

# Concurrent Entanglement Routing for Quantum Networks: Model and Designs

Shouqian Shi<sup>1b</sup>, Xiaoxue Zhang<sup>1b</sup>, *Graduate Student Member, IEEE*,  
and Chen Qian<sup>1b</sup>, *Senior Member, IEEE, ACM*

**Abstract**—Quantum entanglement enables important computing applications such as quantum key distribution. Based on quantum entanglement, quantum networks are built to provide long-distance secret sharing between two remote communication parties. Establishing a multi-hop quantum entanglement exhibits a high failure rate, and existing quantum networks rely on trusted repeater nodes to transmit quantum bits. However, when the scale of a quantum network increases, it requires end-to-end multi-hop quantum entanglements in order to deliver secret bits without letting the repeaters know the secret bits. This work focuses on the entanglement routing problem, whose objective is to build long-distance entanglements via untrusted repeaters for concurrent source-destination pairs through multiple hops. Different from existing work that analyzes the traditional routing techniques on special network topologies, we present a comprehensive entanglement routing model that reflects the differences between quantum networks and classical networks as well as a new entanglement routing algorithm that utilizes the unique properties of quantum networks. Evaluation results show that the proposed algorithm Q-CAST increases the number of successful long-distance entanglements by a big margin compared to other methods. The model and simulator developed by this work may encourage more network researchers to study the entanglement routing problem.

**Index Terms**—Quantum internet, quantum networks, entanglement routing, network modeling.

## I. INTRODUCTION

SECURE information exchange via quantum networks has been proposed, studied, and validated since 1980s [2], [3], [4], [5], [6], [7], [8], [9] and many experimental studies have demonstrated that long-distance secret sharing via quantum networks can become successful in reality, such as the DARPA quantum network [6], [7], SECOQC Vienna QKD network [8], the Tokyo QKD network [5], and the satellite quantum network in China [9]. A quantum network (the foundation of the quantum Internet) is an interconnection of quantum processors and repeaters that can generate, exchange, and process quantum information [10], [11], [12], [13]. It transmits information in the form of quantum bits, called *qubits*, and

Manuscript received 23 November 2022; revised 23 July 2023; accepted 9 December 2023; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. M. Kim. Date of publication 5 January 2024; date of current version 18 June 2024. This work was supported in part by the National Science Foundation under Grant 1932447, Grant 2114113, and Grant 2322919; and in part by the Department of Energy (DoE) under Grant DE-SC0022069. The preliminary version of this paper [DOI: 10.1145/3387514.3405853] appeared at the Proceedings of ACM SIGCOMM, 2020. (Corresponding author: Chen Qian.)

Shouqian Shi was with the University California at Santa Cruz, Santa Cruz, CA 95064 USA. He is now with Google LLC, Sunnyvale, CA 94089 USA (e-mail: sshi27@ucsc.edu).

Xiaoxue Zhang and Chen Qian are with the Department of Computer Science and Engineering, University California at Santa Cruz, Santa Cruz, CA 95064 USA (e-mail: xzhan330@ucsc.edu; cqian12@ucsc.edu).

Digital Object Identifier 10.1109/TNET.2023.3343748

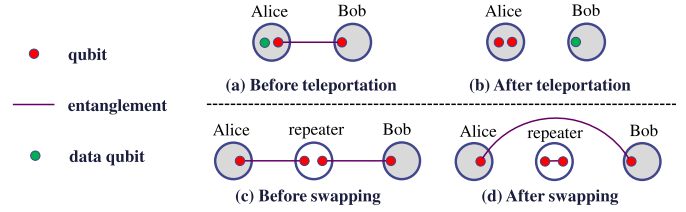


Fig. 1. (a-b) Quantum teleportation to transmit a qubit (consuming a local or distant entanglement). (c-d) Entanglement swapping to build a long-distance entanglement.

stores qubits in quantum memories.<sup>1</sup> Quantum networks are not meant to replace the classical Internet communication. In fact, they supplement the classical Internet and enable a number of important applications such as quantum key distribution (QKD) [2], [3], [16], clock synchronization [17], secure remote computation [18], and distributed consensus [19], most of which cannot be easily achieved by the classical Internet.

Most applications of quantum networks are developed based on two important features of quantum entanglement. 1) Quantum entanglement is inherently private by the laws of quantum mechanics such as the “no-cloning theorem” [20] and hence prevents a third party from eavesdropping the communication [3]. Quantum entanglement enables a perfect solution to the fundamental problem of network security: key distribution (also known as key agreement) [21]. Compared to public key cryptography [22], quantum key distribution (QKD) has provable security in terms of information theory and forward secrecy [11], instead of relying on the computational complexity of certain functions such as factorization. 2) Quantum entanglement provides strong correlation and instantaneous coordination of the communication parties. Hence, quantum entanglement can achieve tasks that are difficult to coordinate in classical networks, and a well-known one is *quantum teleportation*, as shown in Fig. 1 (a) and (b). If a pair of entangled qubits are shared by Alice and Bob, then Alice can send one qubit, encoded with several bits of secret information, to Bob with the help of quantum measurement and the classical Internet [23]. Hence, QKD can be achieved via quantum entanglement.

Both academia and industry have a time-to-time debate on when a practical quantum computer will be available with a sufficient amount of qubits to implement the proposed quantum algorithms, such as Shor’s integer factorization [24]. It seems that well-functioning quantum computers might not become available in the near future. However, many

<sup>1</sup>e.g., transmitting a pair of entangled photons and storing the entanglement state into a pair of nitrogen-vacancy centers in two remote diamonds [14], [15].

applications of quantum networks can be implemented with one or two qubits. Considering the QKD example, we are able to distribute a secret bit with only one entanglement pair. By repeating the 1-pair QKD process we can generate secret keys with a sufficient length.

To generate a quantum entanglement between two parties Alice and Bob, an entangled pair of photons are created, and each photon is sent to a party through a channel, such as an optical fiber. However, the optical fiber is inherently lossy and the success rate  $p$  of establishing an entanglement pair decays exponentially with the physical distance between the two parties [25], [26]. Hence, to increase the success rate of *long-distance* quantum entanglement, a number of *quantum repeaters* need to be deployed between two long-distance communication parties [11], [26]. Many existing quantum networks [5], [6], [8], [9] rely on “*trusted repeaters*” to relay entanglements. Each trusted repeater gets the *actual data qubit* teleported from the sender and teleports the data qubit to the receiver, similar to the “store-and-forward” process in classical networks. A more attractive approach is to use quantum swapping [4], [27], [28]. As shown in Fig. 1 (c) and (d), via entanglement swapping, a quantum repeater that holds entanglements to both Alice and Bob can turn the two one-hop entanglements into one direct entanglement between Alice and Bob. Multi-hop swapping is also possible with a path of repeaters holding entanglements with their predecessors and successors. During quantum swapping, a repeater does not know the qubit information hence it does not have to be trusted.

This work focuses on a key problem called *entanglement routing*, whose objective is to build long-distance entanglements through multiple hops of quantum repeaters and entanglement swapping, even if the *repeaters may be untrusted or corrupted* [29], [30]. Entanglement routing has not been thoroughly investigated but is *necessary in future large-scale quantum networks*: When a quantum network scale increases, similar to the Internet, users do not always trust all forwarding devices between the source and destination or some trusted repeaters may be corrupted. In addition, a large number of trusted repeaters increase the attack space and the vulnerability of the whole system. Entanglement routing finds an end-to-end path of concurrent quantum entanglement through a number of repeaters and performs quantum swapping without letting the repeaters know the data bits. This can be considered on the network layer of a quantum network [15]. Existing works that investigate the entanglement routing problem of quantum networks are limited to analyzing the traditional routing algorithms (Dijkstra shortest paths, multipath routing, and greedy routing) on special network topologies (ring, sphere, or grid), such as the very recent ones [29], [30].

Similar to other network routing problems, entanglement routing is a distributed algorithm design problem to utilize the underlying link-layer models [15], [29], [30]. While the physical layer and link layer studies of quantum networks require experimental validations on special and expensive hardware, the entanglement routing algorithms can be comprehensively evaluated via simulations as long as the link-layer model reflects the practical physical facts, similar to prior studies for

intra-/inter-domain routing, wireless multi-hop routing [31], data center routing [32], [33], [34], etc.

To our knowledge, this is the first work of a comprehensive protocol design specifically for entanglement routing in quantum networks, with new models, new metrics, and new algorithms, working on arbitrary network topologies – and no existing work has considered all of above. We present a comprehensive entanglement routing model that reflects the difference between quantum networks and classical networks and propose new entanglement routing designs that utilize the unique properties of quantum networks. We propose a few routing metrics that particularly fit quantum networks instead of using hop-count and physical distance. The proposed algorithms include realistic protocol-design considerations such as arbitrary network topologies, multiple concurrent sources and destinations to compete for resources, link state exchanges, and limited qubit capacity of each node, most of which have not been considered by prior studies. Evaluation results show that the proposed algorithm Q-CAST increases the number of successful long-distance entanglements by a big margin compared to other known methods. More importantly, this study may encourage more network researchers to study the entanglement routing problem. We present and clarify the models and problems of entanglement routing, with the comparison of similar terms and concepts used in classical network research. A simulator with algorithm implementation, topology generation, statistics, and network visualization functions is available on this link [35].

The rest of this paper is organized as follows. Section II presents the related work of quantum network routing and Section III introduces the network model. We present the algorithm designs in Section IV. The evaluation results are shown in Section VI. We discuss some related issues in Section VII and conclude this work in Section VIII.

## II. RELATED WORK

Quantum information exchange has been proposed, studied, and validated for more than 20 years [2], [3], [4], [5], [6], [8], [9]. The concept of quantum networks is first introduced by the DARPA quantum network project aiming to implement secure communication in the early 2000s [6]. Recent implementations include the SECOQC Vienna QKD network [8], the Tokyo QKD network [5], and China’s satellite quantum network [9]. These experimental works rely on trusted repeaters.

In order to design future large quantum networks in which repeaters may not trust each other, one fundamental problem is to route quantum entanglements with high reliability in quantum repeater networks [36]. Van Meter et al. studies applying Dijkstra algorithm to repeater network [37]. Pirandola et al. discuss the limits of repeaterless quantum communication [26] and propose multi-path routing in a diamond topology [38]. Schoute et al. [39] propose a framework to study quantum network routing. However, their discussion is only limited to ring or sphere topology. Das et al. [40] compare different special topologies for entanglement routing. Caleffi [41] studies the optimal routing problem in a chain of repeaters. Pant et al. [29] propose solutions for entanglement routing in grid networks. [30] proposes virtual-path based greedy

routing in ring and grid networks and also a continuous timing model. Vardoyan et al. [42] study a quantum entanglement switch in a star topology. All these studies assume specialized network topologies such as a grid or ring, which may be over-simplified. The topologies of practical quantum networks may be arbitrary graphs because 1) the end hosts in quantum networks must exist on specified locations according to application requirements, instead of following certain topologies; 2) deploying unnecessary devices just to create a certain topology is a waste of resource.

Recently, Dahlberg et al. [15] provide a reference model of the quantum network stack, which contains the physical layer, link layer, network layer, and transport layer. Based on that, they provide a reliable physical and link layer protocol for quantum networks on the NV hardware platform. The routing algorithms proposed in our paper fit in the ‘network layer’ [15] to provide the concurrent entanglement routing solutions, leveraging the services in the quantum link layer. Some online routing algorithms are proposed based on Q-PAST [43], [44], and some offline routing algorithms are proposed based on Q-CAST [45], [46]. Transport layer algorithms and protocols are explored [47], [48]. Opportunistic routing has also been proposed for quantum networks [49].

### III. NETWORK AND SECURITY MODELS

The network model used in this study follows the facts from existing physical experiments [4], [27], [28], [50] and the corresponding studies [29], [39], [40] to reflect a practical quantum network. Compared to prior models used in existing studies of quantum network performance [29], [30], [39], [40], this model includes many practical considerations, *e.g.* the dynamics of quantum links, definition and comparison of different routing metrics, concurrent source-destination pairs, limited qubit capacity of each node, clear differentiation of the network topology and link state information, and limited link state propagation in a time slot.

#### A. Basic Physical Concepts

Though the Physical rules and techniques are out of the scope of this work, we present some fundamental concepts related to quantum entanglement routing. More detailed explanations can be found in [51].

**1. Quantum entanglement.** Quantum entanglement is a strong correlation between two or more quantum systems, even when they are physically separated. When two or more qubits become entangled, their quantum states become intertwined and cannot be described independently of each other, regardless of the distance between them. For example, suppose two maximally entangled qubits  $A$  and  $B$  described as  $|\Phi\rangle = (|0\rangle_A|0\rangle_B + |1\rangle_A|1\rangle_B)/\sqrt{2}$ . Here,  $|\Phi\rangle$  denotes the state of the whole system, and  $|0\rangle_A$  means  $A$  is at state  $|0\rangle$ . This formula means that if a measurement performed on qubit  $A$  yields state  $|1\rangle$ , the measurement on qubit  $B$  will always result in  $|1\rangle$ , vice versa. Note that the possible state of a two maximally entangled qubits can be any unit vector in the space defined by the following four basis in Eq. 1, called **Bell basis** [3].

$$|\Phi^+\rangle = (|0\rangle_A|0\rangle_B + |1\rangle_A|1\rangle_B)/\sqrt{2}$$

$$|\Phi^-\rangle = (|0\rangle_A|0\rangle_B - |1\rangle_A|1\rangle_B)/\sqrt{2}$$

$$|\Psi^+\rangle = (|0\rangle_A|1\rangle_B + |1\rangle_A|0\rangle_B)/\sqrt{2}$$

$$|\Psi^-\rangle = (|0\rangle_A|1\rangle_B - |1\rangle_A|0\rangle_B)/\sqrt{2} \quad (1)$$

#### 2. Quantum teleportation and entanglement swapping

As shown in Fig. 2, suppose there are two communication parties Alice and Bob and three memory qubits  $A$ ,  $B$ , and  $C$ .  $A$  and  $C$  are held by Alice, while  $B$  is held by Bob, who may be far away from Alice. Further, assume  $A$  and  $B$  are entangled. **Quantum teleportation** enables transmitting the full quantum state from  $C$  to  $B$ , with the help of classical communication, consuming the entanglement  $A$ - $B$ . The workflow of quantum teleportation is as follows. Alice first carefully measures the two-qubit system  $A$  and  $C$  to probabilistically project the system of  $A$  and  $C$  to one of the four Bell basis in Eq. 1, this operation is called **Bell state measurement (BSM)** [3]. BSM does not reveal the full quantum state of  $A$  or  $C$ . The measurement result is 2 classical bits encoding one of the 4 possible Bell states. Alice then sends the 2-bit result to Bob via classical communication in plain text. Then Bob applies 0 to 2 quantum gate operations to qubit  $B$  according to the BSM result. The quantum gates are Pauli-Z and Pauli-X [3]. Then, qubit  $B$  holds the same quantum state as qubit  $C$  before the teleportation, while  $C$  does not hold its original state any more. Note that 1) the 2-bit result can be sent in plain text without hurting the confidentiality; 2) the quantum teleportation transfers the full quantum state from  $C$  to  $B$ .

With quantum teleportation, the entanglement swapping is easily achieved. As shown in Fig. 3, at the repeater, it teleports the qubit  $R_1$  to qubit  $B$ . Since the full quantum state of  $R_1$  is teleported to  $B$ , the previous entanglements  $A$ - $R_1$  and  $R_2$ - $B$  are then transformed into entanglements  $A$ - $B$  and  $R_1$ - $R_2$ .

#### 3. Fidelity of quantum states.

In practice, quantum states are not ideal: the quantum channels, quantum memories, quantum gate operations are imperfect, and a qubit naturally degrade due to the noises in the environment. With a low-fidelity entanglement, the amount of information we can transmit via quantum teleportation will also be low. A high fidelity entanglement can be achieved via quantum distillation, consuming multiple low-fidelity entanglements.

Though we do not model the fidelity, our model can be extended to allow non-perfect channels, memory, and operations. As for degradation with time, the entanglements are all generated and consumed in the same time slot, the fidelity degradation is constant for one-hop or multi-hop entanglements.

#### B. Network Components

There are three main components in a quantum network [11], [52], explained as follows.

**1. Quantum processors** are similar to the end hosts in classical networks, which are connected to a certain number of other quantum processors by quantum channels to form a quantum network and run the network applications to communicate with each other. Different from classical end hosts,



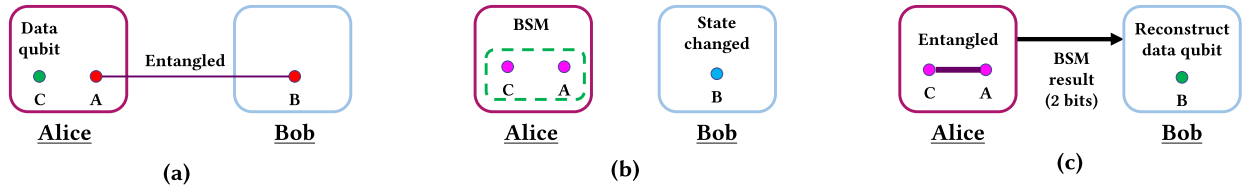


Fig. 2. Quantum teleportation.

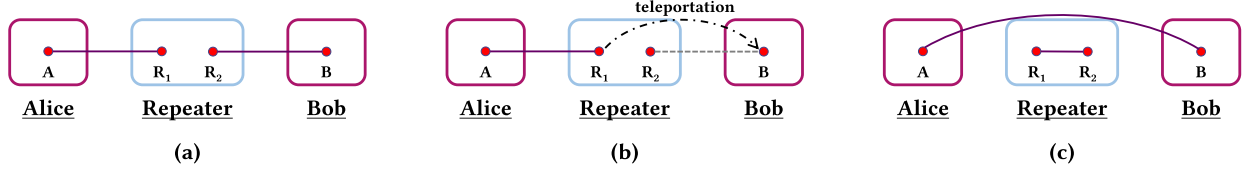


Fig. 3. Entanglement swapping via quantum teleportation.

each quantum processor is equipped with a certain number of memory qubits and necessary hardware to perform quantum entanglement and teleportation on the qubits. All quantum processors are connected via the classical Internet and are able to freely exchange classical information.

**2. Quantum repeaters.** As it is difficult to directly establish an entanglement between two remote quantum processors, quantum repeaters are used as relays. Quantum repeaters support long-distances entanglement sharing via quantum swapping. A quantum repeater may also connect to other repeaters and quantum processors via the classical Internet to exchange the control messages.

Every quantum processor also includes the complete function of a repeater. Hereafter we call both quantum processors and repeaters as *nodes*.

**3. Quantum channels.** A quantum channel connecting two nodes supports the transmission of qubits. The physical material of quantum channels may be polarization-maintaining optical fibers. A quantum channel is inherently lossy: the success rate of each attempt to create an entanglement of a quantum channel  $c$  is  $p_c$ , which decreases exponentially with the physical length of the channel:  $p_c = e^{-\alpha L}$ , where  $L$  is the physical length of the channel and  $\alpha$  is a constant depending on the physical media [11], [26], [29], [53].<sup>2</sup> If an attempt is successful, the two quantum processors share an entanglement pair, and there is a *quantum link* on this channel.

**Network topology.** Consider a network of quantum nodes described by a multigraph  $G = \langle V, E, C \rangle$ .  $V$  is the set of  $n$  nodes. Each node  $u$  is a quantum processor, equipped with a limited number  $Q_u$  of qubits to build quantum links. All nodes are connected via classical networks, *i.e.*, the Internet, and every node has a certain level of classical computing and storage capacity, such as a desktop server.  $E$  is the set of edges in the graph. An edge existing between two nodes means that the two nodes share one or more quantum channels.  $C$  is the set of all quantum channels, each of which is identified by its two end nodes. The number of channels on an edge is called the *width*  $W$  of the edge.

<sup>2</sup>The success rate of a link is determined by the physical layer and link layer, taking into account the channel transmissivity, fidelity of transmitted entanglements, number of permitted entanglement trials in one phase, and the link layer algorithm [15], [29]. In the link layer, a channel is allowed multiple attempts to build a link, and the link is established on the first successful attempt. The  $p_c$  here is the overall success rate.

A node can assign/bind each of its quantum memory qubits to a quantum channel [54], [55], such that no qubit is assigned to more than one channel, and no channel is assigned more than one qubits at the same end of it. Channels that are assigned qubits at its both ends are *bound channels*, other channels are *unbound channels*. There could be more than one bound channels between two nodes. And two neighbor nodes may share multiple quantum links. To create a quantum entanglement, two neighbor nodes make a number of quantum entanglement attempts at the same time on the bound channels connecting them.

### C. Communication and Security Model

For each round of communication, the source and destination are two mutually trusted quantum processors, but they may not trust other nodes. The source aims to deliver secret bits to the destination without letting other repeaters know, via a path of quantum swapping (explained in Sec. III-D). All nodes will follow the protocol but may seek to get the secret information sent from the source to the destination, similar to the “honest-but-curious” model in classic network security. Once an intermediate node measures the information to perform passive eavesdropping, such behavior will be detected by the two endpoints according to the no-cloning theorem.

In addition, an external classical ‘network information server’ may be trusted to maintain the following information and send delta updates to all nodes in the network when necessary: 1) the network topology and 2) the current source-destination pairs (S-D pairs) that need to establish long-distance entanglements. The network information server may work in an honest-but-curious way and it will not know the communication content. Hence even if a network information server may be compromised – which can be detected by classical auditing methods – it will not hurt the confidentiality for previous, on-going, and future communications.

### D. Quantum Swapping via a Path

**Time slots.** For multi-hop entanglement swapping, all nodes on the path need to establish and hold quantum entanglements with its predecessor and successor at the same time. Hence, some level of time synchronization among all nodes is

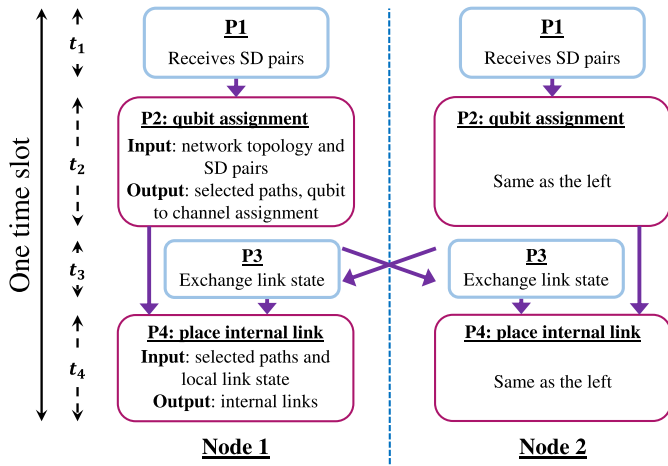


Fig. 4. Phases in a time slot. Nodes 1 and 2 are two arbitrary neighbor nodes and run the same algorithm.

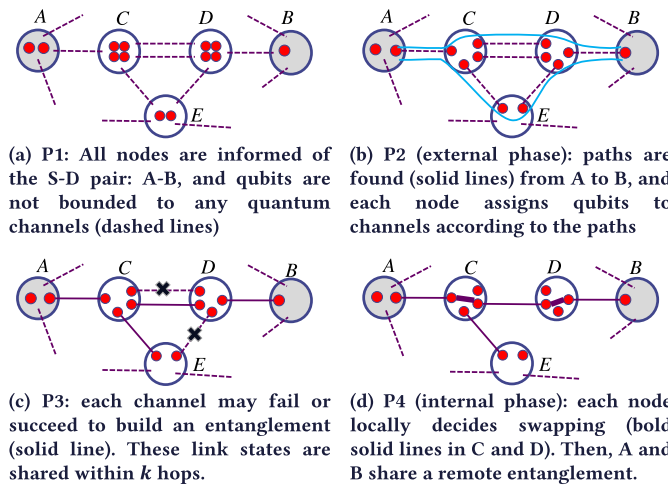


Fig. 5. Phases in a time slot. Entanglement routing aims to build end-to-end paths for S-D pairs (A-B in this example).

necessary, which can be achieved by existing current synchronization protocols via Internet connections. Time is loosely synchronized in *time slots* [29]. The global network topology  $G = \langle V, E, C \rangle$ , which is relatively stable, should be common knowledge for all nodes before any time slot.

As shown in Fig. 4 and Fig. 5, each time slot includes four phases as an extended model from [29]. In Phase One (P1), via the Internet, all nodes receive the information of the current S-D pairs that need to establish long-distance entanglements. As an example in Fig. 5(a), each node has a number of qubits (red dots) and multiple quantum channels (dashed lines) connecting neighbors. Two neighbors may share multiple channels. Suppose  $\langle A, B \rangle$  is the only S-D pair for this time slot, and all nodes are informed of the S-D pair.

Phase Two (P2) is called the external phase [29]. In P2, paths are found for the S-D pairs, according to an identical routing algorithm running on all nodes that produces consistency results. Each node then binds its qubits to channels and attempts to generate quantum entanglements with neighbors on the bound channels [54], [55]. As an example in Fig. 5(b), two paths (solid curves) are calculated to connect A and B. A path is identified by the sequence of the nodes along the path  $v_0, v_1, \dots, v_h$  and the *path width*  $W$ , meaning

each edge of the path has at least  $W$  parallel channels. The path  $\langle (v_0, v_1, \dots, v_h), W \rangle$  is also called a  $(W, h)$ -path, or a  $W$ -path.  $C$ ,  $D$ , and  $E$  are nodes on the paths and work as repeaters. Since qubits are limited resources, some channels are not assigned qubits and thus not used in this time slot. During P2, each channel can make a number  $n_c$  of attempts [56],  $n_c \geq 1$ , until a link is built or timeout. After P2, some quantum links may be created as shown in Fig. 5(b). We call the information of these links as *link states*. Compared to the same term in link-state routing of classical networks [57], the quantum link states are *highly dynamic and nondeterministic*.

In Phase Three (P3), each node knows its own link states via classical communications with its neighbors [29] and shares its link states via the classical network, as shown in Fig. 5(c). Since entanglements will quickly decay, each node can only exchange the link states with a subset of other nodes. P3 only includes classical information exchange.

In Phase Four (P4), also called the internal phase [29], nodes perform entanglement swapping to establish long-distance quantum entanglement using the successful quantum links. Each node *locally* determines the swapping of successful entanglements, which can be considered as placing an *internal link* between two qubits as shown in Fig. 5(d). Each swapping succeeds at a device-dependent probability  $q$ . A and B can successfully share an entanglement qubit pair (*an ebit*) if there is an end-to-end path with both external and internal links as shown in Fig. 1(d).

After P4, the secret bit can be teleported from the source to the destination. Eavesdropping attempts at any repeater will be detected hence *the confidentiality is preserved*.

**Local knowledge of link-state.** P3 and P4 should be short such that the successful entanglements built in P2 do not decay. Hence, it is impractical for a node to know the global link states within such a short time as the classical network has latencies [29]. A practical model is to allow each node to know the link states of its  $k$ -hop neighbors,  $k \geq 1$  [30]. The swapping decisions in P4 thus include the  $k$ -hop link-state information as the input. It is obvious that the routing path selection could be sub-optimal without global link-state knowledge.

**Exclusive qubit/channel reservation.** In P2 of each time slot, to establish a single link on a channel, each end of the channel is assigned a qubit. This qubit-channel assignment is exclusive: one qubit cannot be shared by other channels, and no more qubits can be assigned to a channel. In P4, to generate distant entanglements from local ones, quantum swapping is performed on pairs of links. This quantum swapping is also exclusive and a single link cannot be used for more than one swapping. Hence, the qubits and channels are precious *routing resources* and should be carefully managed.

### E. The Entanglement Routing Problem

This work studies the entanglement routing problem: we are given a quantum network with an arbitrary network graph  $G = \langle V, E, C \rangle$  and a number of source-destination (S-D) pairs  $\langle s_1, d_1 \rangle, \langle s_2, d_2 \rangle, \dots, \langle s_m, d_m \rangle$ , where  $s_i, d_i \in V$ . The

number of memory qubits of a node  $u \in V$  is  $Q_u$ , and each edge  $e \in E$  consists of one or more channels from  $C$ . For each bound channel  $c$ , a link is successfully built at a probability  $p_c$  in P2. In P3, each node gets the link-state information of its  $k$ -hop neighbors. Each node decides the swapping of its internal qubits in P4 *locally*, and each swapping succeeds in probability  $q$ . The objective of entanglement routing is to *maximize the number of ebits* delivered for all S-D pairs in each time slot. Each ebit must be delivered by a long-distance quantum entanglement, built by a path of successful quantum links from the source to the destination. Each S-D pair may share multiple ebits. The number of ebits for one S-D pair in one time slot is also called the *throughput* between the S-D pair. The objective can then be set to maximizing the overall throughput in the network.

This objective does not consider fairness among different S-D pairs, but we show the proposed algorithms achieve a certain level of fairness as in Section VI. In addition, in Section VII we propose a simple extension to our designs to achieve better fairness among S-D pairs.

#### F. Compared With Classic Network Routing

We summarize the differences between quantum entanglement routing and classic network routing. We show that existing routing techniques are **not sufficient** to solve the entanglement routing problem.

**Term clarification.** Edges, channels, and links have *different* definitions in this model, although they are used interchangeably in classic networks. Besides, the network topology and global link states may be considered as similar information in classic routing such as OSPF [57]. However, in a quantum network, while the network topology (nodes and channels) is stable and known to all nodes, the link states (whether the entanglements succeeded) are dynamic and only shared locally in P3 and P4 of each time slot.

**Versus routing in wired packet-switching networks.** Link-state and distance-vector are two main types of routing protocols for packet-switching networks. Main differences: 1) Packet switching relies on either link-state broadcast or multi-round distance vector exchanges to compute the shortest paths. However, in a quantum network, link states are probabilistic and vary in different time slots. There is no time for global link-state broadcast or distance vector convergence, because entanglements on the links will quickly decay. 2) Quantum links are highly unreliable while wired links are relatively reliable. 3) A quantum link cannot be shared by multiple S-D pairs, which is allowed in classic packet switching. If a link is claimed by multiple S-D pairs, it can only satisfy one of them. Hence, the “shortest paths” computed by classic routing will not always be available. 4) Classic packets can be buffered on any node for future transmission. In quantum networks, links on a path must be successful in the same time slot.

**Versus routing in multi-hop wireless networks,** such as mobile ad hoc networks [31] and wireless sensor networks [58]. Main differences: 1) For an ad hoc wireless node, neither the network topology nor global link state is known. For a quantum node, although link state is local

information, the network topology is known in advance via the Internet. 2) An ad hoc wireless node can keep sending a packet until the transmission is successful or a preferred receiver moves close to it. Each quantum link can only be used once and all links on an end-to-end path must be available simultaneously. 3) Existing wireless ad hoc routing methods such as DSR [59], AODV [60], and geographic routing [61], [62], [63] are all packet-switching protocols and do not fit quantum networks. Also, they do not take the global network topology information.

**Versus circuit-switching, virtual circuit, and flow scheduling in SDN.** Circuit switching, virtual circuit, and flow scheduling in software defined networks (SDNs) all need to pre-determine the end-to-end paths and reserve certain resources on the paths, such as [32], [64], [65], and [66], which share similarity with entanglement routing. The main difference is that in a quantum network, though the topology (nodes and channels connecting them) is relatively stable, reserved paths for an S-D pair are not reliable because links may arbitrarily fail. Hence, more robust algorithms are required. Besides, to build a long-distance entanglement along a path, all hops of the path should have one or more success quantum link at the same time. Hence, time is divided into slots and phases for synchronization. Due to the two differences above, the algorithm of entanglement routing is very different from that of circuit-switching, in the following two novel designs: 1) multiple paths are selected in P2, based on global and stable network topology; and 2) path recovery on P4, based on local and probabilistic link states. The recently proposed multipath routing for quantum entanglements [29] is a circuit-switching style protocol and will be compared to our work in Sec. VI.

## IV. ENTANGLEMENT ROUTING ALGORITHMS

The proposed entanglement routing algorithms utilize the unique properties of quantum networks that have not been explored in classic network routing. Compared to recent quantum network studies [29], [30], [39], [40], the proposed protocols follow a standard protocol-design approach and use more realistic network models: arbitrary network topologies, multiple concurrent S-D pairs to compete for links, link state exchanges, and limited qubit capacity of each node.

### A. Main Ideas

Our design is based on the following three innovative ideas to utilize the **unique features** of a quantum network:

**1. Path computation based on global topology and path recovery based on local link states.** The quantum network graph  $G = \langle V, E, C \rangle$  is relatively stable and hence can be known to every node. However, the link states are highly dynamic and probabilistic in each time slot. The frequent link state changes cannot be propagated throughout the whole network, especially when the entanglements decay quickly.

**2. Minimize the entanglement decay via distributed algorithms.** Under the above idea 1 and the time slot model, the proposed algorithms are carefully designed to avoid any

global communication or any serial quantum/classical operations while the linkwise entanglements are preserved in quantum memories. Especially, in P2, the paths are selected **before** any entanglement is established; in P3, the link states of each node are propagated to a **constant** number of neighbors  $k$ , regardless of the scale network scale; and in P4, each node **independently** and **concurrently** make swapping decisions. We show in Section VI that the local link state exchange and local swapping decision match the unique features of quantum network and perform well.

**3. Wide paths are preferred.** Recall that on a  $W$ -path, each edge has at least  $W$  parallel channels. Fig. 6(a) shows an example of a 2-path from  $A$  to  $B$ . Compared to two disjoint paths shown in Fig. 6(b), which cost the same amount of qubits and channels, the wide path is more reliable because it only fails when two links fail simultaneously at a single hop. To achieve high throughput on a path with  $W > 1$ , nodes should share a consensus on how to perform swapping (place internal links in Fig. 6) instead of making choice randomly. Each channel is assigned a globally unique ID. During P4, each node places an internal link between the link with the smallest ID to its predecessor and the link with the smallest ID to its successor. And it repeats this process until no internal link can be made for this path.

Formally, we may define a *routing metric*, called expected throughput (EXT)  $E_t$ , to quantify an end-to-end path on the network topology. The EXT of a path is the expected number of ebits established in a single time slot. For a  $(W, h)$ -path  $P$ , suppose the success rate of a single channel on the  $i$ -th hop is  $p_i$ , where  $i \in \{1, 2, \dots, h\}$ . We denote the probability of the  $k$ -th hop on the path having exactly  $i$  successful links as  $Q_k^i$ , and the probability of the channel with the fewest successful links among the first  $k$  hops of path  $P$  has  $i$  successful links as  $P_k^i$ . Then we get the recursive formula set, for  $i \in \{1, 2, \dots, W\}$  and  $k \in \{1, 2, \dots, h\}$ :

$$\begin{aligned} Q_k^i &= \binom{W}{i} p_k^i (1 - p_k)^{W-i} \\ P_1^i &= Q_1^i \\ P_k^i &= P_{k-1}^i \cdot \sum_{l=i}^W Q_k^l + Q_k^i \cdot \sum_{l=i+1}^W P_{k-1}^l \end{aligned} \quad (2)$$

Further, for a  $(W, h)$ -path  $P$ , the expected number of entanglements established in a single time slot is calculated by the weighted sum of probability  $P_h^i$ ,  $i \in \{1, 2, \dots, W\}$ . Considering each successful entanglement requires  $h-1$  swapping operations, we get the EXT:

$$E_t = q^{h-1} \cdot \sum_{i=1}^W i \cdot P_h^i \quad (3)$$

We show some numerical results. For simplicity, we set  $p_1, p_2, \dots, p_h = p$ , and let  $q$  be 0.95, and let  $p$  be 0.9 or 0.6. We vary  $W$  from 1 to 3 and the  $h$  from 1 to 10, and the results of the EXT of a  $W$ -path are shown in Figures 7 and 8. It is obvious that a  $W$ -path has a significant improvement of EXT over a 1-path, for more than a factor of  $W$ .

**4. Offline computation versus contention-aware online path selection.** In different time slots, the S-D pairs may

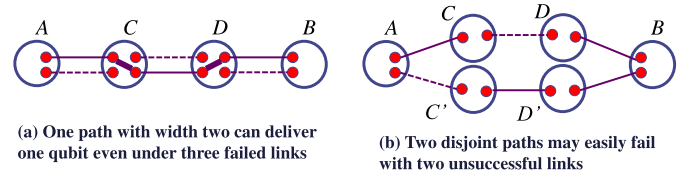


Fig. 6. A wide path (subfigure a) is more reliable than disjoint paths (subfigure b) using the same resource.

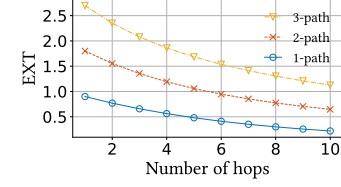


Fig. 7. EXT,  $p = 0.9$ ,  $q = 0.95$ .

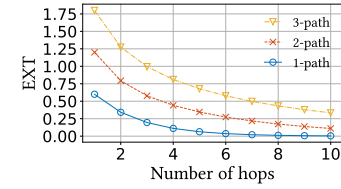


Fig. 8. EXT,  $p = 0.6$ ,  $q = 0.95$ .

be different. We propose two approaches to select paths for S-D pairs in each time slot. The first approach utilizes *offline computation*, which happens before any time slot, e.g., during system initialization. Multiple paths for *each* potential S-D pair are pre-computed and stored by all nodes as common knowledge. In P2 of each time slot, nodes select the pre-computed paths for current S-D pairs. The *contention-aware* online algorithm, however, does not pre-compute the paths for all S-D pairs. At each time slot, the algorithm finds contention-free paths for current S-D pairs. A set of paths are ‘contention-free’ if the network can simultaneously satisfy the qubit and channel requirement for all the paths in full width.

#### B. Q-PASS: Pre-Computed pAth Selection and Segment-Based Recovery

**1) Algorithm Overview:** We present the algorithm Q-PASS, whose workflow follows the four-phase time slot model with an additional offline phase. The core idea of Q-PASS is to pre-compute potential ‘good’ paths between *all possible S-D pairs* based on the network topology  $G$ . Then in each time slot, every node uses an online algorithm to make qubit-to-channel assignments based on the pre-computed paths of *current S-D pairs* and make local swapping decisions based on local link states. The design includes both offline and online algorithms.

The offline computation happens at the system initialization and after the network topology changes. The results of an offline phase can be used by many succeeding time slots until a topology change happens. Hence, we may assume the time for an offline period is sufficiently long. The offline algorithm runs at the network information server, which is honest but curious, with replica servers for robustness. These servers connect to all quantum nodes via classical networks. The output of the offline algorithm is the ‘candidate paths’ for all possible S-D pairs. The candidate paths of each S-D pair are paths connecting the S-D nodes and with the smallest values of the selected metric.



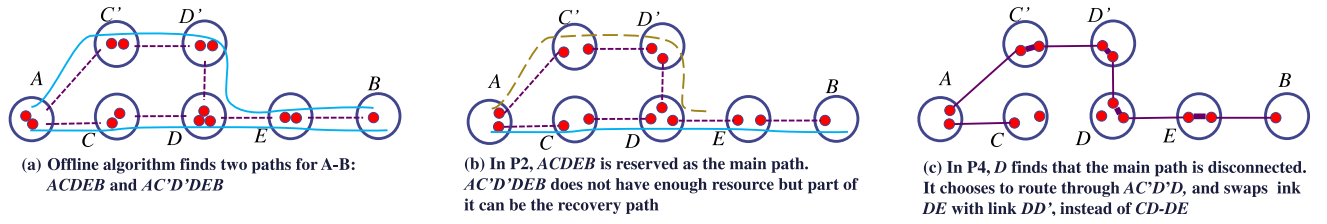


Fig. 9. Example of Q-PASS. Suppose  $\langle A, B \rangle$  is the only S-D pair.

The algorithm of each time slot follows the four-phase time slot model shown in Fig. 4 and runs on each node in a distributed and concurrent manner. It should be fast and only use the  $k$ -hop link-state information. P1 and P3 only include standard processes and do not have special algorithmic designs. Q-PASS P2 takes the candidate paths from the offline algorithm and the S-D pairs as the input. It computes a number of selected paths for the S-D pairs and its local qubit-to-channel assignment. Note that the inputs are globally consistent on all nodes. Hence, the selected paths are also consistent on different nodes. The assignment will produce a number of successful links in P2. And in P3, nodes exchange the link states with their  $k$ -hop neighbors. Q-PASS P4 uses the selected paths and link state information as the input to compute the swapping decisions (i.e., internal links). After P4, possible long-distance entanglement can be built for S-D pairs. We present the algorithms in detail.

2) *Offline Path Computation*: The offline algorithm should find multiple paths for each S-D pair to provide multiple candidate paths in P2 of each time slot. We use Yen's algorithm [67] to get multiple paths for each pair. Note that the results of Yen's algorithm are not contention-free: the paths may overlap at nodes or channels, and in a single time slot, the network may not have enough qubits or channels to satisfy all the candidate paths for an S-D pair.

Yen's algorithm implicitly requires a selection of the routing metric. As shown in Equ. 2, computing the proposed routing metric EXT involves recursions, which may be prohibitively slow for multi-path computation for all possible S-D pairs. Hence, we propose three routing metrics, which are suboptimal in overall throughput but faster to compute. 1) **Sum of node distances (SumDist)**. SumDist is computed as  $\sum L_i$ , where  $L_i$  is the length of any channel on the  $i$ -th hop of the path. As the success rate of a channel decreases exponentially with the physical distance  $L$ , SumDist can partially reflect the difficulty of a path. 2) **Creation Rate (CR)**. CR is computed as  $\sum 1/p_i$ , where  $p_i$  is the success rate of any channel on the  $i$ -th hop of the path. Compared to SumDist, CR further considers the path width. 3) **Bottleneck capacity (BotCap)**. From Figures 7 and 8, the path width  $W$  has a greater impact on the path quality. The BotCap metric is  $-W$ , prefers wider paths over narrower paths, and uses the CR to break ties for paths with the same width. We consider the routing metric as a design parameter, and their efficiency is compared in Section VI.

For each possible S-D pair, the server running the offline algorithm will use Yen's algorithm to get  $N = 25$  paths (*offline paths*) for the pair and tell *each node in the network* about the resulting paths. An example is shown in Fig. 9(a), the offline

algorithm finds  $ACDEB$  and  $AC'D'DEB$  as two candidate paths.  $N$  will grow by 50% percent in the next offline phase if the paths happen to be not enough for a pair.

3) *P2 Algorithm of Q-PASS*: The P2 algorithm runs on each node locally. The inputs are all the offline paths  $P$  (known before P1) and the S-D pairs (received in P1)  $O = \{o_i\}$ , where  $o_i$  is an S-D pair  $\langle s_i, d_i \rangle$ . The output is an ordered list of selected paths  $P'$ , each of which connects a single S-D pair in  $O$ . According to the output path list, each node performs the local qubit-to-channel assignment and tries to establish entanglements on the bound channels with neighbor nodes to build quantum links on these paths. Since  $P$  and  $O$  are globally known for all nodes, the output  $P'$  is also consistent on different nodes, similar to the global consistency of classical link-state routing.

The algorithm consists of two steps. **Step 1)** The paths computed from the offline algorithm for all S-D pairs are retrieved and put into a priority queue, ordered by the selected routing cost metric. Then from the path with the lowest routing cost to the highest, channels and qubits taken by the path are reserved exclusively. If a path has width  $w$  by the offline algorithm, but currently available resource can only support width  $0 \leq w' < w$ , then the path is reinserted to the queue with an updated metric calculated from  $w'$ . This process ends until no paths can be fully satisfied. The paths selected in Step 1 are called *major paths*. **Step 2)** After Step 1, the queue contains all unsatisfiable paths in the ascending order of the routing metric. Each unsatisfiable path, however, may contain one or more satisfiable segments or 'partial paths'. The partial paths can be used to recover link failures for the major paths, and thus are called *recovery paths*. The qubits and channels for recovery paths are reserved in the order of its priority in the queue.

For example in Fig. 9,  $ACDEB$  and  $AC'D'DEB$  are two candidate paths. In Step 1,  $ACDEB$  and  $AC'D'DEB$  are put into a priority queue, and  $ACDEB$  is more prioritized than the other and is reserved as the major path. Since  $D$ ,  $E$ , and  $B$  do not have enough resources for  $AC'D'DEB$ , Step 1 stops. In Step 2,  $AC'D'D$  is reserved as a recovery path. When the two steps finish, all nodes know the same set of selected major paths and recovery paths because they share the same set of inputs: network topology, S-D pairs, and offline paths. Hence, each node assigns its qubits to the corresponding channels and try to generate quantum links together with the neighbors. For example, node  $A$  in Fig. 9(b) will assign one qubit to the channel to  $C$  and another to the channel to  $C'$ , and try to entangle with  $C$  and  $C'$  via channel  $AC$  and  $AC'$  respectively.

4) *P4 Algorithm of Q-PASS*: If the entanglement attempts in P2 always succeed, each node just performs entanglement



swapping to connect the links on the major paths, and the whole paths will be successful. In practice, however, link failures happen at a high probability and are not predictable. The P4 algorithm focuses on the recovery of broken major paths based on the recovery paths established in P2. The inputs of P4 algorithm are: 1) S-D pairs from P1, 2) a major path list and a recovery path list from P2, and 3) the  $k$ -hop link states of this node from P3.

We propose segment-based path recovery for P4. On each node, each major path given by P2  $\langle (v_0, v_1, \dots, v_h), W \rangle$  is divided into  $\lceil h/(k+1) \rceil$  segments, each with width  $W$ :  $(v_0, v_1, \dots, v_{k+1}), (v_{k+1}, v_{k+2}, \dots, v_{2k+2}), \dots, (v_{\lceil h/(k+1) \rceil(k+1)-1}, v_{\lceil h/(k+1) \rceil(k+1)}, \dots, v_{h-1}, v_h)$ . The length of the segments is set to  $k+1$  such that each node knows the states of all links on the segment containing it, via the  $k$ -hop link states received in P3. Then for segment  $(v_{i_0}, v_{i_1}, \dots, v_{i_{k+1}})$ , each node on it finds paths connecting the two ends,  $v_{i_0}$  and  $v_{i_{k+1}}$ , using successful links in the  $k$ -hop neighborhood.

An example is shown in Fig. 9. Assume  $k = 1$ , and thus each node knows the link states of its 1-hop neighbors. The major path  $ACDEB$  is divided into two segments  $ACD$  and  $DEB$ , such that all nodes on a single segment know this segment is successful or not. If not, they will try to use a recovery path. In this example,  $A$ ,  $C$ , and  $D$  know link  $C-D$  fails. Hence, the recovery path  $AC'D'D$  is taken by  $D$ . The distributed recovery path selection is consistent among all nodes because recovery paths are found from local link states known to all involved nodes, and the recovery paths are ordered deterministically via the specified routing metric.

### C. Q-CAST: Contention-Free pAth Selection at runTime

The offline algorithm in Q-PASS has two fundamental disadvantages. 1) It has to compute candidate paths for  $n(n-1)/2$  pairs because it does not know the runtime S-D pairs. 2) The candidate paths exhibit a low utilization rate due to severe resource contention among them. Q-CAST does not require any offline computation and always finds the paths if only paths exist in the residual graph. For example in Fig. 10(a),  $AB$  and  $A'B'$  are two S-D pairs. The offline algorithm of Q-PASS finds 25 paths for  $AB$ , passing by nodes  $M_1, \dots, M_{25}$ . But a single path  $A', M_1, \dots, M_{25}, B'$  takes all available qubits on  $M_i$ , and thus in the residual graph, all 25 candidate paths of  $AB$  fail, though paths  $AA'B$  and  $AB'B$  exist outside the offline paths, which are correctly found and reserved by Q-CAST online algorithm as shown in Fig. 10(b). Due to unpredictable combinations of S-D pairs and the resulting residual graphs, it is hard to pre-calculate and store the paths for all S-D pair combinations.

1) *Algorithm Overview*: Q-CAST does not require any offline computation and follows the four-phase model in Fig. 4. Q-CAST P1 and P3 are standard procedures similar to those of Q-PASS. The inputs of Q-CAST P2 are the network topology and the S-D pairs. In P2, Q-CAST selects major paths for each S-D pair, *without resource contention*. Besides, contention-free recovery paths are also selected in P2. P4 takes the major paths and recovery paths from P2 and the link states from P3 to compute the swapping decisions.

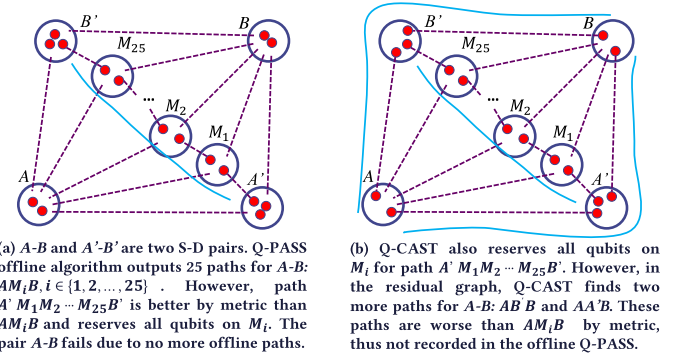


Fig. 10. Comparison of Q-PASS and Q-CAST.

2) *P2 Algorithm of Q-CAST*: The core task for Q-CAST P2 is to find multiple paths based on the knowledge of S-D pairs, and the paths should be contention-free on qubits and channels. Yen's algorithm [67] does not satisfy the requirements because its output paths are highly overlapped. Note, Q-PASS uses Yen's algorithm to find offline paths because the resulting overlapped path naturally provides small detours, serving as recovery paths for major paths. We propose to search multiple contention-free paths for online S-D pairs using a greedy algorithm, which runs as follows. **Step 1**) For every S-D pair, it uses the Extended Dijkstra's algorithm (described later) to find the best path in terms of the routing metric EXT (Equation 3) between this pair. **Step 2**) Among the best paths of all S-D pairs, it further selects the path with the highest EXT and reserve the resources (qubits and channels) of this path, and the network topology is updated to the residual graph by removing the reserved resources. Steps 1) and 2) are repeated with the residual graph, until no more path can be found, or the number of paths exceeds 200 – a value limiting the number of paths to avoid unnecessary computation. We call this algorithm as *Greedy EDA (G-EDA)*.

The above process aims to maximize the network throughput but does not consider fairness among S-D pairs. We will discuss how to balance throughput and fairness in Section VII, and this could also be a future research topic.

**The optimal routing metric.** To find the optimal path under the EXT metric in a quantum network, the classical Dijkstra's algorithm fails because it only finds the shortest path when the routing metric is 'additive'. Here, additive means the *sum* of the costs of all edges on the path is exactly the cost of the whole path. Obviously, the EXT  $E_t$  computed by Equation 3 is not additive. We propose the *Extended Dijkstra's algorithm (EDA)* to find the best path between any S-D pair for any non-additive but monotonic routing metric. The resulting path gives the maximum evaluation value among all possible paths between the S-D pair, with respect to a routing metric function  $e$ . The input of  $e$  is a path  $\langle p, W \rangle$ , and the output is the path quality evaluation value.

Similar to the original Dijkstra algorithm, EDA also constructs an optimal spanning tree rooted at the source node  $s$ . At the beginning, the *visited set* only includes  $s$ . The evaluation value from  $s$  to an unvisited node  $x$  is set as 0 or the evaluation value  $e(s, x)$  of the edge  $(s, x)$  if  $s$  and  $x$  are neighbors. Each time, the node  $y$  with the maximum evaluation value to  $s$  is added to the visited set and the evaluation values

from  $s$  to any other node  $x$  are updated if  $x$  and  $y$  are neighbors. The algorithm stops when the destination is visited.

We skip the proof of the correctness of EDA due to space limit. Its correctness rely on a fact that the evaluation function  $e$  of a path  $\langle p, W \rangle$  should *monotonically decrease* when extending  $p$  to a longer path  $p'$  by adding another node at the end of  $p$ . Since we use  $E_t$  as the evaluation function, we explain the monotonicity of  $E_t$  without a strict proof. As the  $W$ -path  $p$  grows,  $W$  may stay unchanged or decrease because the new edge may be narrower than  $W$ . In addition, adding one more hop means more hops to be transmitted. Neither of the above can increase  $E_t$ .

Different from the original Dijkstra algorithm, updating the path by adding one hop may cause a re-evaluation of the entire path, rather than simply adding the cost of a link. To avoid expensive recalculation for path updates, one optimization can be applied when using  $E_t$  as the evaluation function. If a  $(W, h)$ -path  $p$  grows by one hop with width  $\geq W$ , then the width of the new path  $p'$  stays unchanged to be  $W$ . Hence, in the calculation of  $E_t(p') = q^{h+1} \cdot \sum_{i=1}^W i \cdot P_{h+1}^i$ , the original values  $P_h^i$  when calculating  $E_t(p)$  can be re-used, which significantly reduces the complexity by performing just one iteration.

**Bound the path length.** We set the upper-bound threshold  $h_m$  for the hopcount of major paths to ensure bounded searching in EDA. During EDA, any path with hopcount larger than  $h_m$  is ignored because it is unlikely to be a good path. The value of  $h_m$  can be determined at system initialization. For a new network  $G$ , 100 pairs of nodes are randomly selected. Then, multipath routing is performed via G-EDA for each pair with  $h_m = \text{inf}$ . Then  $h_m$  is set to equal the largest hopcount of the resulting paths whose  $E_t \geq 1$ .

**Recovery paths.** After finding the major paths via G-EDA, the remaining qubits and channels can be utilized to construct *recovery paths*, each of which ends at two nodes (denote as *switch nodes*) on a single major path. The switch nodes should be no more than  $k$  hops away on a major path, where  $k$  is the link state range, because in P4, the two nodes should ensure consistent swapping decisions.

The recovery paths are found as following. For every node  $x$  on a major path, we use EDA to find  $\leq R$  recovery paths between  $x$  and  $y$  in the residual graph, where  $y$  is the 1-hop ahead node on the major path, and  $R$  is a small constant parameter. When all nodes are processed, the algorithm will iterates further for the recovery paths that covers  $l$  hops on the major path, for  $l = 2, 3, \dots, k$ . In Fig. 11(a), the major path is  $ACDEB$  and three recovery paths are found.

Every node will assign its qubits based on the reserved major paths and recovery paths, without qubit/channel contention.

3) *P4 Algorithm of Q-CAST*: In P4, each node knows the major paths, the recovery paths, and the  $k$ -hop link states. It then makes the swapping decisions locally. The challenges for Q-PASS P4 still present for Q-CAST P4: probabilistic link failures and non-interactive communication between nodes.

We propose an exclusive-or (*xor*,  $\oplus$ ) based algorithm to recover from potential link failures. We define the xor operator of two set of edges  $E_1, E_2$ :  $E_1 \oplus E_2 = E_1 \cup E_2 \setminus (E_1 \cap E_2)$ .

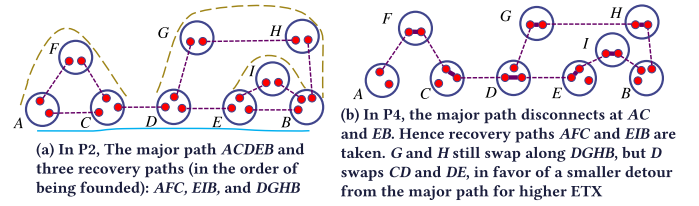


Fig. 11. Example of Q-CAST recovery algorithm.

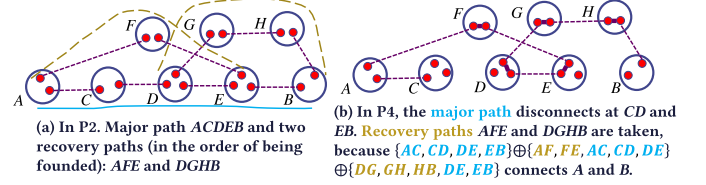


Fig. 12. Example of Q-CAST recovery via exclusive-or.

As both ends (switch nodes) of a recovery path  $p_r$  are on a single major path, a segment of the major path  $p_m$  is covered by  $p_r$ , where  $p_m$  and  $p_r$  form a loop in the graph, called a *recovery loop*. Then, the link recovery algorithm works as following. The major path list is traversed from beginning to end. Each visited major path  $p$  is treated as  $W$  separated 1-paths, where  $W$  is the width of  $p$ . For each separated 1-path, the set  $E$  collects the successful edges of it.  $K$  recovery paths are found, and the edges of the recovery loops of the recovery paths are collected as  $E_{p_1}, E_{p_2}, \dots, E_{p_K}$ , such that the S-D pair is connected in the graph  $\langle V, E \oplus E_{p_1} \oplus E_{p_2} \oplus \dots \oplus E_{p_K} \rangle$ , where  $V$  is the set of nodes on the major path and the  $K$  recovery paths. To break the tie, shorter recovery paths are preferred because shorter paths are more likely to succeed after swapping. The Q-CAST recovery algorithm is different from that of Q-PASS because each recovery path in Q-CAST is dedicated to a single major path, and they are contention-free.

As an example, in Fig. 11, the major path disconnects at  $AC$  and  $EB$ . Nodes  $F, G$ , and  $H$  swap along the recovery path no matter the recovery path is used or not. As switch nodes,  $A$  and  $C$  recover the broken edge  $AC$  by the recovery path  $AFC$ . Both  $D$  and  $E$  know the two recovery paths covering  $EB$ , namely  $DGHB$  and  $EIB$ . The shorter one  $EIB$  is used.  $D$  still swaps qubits on the major path and  $E$  switches to the recovery path.

As another example, in Fig. 12, the major path disconnects at  $CD$  and  $EB$ . Recovery paths  $AFE$  and  $DGHB$  are taken, because the xor of the major path  $ACDEB$  and two recovery loops  $AFEDCA$  and  $DGHBED$  connects  $A$  and  $B$ . Note the edge  $DE$  appears 3 times in the xor and is used “reversely” on the final path.

#### Finding the optimal path selection for Q-CAST.

The contention-free path selection problem is hard by its nature: 1) due to the resource constraints (qubits/channels), path selection depends highly on the link states and hence the search space is much more than the classical algorithms which only depends on the fixed network topology; 2)  $E_t$  is non-linear, which invalids many existing routing algorithms based on the linear additivity property of the routing metric and thus degrades the efficiency of classical algorithms.

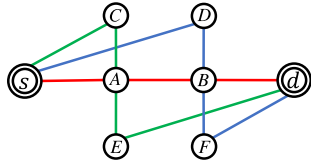


Fig. 13. Counterexample for Q-CAST and max-flow.

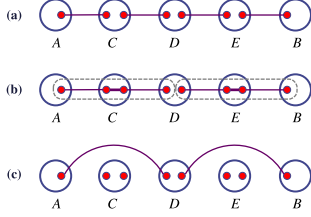


Fig. 14. Logarithmic-time swapping schedule in P4.

Due to the enormous search space, we failed to find the optimal qubit-channel allocation via brute-force even in a small 10-node network. Suppose  $|V| = 10$ , every node has 15 qubits and 6 edges, and each edge is composed of 5 quantum channels. In the brute-force searching, we do not assume the P2 and P4 are carried out based on ‘paths’, but just try all possible assignments of qubits to channels, perform the swapping, calculate the  $E_t$  between the given S-D pair, and record the highest result. For any S-D pair, the search space is  $\sim 2.3 \cdot 10^{36}$  for P2.<sup>3</sup> Even worse, the entanglement swapping in P4 depends on local states, which is prohibitively hard to enumerate all possible swapping combinations. We further show its hardness in two examples.

**Example 1.** Despite its good performance (shown in Section VI), we prove G-EDA is not always the optimal. An example graph is shown in Fig. 13.<sup>4</sup> Suppose all the edges have width 1, all channels have creation rate  $p = 0.99$ , the swapping success rate  $q = 1$ , and all nodes have qubit capacity 2. Then the optimal contention-free paths are the blue path plus the green path. But the G-EDA will only select the red path and stop because  $A$  and  $B$  run out of qubits. The root cause of the sub-optimality of G-EDA is it falls in a local minimum and fails to give the max-flow [68].

**Example 2.** Though the classical max-flow algorithm gives better solution when  $p$  is high, it performs worse than G-EDA when the channels are less stable. Consider the same topology in Fig. 13 with changed parameters. Suppose all blue and green edges have width 1, red edges have width 2, all channels have creation rate  $p = 0.6$ , the swapping success rate  $q = 0.95$ , all nodes have 4 qubits each. Referring Fig. 8, one (2, 3)-path is better than three (1, 4)-paths. Hence, the optimal solution is the red path with  $W = 2$ , which is also found via G-EDA. The max-flow algorithm, however, finds the sub-optimal solution: three paths with  $W = 1$ : the green, blue, and red path. Due to the limited qubit capacity on  $A$  and  $B$ , the other edges on the red path are not taken by max-flow.

<sup>3</sup>This number is got via a recursive algorithm instead of mathematical derivation. Consider the number of unique combinations of 15 indistinguishable balls put into 6 different buckets, each with capacity 5.

<sup>4</sup>Red path:  $(s, A, B, d)$ . Green path:  $(s, C, A, E, d)$ . Blue path:  $(s, D, B, F, d)$ .

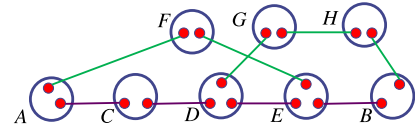


Fig. 15. Logarithmic swapping for Q-CAST.

TABLE I  
TIME AND SPACE COSTS

Algorithm	Time cost	Space cost
EXT	$O(hW)$	$O(W)$
Q-PASS P2	$O(mK_m(h_m + \log(mK_m)))$	$O(mK_m h_m + n)$
EDA	$O(n \log n +  E (h_m W_m))$	$O(n)$

#### D. Logarithmic-Time Swapping Schedule in P4

As discussed in Section III-D, each node locally determines which qubit pairs are swapped, and the swapping is transitive so that a path of neighbor entanglements can be swapped into a single remote entanglement. However, the nodes along the same path cannot swap in parallel, instead, as Fig. 14 shows, we can build the target remote entanglement  $AB$  by divide and conquer: build two sub-paths  $ACD$  and  $DEB$ , and then swap at  $D$ . Each sub-path is divided again, until each part only has no more than 3 nodes [69]. Q-PASS, Q-CAST, and any other routing algorithms are all compatible to this swapping schedule. There is, however, a requirement: all nodes along the path know their positions in the whole schedule, so building any sub-path does not involve a global synchronization.

For Q-CAST, the logarithmic schedule can be obtained via a consensus among all nodes on how the major path is segmented. For example in Fig. 15, the major path is  $ACDEB$ , and the two recovery paths are  $AFE$  and  $DGHB$ . The whole graph is segmented as five disjoint paths:  $ACD$ ,  $DE$ ,  $EB$ ,  $AFE$ , and  $DGHB$ . The division is done such that each path is a line and do not fork. In another word, any switch node is an end node at any path containing it. The P4 works as two sub-phases: first, build all disjoint paths by logarithmic-time swapping in parallel; then the final logarithmic-time swapping among source, destination, and all switch nodes.

#### E. Bounding the Search Space in Practice

To avoid unbound computation and space cost in P2, we set the maximum number of multipath  $K_m = 200$ . We set the maximum path hopcount according to the network itself. For any input  $G$ , 100 S-D pairs are randomly selected, and then multipath routing is performed via G-EDA between each S-D pair. The largest hopcount of selected paths whose  $E_t > 1$  is the maximum hopcount  $h_m$  of all selected paths.

### V. TIME AND SPACE COSTS

We denote the number of nodes as  $n$ , the number of S-D pairs as  $m$ , and the maximum width of paths as  $W_m$ , which is determined by node capacities and edge widths. The time and space costs of the core algorithms are shown in Table I.



## VI. PERFORMANCE EVALUATION

### A. Simulator Implementation

We implement the proposed network models and algorithms on a custom-built time-based simulator, with additional supports for the topology generation, statistics, and network visualization. We do not use packet-based simulation because quantum networks do not use packet switching. The source code repository of the simulator can be found on this link [35].

We do not assume any specific topology and randomly generate quantum networks for simulations. We set the area  $A$  holding quantum networks is a 100K units by 100K units square, each unit may be considered as 1km. The network generation algorithm requires three input parameters: the number of nodes  $n$ , the average number of neighbors  $E_d$ , and the average success rate of all channels  $E_p$ . Nodes are randomly placed and the distance of any two nodes is at least  $\leq 50/\sqrt{n}$  units. The edges are generated by the Waxman model [70] that was used for Internet topologies [71].

After the topology generation, a binary search on the model parameter  $\alpha$  is further carried out to make the average channel success rate to be  $E_p \pm 0.01$ . The number of qubits  $Q$  for each node is independently uniformly picked from 10 to 14. The edge width  $W$  is independently uniformly generated from 3 to 7, for each edge. We pick the range for  $Q$  and  $W$  based on our conjecture of a well-functioning quantum network. Our designs should work on wider ranges, which we cannot cover due to enormous possibilities.

### B. Methodology

We evaluate the throughput, scalability, and fairness of the proposed entanglement routing algorithms. To gain insight into the performance metrics and to provide a reference for future research, we show more simulation statistics: the resource efficiency towards high throughput, the contribution of recovery paths for both algorithms. Each data shown in the section is the average from 10 different network topologies.

We let the number of nodes  $n$  vary in set  $\{50, 100, 200, 400, 800\}$ , average channel success rate  $E_p$  vary in  $\{0.6, 0.3, 0.1\}$ , internal link success rate  $q$  vary in  $\{0.8, 0.9, 1.0\}$ , link state range  $k$  vary in  $\{0, 3, 6, \infty\}$ , average degree  $E_d$  vary in  $\{3, 4, 6\}$ , and the number of S-D pairs  $m$  vary from 1 to 10. To control variable, we show the results under the **reference setting**  $n = 100, E_p = 0.6, q = 0.9, k = 3, E_d = 6, m = 10$ , unless explicitly changed to observe the data trend. For each setting of  $(n, E_p, q, k, E_d, m)$ , 10 random networks are generated, and we simulate 1000 independent time slots on each of the networks.

We compare Q-PASS and Q-CAST with two existing routing algorithms that have been used in quantum network studies: single-link multipath routing (SLMP) [29] (a circuit-switching style protocol) and greedy routing [30] (a distributed protocol).

### C. Evaluation Results

**Throughput.** Figures 16 to 18 show the CDF of throughputs for Q-PASS, Q-CAST, Greedy, and SLMP, under the

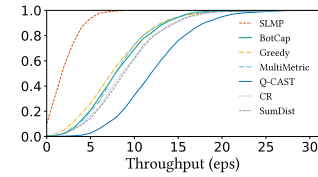


Fig. 16. CDF of throughput under the reference setting.

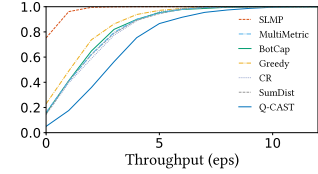


Fig. 17. CDF of throughput,  $E_p = 0.3$ .

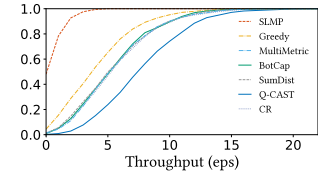


Fig. 18. CDF of throughput,  $n = 400$ .

reference setting. The throughput results are calculated in terms of ebits per time slot (eps). The BotCap, CR, and SumDist are the routing metrics for the Q-PASS, and they are shown separately for better comparison. Despite the multipath routing, SLMP shows the lowest throughput because of the unreliability of a single channel/link. It fails to deliver any ebits in  $>10$  percent of the time slots, and for 90 percent of the time slots, the total throughput between 10 S-D pairs are less than 5. The Greedy enjoys a high throughput, and for more than 90 percent of the time, it delivers more than 15 ebits for 10 random S-D pairs. For Q-PASS, all the three metrics of it exhibit similar throughput, and the CR metric gives the highest throughput among all metrics, which delivers about 2 eps more than the Greedy. Q-CAST shows great advantages over all other algorithms and outperforms the CR about 5 eps. Q-CAST is also the most reliable because it seldom delivers less than 5 eps. Since CR is slightly better than other metrics, we use CR to represent Q-PASS in the following results.

**Vary link state range.** In P3, each node shares its link states with its  $k$ -hop neighbors, and hence,  $k$  influences the path recovery performance. Fig. 19 shows the average throughput on different  $k$ . The Greedy algorithm does not rely on  $k$  and is shown for reference.  $k$  contributes little to the overall performance because most path failures are just one hop  $v_i-v_{i+1}$ , which can be recovered by  $v_i$  and  $v_{i+1}$  with their own link states.  $k = 3$  is sufficient for Q-CAST, and larger  $k$  slightly degrades the throughput because longer and more unreliable recovery paths may be selected. This would occupy the routing resource which could have been allocated to other shorter and more reliable recovery paths.

**Vary link success rates.** Figures 20 and 21 show the average throughput of Q-PASS, Q-CAST, Greedy, and SLMP on different quantum device abilities by varying the average channel success rate and swapping (internal link) success rate. When the channel success rate  $p$  or the swapping success rate  $q$  is small, the overall throughput will be degraded. A robust

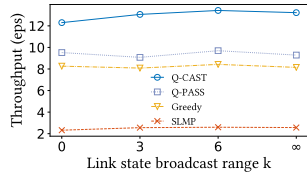


Fig. 19. Throughput vs. LS sharing ranges.

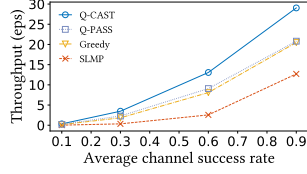


Fig. 20. Throughput vs. channel success rates.

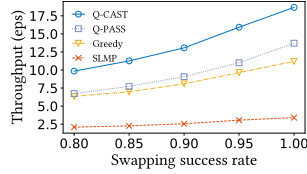


Fig. 21. Throughput vs. swapping success rates.

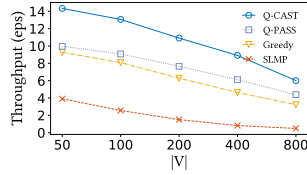


Fig. 22. Throughput vs. network sizes.

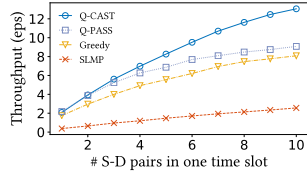


Fig. 23. Throughput vs. # of S-D pairs.

routing algorithm should still perform well on low ability networks. From the figures, the swapping success rate also has big impact on the average throughput, because the link failure in the P2 can be mitigated by the recovery algorithms in P4, but there is no circumvention for swapping errors. And the Q-CAST performs the best among the four algorithms.

**Scalability.** We evaluate the scalability of routing algorithms on two dimensions: the size of the network  $n$  and the number of concurrent S-D pairs  $m$ . A larger network means the average distance of S-D pairs is longer; and more concurrent S-D pairs in one time slot introduce higher level of resource contention. Figures 22 and 23 show the average throughput on the two dimensions. All algorithms exhibit a logarithmic throughput decrease with the number of nodes in the network. Q-CAST outperforms others on all network sizes, and the throughput of Q-CAST is as high as 7.5eps when the network contains 800 nodes. The reason of lower throughput in larger networks is because the average path length is longer for the S-D pairs. Longer paths are more likely to fail in quantum networks. Besides, the throughput of all algorithms grow sub-linearly with the number of S-D pairs, due to resource contentions. Q-CAST outperforms others on most settings, and

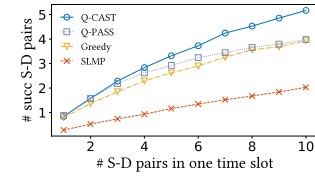


Fig. 24. # of successful concurrent S-D pairs.

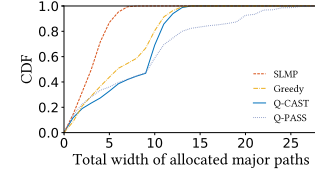


Fig. 25. CDF of # of major paths.

the advantage of Q-CAST over other algorithms grows rapidly with the number of S-D pairs. It is because Q-CAST actively resolves the resource contentions for the S-D pairs.

**Fairness.** Though we aim to maximize the throughput in the current designs, the fairness among the S-D pairs is evaluated. Fig. 24 shows the average number of successful S-D pairs under different numbers of concurrent requests. For a time slot, an S-D pair is successful (*epair*) when they establish at least one ebits after P4. Q-CAST outperforms others and all algorithms grow sub-linearly. Fig. 25 shows the CDF of the number of paths allocated to every S-D pair. A  $W$ -path is counted as  $W$  separate paths. As a baseline requirement, any S-D pair should be allocated at least one major path, which is fulfilled by all algorithms. The SLMP is the fairest. The Q-CAST has a turning point on the CDF figure, which means 40 percent of S-D pairs are allocated less than 9 paths, and the other pairs are allocated 10 to 14 paths, which is very fair. The Q-PASS is the most biased algorithm.

**Recovery paths.** We evaluate the contribution of recovery paths to the overall throughput for both Q-PASS and Q-CAST, by comparing their throughput with that of their recovery path-free versions Q-PASS\R and Q-CAST\R. The results are shown in Fig. 26. The recovery paths contribute about 0.5eps to Q-PASS and 1eps to Q-CAST. We further show the average number of occupied channels in one time slot for Q-CAST, Q-CAST\R, and Q-PASS in Fig. 27, where the  $x$ -axis shows the throughput of each case. Q-PASS is not shown in this figure because it takes way more channels in the recovery paths and the results are not in this range of  $y$ -axis. Q-PASS\R takes times fewer channels compared with Q-PASS, and Q-CAST\R saves 25% channels from the 400 channels taken by Q-CAST.

As the recovery paths are contention-free for Q-CAST, more interesting statistics are collected on Q-CAST recovery paths. The CDF of the width of recovery paths is shown in Fig. 30. The recovery paths can be wider when the number of S-D pairs is small, because of the low resource contention between S-D pairs. For most cases, the widths of recovery paths for a single S-D pair are larger than those of the 10 concurrent S-D pairs by 2. Besides, the CDF of the total number of recovery paths of a single major path is shown in Fig. 31. In larger networks, the major paths are longer, and more recovery paths can be found.

**Resource efficiency.** The efficiency on qubit/channel should be evaluated because high resource occupation incurs

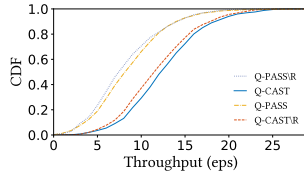


Fig. 26. Contribution of recovery paths.

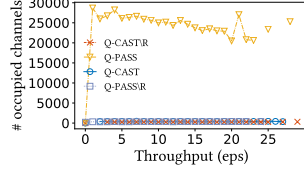


Fig. 27. Overhead of recovery paths.

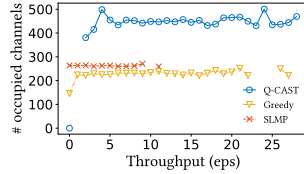


Fig. 28. Channel occupation for different throughput.

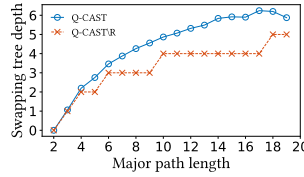


Fig. 29. Logarithmic swapping for Q-CAST.

huge cost on hardware, software, and the classical network communication. For each time slot, we record the ebits delivered (throughput) and the number of occupied channels, and we show the average number of occupied channels on different throughput in Fig. 28. Greedy occupies the least channels to deliver a certain number of throughput, while Q-CAST consistently delivers higher throughput. Another interesting observation is that for different throughput results, the number of occupied channels is similar for each algorithm.

**Logarithmic-time swapping.** We evaluate the efficiency of logarithmic-time swapping of Q-CAST P4, in terms of the number of non-parallelized swappings to deliver a remote entanglement. For each successful epair, we record the length of the major path and the depth of the swapping tree, and we show the average swapping tree depth on different throughput in Fig. 29. We can see a clear logarithmic growth of the swapping steps with the path length, and with recovery paths, the swapping tree depth is slightly higher, but still within a reasonable range. For short paths, the recovery paths do not incur extra swapping depth.

**Summary of evaluations.** Q-CAST exhibits much higher throughput, robustness, and scalability than other routing algorithms. Q-PASS also shows good throughput and the metric CR provides the highest throughput for Q-PASS. If the minimum resource utilization is a concern for some quantum networks, recovery paths for both algorithms can be disabled for better efficiency. Q-CAST\R is a good balance between throughput and resource efficiency. Considering logarithmic-time swapping, P4 swapping time can be reduced for Q-CAST, even if allowing recovery paths.

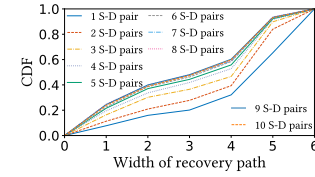


Fig. 30. CDF of the width of recovery paths.

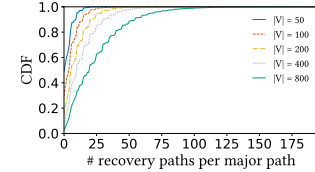


Fig. 31. CDF of # of recovery paths on a single major path.

## VII. DISCUSSION

**Better fairness.** The algorithms proposed in this paper aim to maximize throughput, and each time slot is considered totally separately. A simple extension, however, is available to both Q-PASS and Q-CAST to provide better fairness while maintaining high throughput. For any S-D pair that has failed to share an ebit in a slot  $T$ , the pair and the failing streak  $\langle (s, d), 1 \rangle$  are broadcast to all nodes in P1 in the slot  $T + 1$ . The routing metric of all paths connecting this S-D pair is multiplied with a factor such as 1.1, which means their paths are slightly over-evaluated, and thus are more likely to be selected. If the pair still fails, the failing streak increases to a higher factor such as 2, and the related routing metric is multiplied with  $1.1^2$  in  $T + 2$ . Eventually, this pair will succeed.

**Prioritized routing.** Both Q-PASS and Q-CAST are extendable to support simple prioritized routing. Suppose S-D pairs are in different priority classes, identified by the number  $1, 2, \dots, 10$ , and the priority is ‘hard’ – a single S-D pair in priority class  $c$  is far more valuable than all S-D pairs in priority class  $c - 1$  and lower. In P2 of Q-PASS and Q-CAST, the offline paths (only Q-PASS) and online paths (both algorithms) of the highest priority S-D pair are selected until no more path is available. More paths are then selected in the residual graph for S-D pairs in lower priority classes. The P4 of Q-CAST is not modified because the selected paths have no contention. In P4 of Q-PASS, the paths of the highest priority S-D pair are recovered first.

**Continuous model.** The time slot model used by this work is called the on-demand model in [29] and [30] because entanglements are built and swapped in a single time slot, after knowing the demand of the network. The continuous model is proposed in [30], where the linkwise entanglements are continuously created in the background, and preserved in the quantum memory before actually used in a long-distance entanglement. After knowing the target S-D pair, the path discovery algorithm runs hop-by-hop to find a viable path, and do the swapping along the found path. However, compared to the proposed time slot model, this approach is more prone to fidelity degradation with time. 1) The linkwise entanglements must be continuously preserved (and decaying with time) in the quantum memory during the path discovery phase, which is carried out via classical communication and computation hop-by-hop, drawing it impractical for a large scale network as



the communication and computation costs per hop are both at millisecond level, and the entanglements can only be preserved in about 10 milliseconds. 2) As the entanglement creations are not synchronized, in the path discovery phase, the selected linkwise entanglements differ a lot in their preservation time, as hence differs a lot in their fidelity. After swapping along the selected path, the final fidelity is bounded by the lowest fidelity of the linkwise entanglements used in that path. That raises a hard trade-off between the number of qualified linkwise entanglements and the end to end entanglement quality. As a contrast, the time slot model used by this work is more carefully designed to align the linkwise entanglement creation and to scale with the network. The path is determined in P2 before any quantum entanglement is created. P3 is the only classical communication and computation during the preservation of linkwise entanglements, and it only allows each node communicating with a constant number of physical neighbors before making the swapping decision. With is novel design, the algorithms under time slot model naturally enjoys high fidelity entanglements and scales perfectly with the network size. Besides, the continuous model proposed in [30] only works for a single S-D pair. If concurrent S-D pairs exist, routing may easily fail because some links are occupied by the holding paths, unless there is global link state broadcast that causes high communication cost and long latency. We consider the continuous model as another direction which requires a co-design of the network topology and the routing algorithm. Designing routing algorithms under the continuous model to minimize the latency in routing decisions and decently handling the fidelity degradation under unaligned linkwise entanglements are two interesting research directions.

We also investigate devices whose entanglement life time might span across multiple slots. The results are published [72].

## VIII. CONCLUSION AND FUTURE WORK

This work presents a new entanglement routing model of quantum networks that reflects the differences compared to classical networks and new entanglement routing algorithms that utilize the unique properties of quantum networks. The proposed algorithm Q-CAST increases network throughput by a big margin compared to other methods. We expect more future research will be conducted on the entanglement routing problem and could contribute to the eventual success of quantum networks.

## ACKNOWLEDGMENT

The authors would like to thank Liang Jiang, Rui Li, Peter Young, Hongqiang Harry Liu, Shahrooz Pouryousef, Ruilin Zhou, and the anonymous reviewers for their suggestions and comments.

## REFERENCES

- [1] S. Shi and C. Qian, "Concurrent entanglement routing for quantum networks: Model and designs," in *Proc. ACM SIGCOMM*, 2020, pp. 62–75.
- [2] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," in *Proc. Int. Conf. Comput., Syst. Signal Process.*, 1984, pp. 175–179.
- [3] A. K. Ekert, "Quantum cryptography based on Bell's theorem," *Phys. Rev. Lett.*, vol. 67, p. 661, Aug. 1991.
- [4] J.-W. Pan, D. Bouwmeester, H. Weinfurter, and A. Zeilinger, "Experimental entanglement swapping: Entangling photons that never interacted," *Phys. Rev. Lett.*, vol. 80, no. 18, pp. 3891–3894, May 1998.
- [5] M. Sasaki et al., "Field test of quantum key distribution in the Tokyo QKD network," *Opt. Exp.*, vol. 19, no. 11, pp. 10387–10409, 2012.
- [6] C. Elliott, "Building the quantum network," *New J. Phys.*, vol. 4, p. 46, Jan. 2002.
- [7] C. Elliott, D. Pearson, and G. Troxel, "Quantum cryptography in practice," in *Proc. ACM SIGCOMM*. New York, NY, USA: Association for Computing Machinery, 2003, pp. 227–238.
- [8] M. Peev, "The SECOQC quantum key distribution network in Vienna," *New J. Phys.*, vol. 11, Jul. 2009, Art. no. 075001.
- [9] J. Yin et al., "Satellite-based entanglement distribution over 1200 kilometers," *Science*, vol. 356, no. 6343, pp. 1140–1144, Jun. 2017.
- [10] H. J. Kimble, "The quantum Internet," *Nature*, vol. 453, pp. 1023–1030, Jun. 2008.
- [11] S. Wehner, D. Elkouss, and R. Hanson, "Quantum Internet: A vision for the road ahead," *Science*, vol. 362, no. 6412, Oct. 2018, Art. no. eaam9288.
- [12] M. Caleffi, A. S. Cacciapuoti, and G. Bianchi, "Quantum Internet: From communication to distributed computing!" in *Proc. NANOCOM*, Sep. 2018, pp. 1–4.
- [13] A. S. Cacciapuoti, M. Caleffi, F. Tafuri, F. S. Cataliotti, S. Gherardini, and G. Bianchi, "Quantum Internet: Networking challenges in distributed quantum computing," *IEEE Netw.*, vol. 34, no. 1, pp. 137–143, Jan. 2020.
- [14] L. Childress and R. Hanson, "Diamond NV centers for quantum computing and quantum networks," *MRS Bull.*, vol. 38, no. 2, pp. 134–138, Feb. 2013.
- [15] A. Dahlberg et al., "A link layer protocol for quantum networks," in *Proc. ACM SIGCOMM*, Aug. 2019, pp. 159–173.
- [16] S. Pirandola et al., "Advances in quantum cryptography," 2019, *arXiv:1906.01645*.
- [17] P. Komar et al., "A quantum network of clocks," *Nature Phys.*, vol. 10, pp. 582–587, Jun. 2014.
- [18] A. Broadbent, J. Fitzsimons, and E. Kashefi, "Universal blind quantum computation," in *Proc. 50th Annu. IEEE Symp. Found. Comput. Sci.*, Oct. 2009, pp. 517–526.
- [19] V. S. Denchev and G. Pandurangan, "Distributed quantum computing: A new frontier in distributed systems or science fiction?" *ACM SIGACT News*, vol. 39, no. 3, pp. 77–95, Sep. 2008.
- [20] J. L. Park, "The concept of transition in quantum mechanics," *Found. Phys.*, vol. 1, no. 1, pp. 23–33, 1970.
- [21] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.
- [22] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 26, no. 1, pp. 96–99, Jan. 1983.
- [23] M. Riebe et al., "Deterministic quantum teleportation with atoms," *Nature*, vol. 429, no. 6993, pp. 734–737, Jun. 2004.
- [24] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annu. Symp. Found. Comput. Sci.*, 1994, pp. 124–134.
- [25] S. Pirandola, R. García-Patrón, S. L. Braunstein, and S. Lloyd, "Direct and reverse secret-key capacities of a quantum channel," *Phys. Rev. Lett.*, vol. 102, no. 5, Feb. 2009, Art. no. 050503.
- [26] S. Pirandola, R. Laurenza, C. Ottaviani, and L. Banchi, "Fundamental limits of repeaterless quantum communications," *Nature Commun.*, vol. 8, no. 1, p. 15043, Apr. 2017.
- [27] S. Olmschenk, D. N. Matsukevich, P. Maunz, D. Hayes, L.-M. Duan, and C. Monroe, "Quantum teleportation between distant matter qubits," *Science*, vol. 323, no. 5913, pp. 486–489, Jan. 2009.
- [28] H. Bernien et al., "Heralded entanglement between solid-state qubits separated by three metres," *Nature*, vol. 497, no. 7447, pp. 86–90, May 2013.
- [29] M. Pant et al., "Routing entanglement in the quantum Internet," *npj Quantum Inf.*, vol. 5, no. 1, p. 25, Mar. 2019.
- [30] K. Chakraborty, F. Rozpedek, A. Dahlberg, and S. Wehner, "Distributed routing in a quantum internet," 2019, *arXiv:1907.11630*.
- [31] X. Hong, K. Xu, and M. Gerla, "Scalable routing protocols for mobile ad hoc networks," *IEEE Netw.*, vol. 16, no. 4, pp. 11–21, Jul. 2002.
- [32] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, "Hedera: Dynamic flow scheduling for data center networks," in *Proc. USENIX NSDI*, 2010, pp. 1–15.

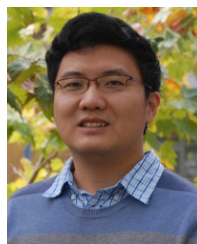
- [33] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey, "Jellyfish: Networking data centers randomly," in *Proc. USENIX NSDI*, 2012, pp. 1–14.
- [34] Y. Yu and C. Qian, "Space shuffle: A scalable, flexible, and high-bandwidth data center network," in *Proc. IEEE 22nd Int. Conf. Netw. Protocols*, Oct. 2014, pp. 13–24.
- [35] 2019. *Source Code of the Quantum Routing Simulations*. [Online]. Available: <https://github.com/QianLabUCSC/QuantumRouting>
- [36] R. V. Meter and J. Touch, "Designing quantum repeater networks," *IEEE Commun. Mag.*, vol. 51, no. 8, pp. 64–71, Aug. 2013.
- [37] R. Van Meter, T. Satoh, T. D. Ladd, W. J. Munro, and K. Nemoto, "Path selection for quantum repeater networks," *Netw. Sci.*, vol. 3, nos. 1–4, pp. 82–95, Dec. 2013.
- [38] S. Pirandola, "End-to-end capacities of a quantum communication network," *Commun. Phys.*, vol. 2, no. 1, p. 51, May 2019.
- [39] E. Schoute, L. Mancinska, T. Islam, I. Kerenidis, and S. Wehner, "Shortcuts to quantum network routing," 2016, *arXiv:1610.05238*.
- [40] S. Das, S. Khatri, and J. P. Dowling, "Robust quantum network architectures and topologies for entanglement distribution," *Phys. Rev. A, Gen. Phys.*, vol. 97, no. 1, Jan. 2018, Art. no. 012335.
- [41] M. Caleffi, "Optimal routing for quantum networks," *IEEE Access*, vol. 5, pp. 22299–22312, 2017.
- [42] G. Vardoyan, S. Guha, P. Nain, and D. Towsley, "On the stochastic analysis of a quantum entanglement switch," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 47, no. 2, pp. 27–29, Dec. 2019.
- [43] Y. Zhao and C. Qiao, "Redundant entanglement provisioning and selection for throughput maximization in quantum networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2021, pp. 1–10.
- [44] L. Yang, Y. Zhao, H. Xu, and C. Qiao, "Online entanglement routing in quantum networks," in *Proc. IEEE/ACM 30th Int. Symp. Quality Service (IWQoS)*, Jun. 2022, pp. 1–10.
- [45] Y. Zeng, J. Zhang, J. Liu, Z. Liu, and Y. Yang, "Multi-entanglement routing design over quantum networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2022, pp. 510–519.
- [46] Y. Zeng, J. Zhang, J. Liu, Z. Liu, and Y. Yang, "Entanglement routing design over quantum networks," *IEEE/ACM Trans. Netw.*, early access, pp. 1–16, Jun. 2023.
- [47] 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.
- [48] M. Liu, J. Allcock, K. Cai, S. Zhang, and J. C. S. Lui, "Quantum networks with multiple service providers: Transport layer protocols and research opportunities," *IEEE Netw.*, vol. 36, no. 5, pp. 56–62, Sep. 2022.
- [49] A. Farahbakhsh and C. Feng, "Opportunistic routing in quantum networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2022, pp. 490–499.
- [50] D. L. Moehring et al., "Entanglement of single-atom quantum bits at a distance," *Nature*, vol. 449, no. 7158, pp. 68–71, Sep. 2007.
- [51] A. S. Cacciapuoti, M. Caleffi, R. van Meter, and L. Hanzo, "When entanglement meets classical communications: Quantum teleportation for the quantum Internet," *IEEE Trans. Commun.*, vol. 68, no. 6, pp. 3808–3833, Jun. 2020.
- [52] S. Pirandola and S. L. Braunstein, "Physics: Unite to build a quantum Internet," *Nature*, vol. 532, no. 7598, pp. 169–171, Apr. 2016.
- [53] M. Takeoka, S. Guha, and M. M. Wilde, "Fundamental rate-loss tradeoff for optical quantum key distribution," *Nature Commun.*, vol. 5, no. 1, p. 5235, Oct. 2014.
- [54] Y. Lee, E. Bersin, A. Dahlberg, S. Wehner, and D. Englund, "A quantum router architecture for high-fidelity entanglement flows in quantum networks," 2020, *arXiv:2005.01852*.
- [55] D. Luong, L. Jiang, J. Kim, and N. Lütkenhaus, "Overcoming lossy channel bounds using a single quantum repeater node," *Appl. Phys. B, Lasers Opt.*, vol. 122, no. 4, p. 96, Apr. 2016.
- [56] P. C. Humphreys et al., "Deterministic delivery of remote entanglement on a quantum network," *Nature*, vol. 558, no. 7709, pp. 268–273, Jun. 2018.
- [57] J. Moy, *OSPF Version 2*, document RFC 2328, Internet Engineering Task Force, Fremont, CA, USA, 1998.
- [58] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, Mar. 2002.
- [59] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*. Norwell, MA, USA: Kluwer Academic, 1996, pp. 153–181.
- [60] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. IEEE Workshop Mobile Comput. Syst. Appl.*, Feb. 1999, pp. 90–100.
- [61] B. Karp and H.-T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. 6th Annu. Int. Conf. Mobile Comput. Netw.*, 2000, pp. 243–254.
- [62] S. S. Lam and C. Qian, "Geographic routing in  $d$ -dimensional spaces with guaranteed delivery and low stretch," in *Proc. ACM SIGMETRICS*, 2011, pp. 217–228.
- [63] C. Qian and S. S. Lam, "Greedy distance vector routing," in *Proc. 31st Int. Conf. Distrib. Comput. Syst.*, Jun. 2011, pp. 857–868.
- [64] J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts, "On-line load balancing with applications to machine scheduling and virtual circuit routing," in *Proc. 25th Annu. ACM Symp. Theory Comput. (STOC)*, 1993, pp. 623–631.
- [65] R. Rooholamini, V. Cherkassky, and M. Garver, "Finding the right ATM switch for the market," *IEEE Comput.*, vol. 27, no. 4, pp. 16–28, Apr. 1994.
- [66] R. Cole et al., "Randomized protocols for low-congestion circuit routing in multistage interconnection networks," in *Proc. 13th Annu. ACM Symp. Theory Comput. (STOC)*, 1998, pp. 378–388.
- [67] J. Y. Yen, "Finding the  $K$  shortest loopless paths in a network," *Manage. Sci.*, vol. 17, no. 11, pp. 712–716, Jul. 1971.
- [68] T. Harris and F. Ross, "Fundamentals of a method for evaluating rail net capacities," Rand Corp., Santa Monica, CA, USA, Tech. Rep. RM-1573, 1955.
- [69] 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.
- [70] B. M. Waxman, "Routing of multipoint connections," *IEEE J. Sel. Areas Commun.*, vol. 6, no. 9, pp. 1617–1622, Dec. 1988.
- [71] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: An approach to universal topology generation," in *Proc. Int. Workshop Model. Anal. Simul. Comput. Telecommun. Syst.*, 2001, pp. 346–353.
- [72] Y. Gan, X. Zhang, R. Zhou, Y. Liu, and C. Qian, "A routing framework for quantum entanglements with heterogeneous duration," in *Proc. IEEE Int. Conf. Quantum Comput. Eng. (QCE)*, Sep. 2023, pp. 1132–1142.



**Shouqian Shi** received the B.Sc. degree in physics and the B.E. degree in computer science and engineering from the University of Science and Technology of China in 2014 and the Ph.D. degree in computer engineering from the University of California, Santa Cruz, in 2021. He is currently with Google LLC. His research interests include computer networks, distributed systems, and other emerging computer systems, including quantum networks, cloud and edge computing, software-defined networks, network security and privacy, and network verification.



**Xiaoxue Zhang** (Graduate Student Member, IEEE) received the B.E. degree in computer science and engineering from the University of Science and Technology of China in 2019. She is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, University of California, Santa Cruz. Her research interests include computer networks, payment channel networks, and routing. She also focuses on the scalability and security problem of blockchain.



**Chen Qian** (Senior Member, IEEE) received the B.Sc. degree in computer science from Nanjing University in 2006, the M.Phil. degree in computer science from The Hong Kong University of Science and Technology in 2008, and the Ph.D. degree in computer science from The University of Texas at Austin in 2013. He is currently a Professor with the Department of Computer Science and Engineering, University of California, Santa Cruz. He has published more than 80 research papers in a number of top conferences and journals, including ACM SIGMETRICS, IEEE ICNP, IEEE ICDCS, IEEE INFOCOM, IEEE PerCom, ACM UBICOMP, ACM CCS, IEEE/ACM TRANSACTIONS ON NETWORKING, and IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS. His research interests include computer networking, quantum networks, data-center networks, cloud computing, the Internet of Things, and software-defined networks. He is a Senior Member of ACM. He received the NSF Career Award in 2018.