  $w_{\text{src}}^{(k)} \leftarrow w_{\text{fair}}^{(k)}$ ;
- 9 **end**
- 10 **return**  $w_{\text{src}}^{(k)}$ ;

---



---

#### Algorithm 5 Entanglement Allocation Strategy (EAS)

---

**Input:** The current node  $n_i$ ; The request  $r^{(k)}$ ;  
The next-hop link  $l_{i,j}$ ; Resources on next-hop link  $LR_{i,j}$ ; Requests on next-hop link  $R_{i,j}$ ;  
**Output:** Applying resources set  $a_{i,j}^a$  on the link  $l_{i,j}$ ;

- 1  $\alpha_{\text{fair}}^{(k)} = \frac{L^{(k)}}{\sum_{r^{(q)} \in R_{i,j}} L^{(k)}};$
- 2  $\alpha_{\text{eff}}^{(k)} = \frac{w_{\text{src}}^{(k)}}{\sum_{r^{(q)} \in R_{i,j}} w_{\text{src}}^{(q)}};$
- 3  $\alpha^{(k)} \leftarrow \max\{\alpha_{\text{fair}}^{(k)}, \alpha_{\text{eff}}^{(k)}\};$
- 4  $w_i \leftarrow \alpha^{(k)} \cdot |LR_{i,j}|;$
- 5 **if**  $i > j$  **then**
- $a_{i,j}^a \leftarrow LR_{i,j}[:w_i];$
- 7 **else**
- $a_{i,j}^a \leftarrow LR_{i,j}[-w_i:];$
- 9 **end**
- 10 **return**  $a_{i,j}^a$ ;

---

is larger, it implies that at least  $\alpha_{\text{fair}}^{(k)} \cdot |LR_{i,j}|$  resources must be reserved for one request to ensure fairness. On the other hand, if  $\alpha_{\text{eff}}^{(k)}$  is larger, it suggests that some resources may remain unused, and the request can utilize up to  $\alpha_{\text{eff}}^{(k)} \cdot |LR_{i,j}|$  resources to fully utilize the free link resources. The final version of the EAS algorithm is presented in Algorithm 5.

In summary, REDP utilizes a probing-based approach to estimate the available bandwidth for each request in the SWS. This expected bandwidth is a useful metric to identify potential unused resources. The EAS algorithm then adapts the fairness factor discussed in Sec. IV-C and the evaluated bandwidth in the SWS to allocate appropriate link resources. It ensures both network-layer fairness and efficiency.

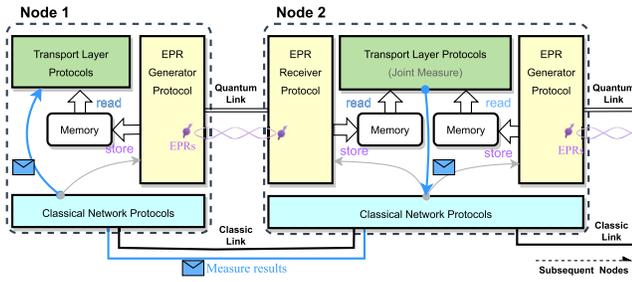


Fig. 4. Modularized double protocol stacks on quantum nodes in simulation.

### E. Discussion

In this section, we discuss the efficiency of the proposed REDP protocol and compare it to centralized schemes.

In the proposed REDP protocol, it takes about  $2t \cdot L$  time to finish one FBP for a  $L$ -hop path. It could be slow in a large network. However, we show that this delay is optimized and necessary whenever we consider the communication delay. For centralized schemes [17], [18], [19], [20], they usually run in periods. In each period, the controller needs to collect information from all nodes in the network. Thus, it takes about  $2t \cdot L_{\max}$  time to collect information and deliver decisions, where  $L_{\max}$  is the distance between the controller and the farthest node. It will usually take more time for such a period in centralized schemes compared to one FBP in our REDP protocol, especially for a larger network.

Besides, one may concern that the REDP protocol operates BSMs hop-by-hop, while repeaters in other schemes can operate BSMs simultaneously, which may lead to a higher performance. However, as we show in Fig. 1, the repeaters still need to send the measurement results to the source node, and the time for BSMs (typically within microseconds [47]) can be ignored when we introduce the communication signal delay. To prove it, we will adopt a concurrent BSMs manner for our baseline schemes in the experiments. Overall, since we reduce the decision-making and resource preservation time, our method should achieve higher throughput and overall fidelity as the entangled pairs spend less time waiting in the quantum memory.

## V. PERFORMANCE EVALUATIONS

### A. Platform and Baselines

We use the network-layer quantum network simulator, named SimQN [22], to evaluate the performance of REDP. It is a modularized simulator useful in large-scale and complex protocol simulation. As shown in Fig. 4, we implement the following three modules on each quantum node in the simulation: 1) we leverage the classic network protocol stacks provided by SimQN to simulate the classic control message communications. 2) we design an EPR Generator/Receiver protocol to continuously distribute link-layer entangled pairs on the links. 3) we implement the remote entanglement distribution protocols, including REDP, two baselines, and two other competitor approaches. In conclusion, we perform full-stack systematic simulations in quantum networks.

We evaluate the REDP-FIFO, REDP-Fair, and the full version of REDP in the simulation. To better show the

trade-offs between centralized methods and decentralized methods (including our proposed REDP protocol), we implement both centralized PS and PU in [19] and a modified version of the decentralized protocol named Tele-DTP [30]. PS/PU algorithms are centralized approaches that require a network controller. To better show the different behaviors of the centralized and decentralized approaches, we design a full-stack corresponding communication protocol for PS/PU. In brief, 1) the controller first collects network states, produces and delivers entanglement distribution instructions. 2) Then, repeaters perform concurrent BSMs in parallel once they receive the controller's instructions and send results to the source node. 3) After remote entanglement distribution, source nodes will report the result to the controller. 4) Finally, if the controller receives the results of all requests, it starts a new round. Tele-DTP, on the other hand, is a decentralized protocol in [30]. The original protocol used a memory allocation strategy to decide the resource allocation. However, it assigns memories and generates entangled pairs on demand after each round, which may bring a longer latency. We modify the original protocol to generate resources continuously but adopt its two core resource allocation algorithms.

Finally, we conduct solid experiments and perform our evaluations in a single-request linear full-homogeneous path topology (Sec. V-B), a multiple-requests dumbbell topology (Sec. V-C) and a large-scale random topology (Sec. V-D).

### B. Evaluations in a Full Homogeneous Path

In the first experiment, we construct a linear full homogeneous path with varying path lengths to evaluate the throughput and fidelity of our approach under different path lengths. The quantum links generate entangled pairs at a constant speed of  $v_{i,j} = 1000Hz$ , which is achievable in [48]. We set  $p_{\text{init}} = 0$ ,  $\mu = 0.2$  db/km and  $D_{i,j} = 100$  km, similar to the settings used in [49]. The propagation delay for both the classical links is set to be a normal distribution  $t \sim N(0.010, 0.004)$  ms to simulate the classical communication delay and potential congestion. We also set the quantum memory size on each link to 100, as in [19]. Qubits suffer from a depolarizing noise [50] in a quantum memory, and the decay rate  $\gamma = 1s$  [51]. Additionally, we introduce dephasing noise during quantum operations and measurements, where an additional Z gate is operated on the target qubit with a probability  $p_d = 0.001$  [28].

This experiment compares the throughput (*i.e.*, entanglement distribution rate) and fidelity between REDP, Tele-DTP, and PS/PU algorithms. The path hop is from 2 to 25, and we run each simulation for about 10 seconds (at the 10 seconds, we send a FIN command to close the request but wait for the last round to finish). For REDP, we set  $\beta = 0.7$ . The result of the performance under different hops is shown in TABLE I. Here, ebit denotes the number of distributed remote entangled pairs. REDP achieves the best throughput performance compared to PS/PU and Tele-DTP. All algorithms suffer from the least throughput downgrade when the path goes longer. We observe that both PS and PU behaviors are the same in this single-path scenario, but the throughput is not

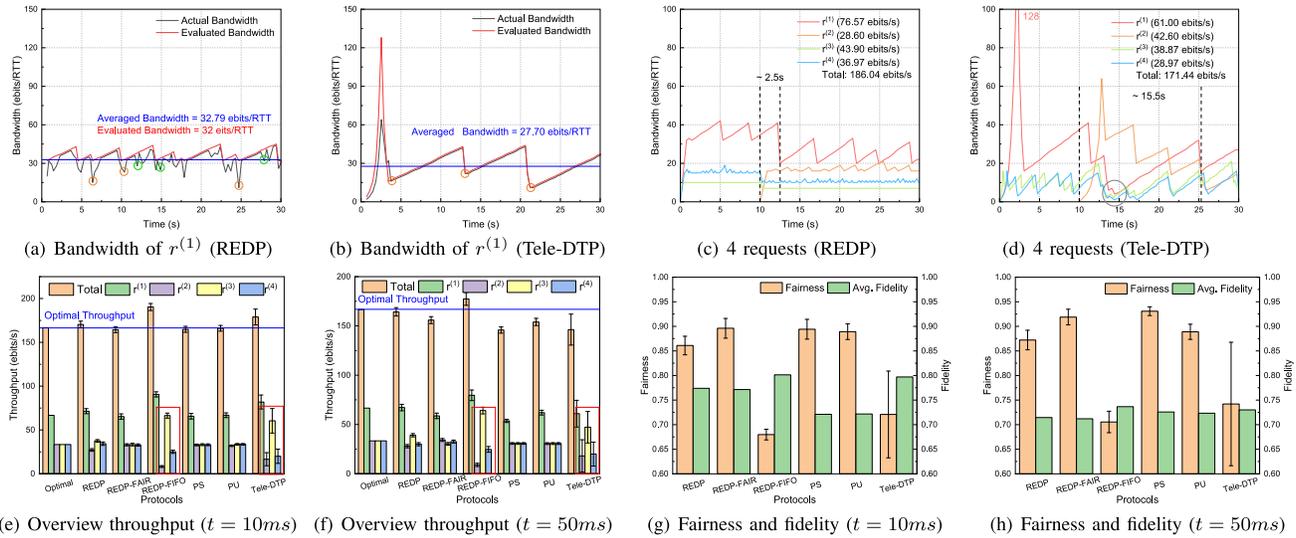


Fig. 5. Performance comparison in terms of bandwidth, throughput, and fairness in a dumbbell topology.

TABLE I

PERFORMANCE COMPARISON IN A LINEAR FULL HOMOGENEOUS PATH

#	Throughput (ebits/s) *			Averaged Fidelity		
	REDP	PS/PU	Tele-DTP	REDP	PS/PU	Tele-DTP
2	97.14 (-0.04%)	97.64 (+0.47%)	97.18	0.9613	0.9338	0.9624
5	95.36 (+4.84%)	94.06 (+3.41%)	90.96	0.7204	0.7047	0.7462
10	94.18 (+11.9%)	94.40 (+12.1%)	84.18	0.6565	0.6565	0.6459
15	92.84 (+16.6%)	91.70 (+15.1%)	79.64	0.6308	0.6309	0.6283
20	91.72 (+15.7%)	89.46 (+12.8%)	79.30	0.6091	0.6091	0.6090
25	88.06 (+40.7%)	78.06 (+24.7%)	62.60	0.5909	0.5909	0.5909

# denotes the number of nodes on the path.

\* The values in ( ) are relative ratios compared to Tele-DTP.

as good as REDP because PS/PU is a centralized algorithm. They need more time overhead to communicate when the path goes longer, and the time to finish one round is larger than that in REDP. Entanglement allocation-based Tele-DTP, on the other hand, uses a statistically-based way to decide the sending window. It is sensitive to the probabilistic fluctuations in the number of link-layer entangled pairs. The sending window is significantly reduced whenever one link generates fewer entangled pairs than expected, causing a low throughput in the following rounds.

In TABLE I, we analyze the average fidelity of all distributed remote entangled pairs. The results indicate that all algorithms achieve a roughly equivalent level of fidelity. However, the fidelity drops significantly as the path length increases. REDP and Tele-QTP achieve a higher fidelity when the path length is smaller than 5, but the difference decreases as the path length grows.

### C. Evaluations in an Asymmetric Dumbbell Topology

To further examine the performance in multiple request scenarios, we use the asymmetric dumbbell topology shown in Fig. 3. Here, there are four requests that run concurrently, and they have different path lengths. This experiment aims to demonstrate that the evaluation results align with the theoretical models. First, it shows whether REDP can achieve fairness for multiple requests with different lengths. Meanwhile, it also indicates whether the window evaluation fits the theoretical models.

In the beginning, we evaluate the theoretical bandwidth and the actual bandwidth in a single request scenario (i.e., only  $r^{(1)}$  in Fig. 3). We compare REDP and Tele-DTP since they both have a bandwidth control mechanism. The link delay is set to  $t \sim N(0.05, 0.02)$  ms to demonstrate the window variation better. The results from REDP and Tele-DTP are depicted in Fig. 5(a) and Fig. 5(b), respectively. We observe that both REDP and Tele-DTP experience bandwidth fluctuations due to probabilistic link entanglement generation failure, indicated by orange circles. However, REDP achieves a superior bandwidth of 32.79 ebits (number of distributed entangled pairs) per round compared to 27.70 ebits per round in Tele-DTP. This advantage can be attributed to three aspects. Firstly, REDP does not require a Slow Start and initiates transmission at the evaluated fair bandwidth  $w_{\text{fair}}^{(1)}$ . Secondly, REDP can ignore small fluctuations of link resource generation, as demonstrated by the green circles in Fig. 5(a). Thirdly, instead of reducing the bandwidth by half, REDP resets the window to the evaluated fair bandwidth  $w_{\text{fair}}^{(1)}$  to avoid severe degradation. Additionally, we observe that the actual bandwidth achieved by REDP (32.79 ebits per round) aligns well with the theoretical value of  $w_{\text{fair}}^{(1)}$  (32 ebits per round).

Furthermore, to show the performance in a dynamic network, we run all four requests and observe the bandwidth when a new request comes. Specifically, request  $r^{(2)}$  initiates at 10 seconds. Fig. 5(c) and Fig. 5(d) show the bandwidth of all four requests in REDP and Tele-DTP. REDP can generally adapt well to the link status changes and bring fewer bandwidth fluctuations than Tele-DTP. Besides, when  $r^{(2)}$  enters the network, the bandwidth stabilizes in approximately 2.5 seconds, whereas Tele-DTP exhibits more significant variations, particularly when  $r^{(2)}$  is in a Slow Start phase (black circle). It takes approximately 15.5 seconds for the network to reach a new stable bandwidth. The overall throughput of REDP is also superior to that of Tele-DTP.

We then evaluate the throughput, fairness, and fidelity of all requests. We use Jain's fairness index [52] to evaluate the fairness and simulate each algorithm 100 times. For the PS/PU algorithm, we select  $n_3$  as the controller since it is

the center of the network. Fig. 5(e) and Fig. 5(f) show the throughput when the average delay  $t$  is 10 ms and 50 ms, respectively. In Fig. 5(e), we find that REDP-FIFO and Tele-DTP do not guarantee strict fairness, as the throughput of  $r^{(2)}$ ,  $r^{(3)}$ , and  $r^{(4)}$  are not roughly equal (in the red square). Other algorithms, including REDP, REDP-Fair, PS, and PU, provide equal throughput for these three requests. Moreover, when  $t = 10ms$ , all algorithms reach the optimal total throughput (the blue line). In Fig. 5(f), REDP provides the best throughput compared to other approaches that guarantee fairness. REDP outperforms REDP-Fair because it considers global information and has better resource utilization. Additionally, it outperforms PS and PU as the centralized controller brings additional classical communication delay.

We examine the network fairness and fidelity for all requests in Fig. 5(g) and Fig. 5(h) when the expected delay is  $t = 10ms$  and  $t = 50ms$ , respectively. In Fig. 5(g), we observe that PS and PU provide the best fairness since they are centralized. Moreover, REDP-Fair provides similar fairness to PS/PU even though it is a decentralized protocol without global information. REDP also provides fairness but is not as good as REDP-Fair, as it prefers utilizing free resources to achieve better throughput. REDP-FIFO has worse fairness. In Fig. 5(h), a similar observation is that REDP can achieve good fairness. As for the fidelity, we find that PS/PU achieves less average fidelity because the centralized control takes more time to complete a round than REDP, and entangled pairs suffer more noise in quantum memories.

#### D. Evaluations in a Large-Scale Topology

This experiment aims to evaluate the performance in a large-scale random network topology. We construct the network topology based on the Waxman algorithm [53], which consists of 50 nodes. We randomly pick 10 requests to evaluate the overall network performance. Notably, among all requests, 9 requests can successfully find a connected path. However, request  $r^{(3)}$  is invalid, as no path connects the source node and the destination node. The average path length of 9 valid requests is 5.9 hops. Specifically, for PS/PU, we designate  $n_1$  as the controller, considering the random topology. In the experiment, we set the propagation delay to  $t \sim N(0.01, 0.004)$  ms and run each approach 100 times.

In Fig. 6(a), the orange bar represents the total throughput. Notably, REDP exhibits the most outstanding throughput among all the competitors. REDP's throughput is 13.9% better than that of the PU algorithm and 39.1% higher than that of Tele-DTP. In terms of fairness, the PS/PU algorithm performs the best with a Jain's index of approximately 0.83. REDP and REDP-Fair have slightly worse fairness, given that REDP is a decentralized protocol without global information. Of the three versions of REDP, REDP-Fair performs the best in fairness. Interestingly, the average fidelity of all algorithms does not show a significant difference. The fidelity of PS/PU (0.67) is slightly worse than decentralized algorithms (0.70).

In Fig. 6(b), we present the Cumulative Distribution Function (CDF) of the throughput for all the approaches. Among them, REDP achieves the highest throughput, as drawn in the

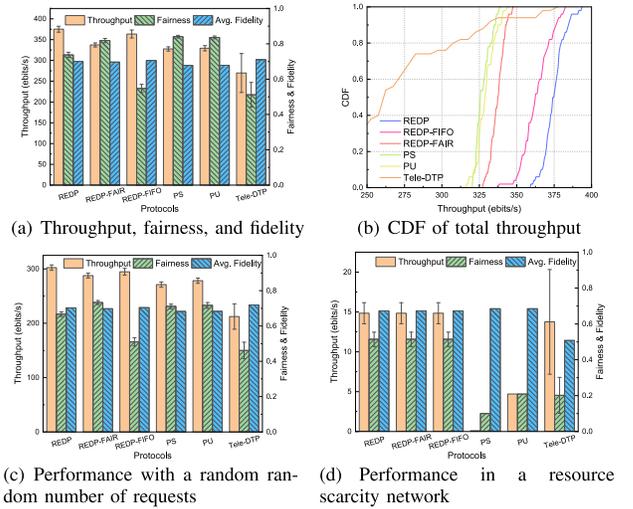


Fig. 6. Performance comparison in a large-scale random topology.

purple line, followed by REDP-FIFO, REDP-Fair, PU, and PS. Entanglement allocation-based Tele-DTP has more throughput fluctuations, indicating that it suffers from probabilistic qubit loss on links.

To evaluate the performance in a more dynamic network, we conduct the experiments that request start (and finish) at the first (and the last) 30% of the simulation time. It illustrates the overall performance when the network serves a different number of requests. The results in Fig. 6(c) show that REDP still achieves the best throughput and good fairness compared to both the centralized and distributed schemes, which is similar to Fig. 6(a). It indicates that the proposed REDP protocol can fit dynamic changes in the network.

Finally, we set the quantum memory size  $C = 1$  to demonstrate the performance when the network is in an extreme resource scarcity situation, as shown in Fig. 6(d). We observe that all three versions of the REDP protocol behave the same, as the previous request will consume all the resources. REDP still has performance superior compared to other baselines in throughput and fairness. We notice that PS does not work well because it only adopts a link-layer resource allocation.

## VI. CONCLUSION

In this paper, we designed a reliable entanglement distribution protocol, REDP, to address the crucial entanglement distribution procedure in quantum networks. The major challenge is to reach a precise path consensus for distributing entangled pairs and determining resource allocation, even in decentralized multiple requests large-scale quantum networks. We proposed the FBP protocol to reach a consistent consensus among all nodes on a path to control the entanglement distribution procedure with the slightest time delay, thus reducing the fidelity downgrade. Furthermore, we presented EAS and SAS algorithms based on probabilistic modeling to ensure network fairness and efficiency properties. We use the quantum simulation platform to conduct adequate, extensive, and the most realistic full-stack simulations in quantum networks. Compared to the latest work, our results demonstrate the outstanding performance of REDP in terms of throughput, fairness, and fidelity, which meets our design goals. Our work

also highlights the critical impact of communication overhead in both classic and quantum transmission and demonstrates the feasibility of building a large-scale decentralized quantum network.

## REFERENCES

- [1] A. S. Cacciapuoti, J. Illiano, S. Koudia, K. Simonov, and M. Caleffi, "The quantum Internet: Enhancing classical Internet services one qubit at a time," *IEEE Netw.*, vol. 36, no. 5, pp. 6–12, Sep. 2022.
- [2] S. Wehner, D. Elkouss, and R. Hanson, "Quantum Internet: A vision for the road ahead," *Science*, vol. 362, no. 6412, Oct. 2018.
- [3] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," *Theor. Comput. Sci.*, vol. 560, pp. 7–11, Dec. 2014.
- [4] W.-Y. Hwang, "Quantum key distribution with high loss: Toward global secure communication," *Phys. Rev. Lett.*, vol. 91, no. 5, Aug. 2003, Art. no. 057901.
- [5] D. Cuomo, M. Caleffi, and A. S. Cacciapuoti, "Towards a distributed quantum computing ecosystem," *IET Quantum Commun.*, vol. 1, no. 1, pp. 3–8, Jul. 2020.
- [6] E. Rieffel and W. Polak, "An introduction to quantum computing for non-physicists," *ACM Comput. Surv.*, vol. 32, no. 3, pp. 300–335, Sep. 2000.
- [7] P. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annu. Symp. Found. Comput. Sci. (FOCS)*, 1994, pp. 124–134.
- [8] Z. Eldredge, M. Foss-Feig, J. A. Gross, S. L. Rolston, and A. V. Gorshkov, "Optimal and secure measurement protocols for quantum sensor networks," *Phys. Rev. A*, vol. 97, no. 4, Apr. 2018, Art. no. 042337.
- [9] E. O. Ilo-Okeke, L. Tessler, J. P. Dowling, and T. Byrnes, "Remote quantum clock synchronization without synchronized clocks," *NPJ Quantum Inf.*, vol. 4, no. 1, p. 40, Aug. 2018.
- [10] J. Illiano, M. Caleffi, A. Manzalini, and A. S. Cacciapuoti, "Quantum Internet protocol stack: A comprehensive survey," *Comput. Netw.*, vol. 213, Aug. 2022, Art. no. 109092.
- [11] I. Marcikic, H. de Riedmatten, W. Tittel, H. Zbinden, M. Legré, and N. Gisin, "Distribution of time-bin entangled qubits over 50 km of optical fiber," *Phys. Rev. Lett.*, vol. 93, no. 18, Oct. 2004, Art. no. 180502.
- [12] J. Yin et al., "Satellite-to-ground entanglement-based quantum key distribution," *Phys. Rev. Lett.*, vol. 119, Nov. 2017, Art. no. 200501.
- [13] L. Gyongyosi, S. Imre, and H. V. Nguyen, "A survey on quantum channel capacities," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 1149–1205, 2nd Quart., 2018.
- [14] H.-J. Briegel, W. Dür, J. I. Cirac, and P. Zoller, "Quantum repeaters: The role of imperfect local operations in quantum communication," *Phys. Rev. Lett.*, vol. 81, p. 5932, Dec. 1998.
- [15] L.-M. Duan, M. D. Lukin, J. I. Cirac, and P. Zoller, "Long-distance quantum communication with atomic ensembles and linear optics," *Nature*, vol. 414, no. 6862, pp. 413–418, 2001.
- [16] Z. Li et al., "Building a large-scale and wide-area quantum Internet based on an OSI-alike model," *China Commun.*, vol. 18, no. 10, pp. 1–14, Oct. 2021.
- [17] S. Shi and C. Qian, "Concurrent entanglement routing for quantum networks: Model and designs," in *Proc. 2020 ACM Special Interest Group Data Commun. (SIGCOMM)*. New York, NY, USA: ACM, 2020, pp. 62–75.
- [18] S. Zhang, S. Shi, C. Qian, and K. L. Yeung, "Fragmentation-aware entanglement routing for quantum networks," *J. Lightw. Technol.*, vol. 39, no. 14, pp. 4584–4591, Jul. 15, 2021.
- [19] C. Li, T. Li, Y.-X. Liu, and P. Cappellaro, "Effective routing design for remote entanglement generation on quantum networks," *NPJ Quantum Inf.*, vol. 7, no. 1, p. 10, Jan. 2021.
- [20] Y. Zhao and C. Qiao, "Redundant entanglement provisioning and selection for throughput maximization in quantum networks," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2021, pp. 1–10.
- [21] L. Chen et al., "A heuristic remote entanglement distribution algorithm on memory-limited quantum paths," *IEEE Trans. Commun.*, vol. 70, no. 11, pp. 7491–7504, Nov. 2022.
- [22] L. Chen et al., "SimQN: A network-layer simulator for the quantum network investigation," *IEEE Netw.*, vol. 37, no. 5, pp. 182–189, 2023, doi: 10.1109/MNET.130.2200481.
- [23] L. Gyongyosi and S. Imre, "Advances in the quantum Internet," *Commun. ACM*, vol. 65, no. 8, pp. 52–63, 2022.
- [24] M. Mehic et al., "Quantum key distribution: A networking perspective," *ACM Comput. Surv.*, vol. 53, no. 5, pp. 1–41, 2020.
- [25] 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.
- [26] S. Liu, Y. Lou, Y. Chen, and J. Jing, "All-optical entanglement swapping," *Phys. Rev. Lett.*, vol. 128, no. 6, Feb. 2022, Art. no. 060503.
- [27] D. Ribezzo et al., "Deploying an inter-European quantum network," *Adv. Quantum Technol.*, vol. 6, no. 2, Feb. 2023, Art. no. 2200061.
- [28] M. Ghaderibaneh, C. Zhan, H. Gupta, and C. R. Ramakrishnan, "Efficient quantum network communication using optimized entanglement swapping trees," *IEEE Trans. Quantum Eng.*, vol. 3, pp. 1–20, 2022.
- [29] C. Di Franco and D. Ballester, "Optimal path for a quantum teleportation protocol in entangled networks," *Phys. Rev. A, Gen. Phys.*, vol. 85, no. 1, Jan. 2012, Art. no. 010303.
- [30] Y. Zhao and C. Qiao, "Distributed transport protocols for quantum data networks," *IEEE/ACM Trans. Netw.*, vol. 31, no. 6, pp. 2777–2792, Dec. 2023, doi: 10.1109/TNET.2023.3262547.
- [31] W. Dai, T. Peng, and M. Z. Win, "Optimal remote entanglement distribution," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 3, pp. 540–556, Mar. 2020.
- [32] G. Vardoyan, S. Guha, P. Nain, and D. Towsley, "On the stochastic analysis of a quantum entanglement distribution switch," *IEEE Trans. Quantum Eng.*, vol. 2, pp. 1–16, 2021.
- [33] D. Bertsekas and R. Gallager, *Data Networks*. Nashua, NH, USA: Athena Scientific, 2021.
- [34] J. Li et al., "Fidelity-guaranteed entanglement routing in quantum networks," *IEEE Trans. Commun.*, vol. 70, no. 10, pp. 6748–6763, Oct. 2022.
- [35] N. Yu, C.-Y. Lai, and L. Zhou, "Protocols for packet quantum network intercommunication," *IEEE Trans. Quantum Eng.*, vol. 2, pp. 1–9, 2021.
- [36] W. Kozłowski, A. Dahlberg, and S. Wehner, "Designing a quantum network protocol," in *Proc. Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*. New York, NY, USA: ACM, 2020, pp. 1–16.
- [37] J. Moy, *OSPF Version 2*, document RFC 2328, 1998. Accessed: Nov. 7, 2023. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc9355>
- [38] L. Gyongyosi and S. Imre, "Adaptive routing for quantum memory failures in the quantum Internet," *Quantum Inf. Process.*, vol. 18, no. 2, pp. 1–21, Feb. 2019.
- [39] J. Y. Yen, "An algorithm for finding shortest routes from all source nodes to a given destination in general networks," *Quart. Appl. Math.*, vol. 27, no. 4, pp. 526–530, 1970.
- [40] A. Dahlberg et al., "A link layer protocol for quantum networks," in *Proc. ACM Special Interest Group Data Commun. (SIGCOMM)*. New York, NY, USA: ACM, 2019, pp. 159–173.
- [41] C. Couteau, "Spontaneous parametric down-conversion," *Contemp. Phys.*, vol. 59, no. 3, pp. 291–304, Jul. 2018.
- [42] 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," *Phys. Rev. Lett.*, vol. 76, no. 5, pp. 722–725, Jan. 1996.
- [43] Y. Zhao, G. Zhao, and C. Qiao, "E2E fidelity aware routing and purification for throughput maximization in quantum networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2022, pp. 480–489.
- [44] W. Wei, K. Xue, J. Han, Y. Xing, D. S. L. Wei, and P. Hong, "BBR-based congestion control and packet scheduling for bottleneck fairness considered multipath TCP in heterogeneous wireless networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 914–927, Jan. 2021.
- [45] J. Jaffe, "Bottleneck flow control," *IEEE Trans. Commun.*, vol. COM-29, no. 7, pp. 954–962, Jul. 1981.
- [46] G. E. Box, W. H. Hunter, and S. Hunter, *Statistics for Experimenters*, vol. 664. Reading, MA, USA: Addison-Wesley, 1978.
- [47] J. Nemirovsky and Y. Sagi, "Fast universal two-qubit gate for neutral fermionic atoms in optical tweezers," *Phys. Rev. Res.*, vol. 3, no. 1, Feb. 2021, Art. no. 013113.
- [48] R. Wang et al., "A dynamic multi-protocol entanglement distribution quantum network," in *Proc. IEEE Opt. Fiber Commun. Conf. Exhibit. (OFC)*, Mar. 2022, pp. 1–3.
- [49] C. Cicconetti, M. Conti, and A. Passarella, "Request scheduling in quantum networks," *IEEE Trans. Quantum Eng.*, vol. 2, pp. 2–17, 2021.
- [50] A. Basit, F. Badshah, H. Ali, and G.-Q. Ge, "Protecting quantum coherence and discord from decoherence of depolarizing noise via weak measurement and measurement reversal," *EPL Europhys. Letters*, vol. 118, no. 3, p. 30002, May 2017.
- [51] P. van Loock et al., "Extending quantum links: Modules for fiber- and memory-based quantum repeaters," *Adv. Quantum Technol.*, vol. 3, no. 11, Nov. 2020, Art. no. 1900141.

- [52] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe, "A quantitative measure of fairness and discrimination," *Eastern Res. Lab., Digit. Equip. Corp., Hudson, MA, USA, Tech. Rep.*, 1984, vol. 21, p. 1.
- [53] B. M. Waxman, "Routing of multipoint connections," *IEEE J. Sel. Areas Commun.*, vol. SAC-6, no. 9, pp. 1617–1622, Dec. 1988.

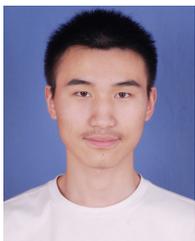


**Lutong Chen** (Graduate Student Member, IEEE) received the bachelor's degree from the School of Cyber Science and Technology, University of Science and Technology of China, in 2020. He is currently pursuing the Ph.D. degree with the School of Cyber Science and Technology. His research interests include quantum networking and network security.



**Kaiping Xue** (Senior Member, IEEE) received the bachelor's degree from the Department of Information Security, University of Science and Technology of China (USTC), in 2003, and the Ph.D. degree from the Department of Electronic Engineering and Information Science (EEIS), USTC, in 2007. From May 2012 to May 2013, he was a Post-Doctoral Researcher with the Department of Electrical and Computer Engineering, University of Florida. Currently, he is a Professor with the School of Cyber Science and Technology, USTC. He is also the

Director of the Network and Information Center, USTC. His research interests include next-generation internet architecture design, transmission optimization, and network security. He is an IET Fellow. His work won the Best Paper Awards from IEEE MSN 2017 and IEEE HotCN 2019, the Best Paper Honorable Mention from ACM CCS 2022, the Best Paper Runner-Up Award from IEEE MASS 2018, and the Best Track Paper from MSN 2020. He serves on the Editorial Board for several journals, including IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING (TDSC), IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS (TWC), and IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT (TNSM). He has also served as the (Lead) Guest Editor for many reputed journals/magazines, including IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC), *IEEE Communications Magazine*, and IEEE NETWORK.



**Jian Li** (Senior Member, IEEE) received the bachelor's degree from the Department of Electronics and Information Engineering, Anhui University, in 2015, and the Ph.D. degree from the Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC), in 2020. From November 2019 to November 2020, he was a Visiting Scholar with the Department of Electronic and Computer Engineering, University of Florida. From December 2020 to December 2022, he was a Post-Doctoral Researcher

with the School of Cyber Science and Technology, USTC. He is currently an Associate Researcher with the School of Cyber Science and Technology, USTC. His research interests include quantum networking, wireless networks, next-generation internet, and network security. He serves as an Editor for *China Communications*.



**Zhonghui Li** (Member, IEEE) received the bachelor's degree from the School of Information and Software Engineering, University of Electronic Science and Technology of China, in 2018, and the Ph.D. degree in information security from the School of Cyber Science and Technology, University of Science and Technology of China (USTC), in 2023. He is currently a Post-Doctoral Researcher with the School of Cyber Science and Technology, USTC. His current research interests include quantum networking and network security.



**Ruidong Li** (Senior Member, IEEE) received the bachelor's degree in engineering from Zhejiang University, China, in 2001, and the Doctor of Engineering degree from the University of Tsukuba in 2008. He is currently an Associate Professor with the College of Science and Engineering, Kanazawa University, Japan. Before joining Kanazawa University, he was a Senior Researcher with the Network System Research Institute, National Institute of Information and Communications Technology (NICT). His current research interests include future networks, big data networking, blockchain, information-centric networks, the Internet of Things, network security, wireless networks, and quantum internet. He is a member of IEICE. He is the Founder and the Chair of the IEEE SIG on big data intelligent networking and the IEEE SIG on intelligent internet edge and the secretary of the IEEE Internet Technical Committee. He also serves as the Chair for conferences and workshops, such as IWQoS 2021, MSN 2020, BRAINS 2020, ICC 2021 NMIC Symposium, ICCN 2019/2020, and NMIC 2019/2020. He organizes special issues for the leading magazines and journals, such as *IEEE Communications Magazine*, IEEE NETWORK, and IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING (TNSE).



**Nenghai Yu** received the bachelor's degree from Nanjing University of Posts and Telecommunications, Nanjing, China, in 1987, the M.E. degree from Tsinghua University, Beijing, China, in 1992, and the Ph.D. degree from the Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC), Hefei, China, in 2004. Currently, he is a Professor with the School of Cyber Science and Technology, USTC. He is the Executive Dean of the School of Cyber Security, USTC, and the Director of the

Information Processing Center, USTC. He has authored or coauthored more than 130 papers in journals and international conferences. His research interests include multimedia security, video processing, information hiding, and quantum networking.



**Qibin Sun** (Fellow, IEEE) received the Ph.D. degree from the Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC), in 1997. He is currently a Professor with the School of Cyber Science and Technology, USTC. He has published more than 120 papers in international journals and conferences. His research interests include multimedia security, network intelligence and security, and quantum networking.



**Jun Lu** received the bachelor's degree from Southeast University in 1985 and the master's degree from the Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC), in 1988. Currently, he is a Professor with the School of Cyber Science and Technology and the Department of EEIS, USTC. He is also the President of Jiaying University. He is an Academician with Chinese Academy of Engineering (CAE). His research interests include theoretical research and system development in the

field of integrated electronic information systems, networks, and information security.