2024 International Conference on Quantum Communications, Networking, and Computing (QCNC) | 979-8-3503-6677-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/QCNC62729.2024.00052

Towards QoS-aware Quantum Networks

Ruilin Zhou, Yuhang Gan, Yi Liu, Katia Obraczka, and Chen Qian
University of California, Santa Cruz

***Abstract*—Quality of Service (QoS) has been studied in classic networks to support various applications. Due to the unique properties of quantum networks, like entanglement and superposition, the definition and requirements of QoS in quantum networks differ significantly from classical networks. Limited work has been done from the application side to investigate these differences. Moreover, quantum applications exhibit distinct service requirements compared to traditional network applications, such as the need for quantum state preservation and fidelity requirements. Therefore, this work aims to analyze and model various quantum applications to identify the fundamental services required in quantum networks to provide application-specific QoS. We further propose a novel QoS routing framework, called Q2R, for quantum networks to demonstrate a possible method to achieve QoS via routing. Q2R computes feasible paths for each request and meets the fidelity and number of qubits requirements. With our framework, we can allocate resources more efficiently and our scheduling algorithm also considers fairness among different requests.**

I. Introduction

The fundamental goal of quantum networks is to generate end-to-end entanglement. These entangled pairs are used to support various quantum network applications. Entanglement enables quantum communication by providing strong correlations and facilitating challenging tasks in classical computation and communication. However, due to the nature of quantum physics, generating entanglement pairs is difficult: created qubits are fragile and cannot be maintained for long periods [1]. Most quantum network operations, like entanglement generation, entanglement swapping, and entanglement purification, are all probabilistic [1], causing the quality of operations to be uncertain. Different quantum network applications have different levels of services in terms of entanglement delivered. Given these distinct characteristics, future quantum networks require novel protocols that differ significantly from classical networks.

This work focuses on Quality of Services (QoS), a concept that has been well-studied in classic networks, in the context of quantum networks. We believe that providing QoS is an inevitable design consideration of practical quantum networks. We first analyze different major applications in quantum networks and identify key QoS metrics for these applications. By modeling these applications' behaviors and traffic patterns, we investigate what services are needed from quantum networks regarding delivered entanglement.

We further show a potential method to provide QoS in quantum networks. We study the entanglement routing problem, which aims to build a long-distance end-to-end entanglement to demonstrate the use of protocols and frameworks to support QoS services in quantum networks. End-to-end entanglement in quantum networks is generated through multiple hops of quantum repeaters using entanglement pair generation, entanglement swapping, and entanglement purification. The entanglement routing problem has drawn great attention recently, and many methods of entanglement routing have been proposed [2]–[4]. Most existing methods focus on the optimization goal of maximizing the routing throughput. However, we argue that maximizing throughput is insufficient to maximize user application satisfaction, which is the ultimate goal of building a quantum network. Similar to the quality of service (QoS) requirements in classic Internet, the applications of quantum networks also have requirements on the quality of the qubits they want to deliver. For example, quantum key distribution (QKD) requires a certain level of fidelity of the delivered qubits to generate secret keys, a fidelity of 81% is needed for a practical key generation rate [5]. From the application perspective, we argue that the metric *goodput*, which is defined as the number of qubits that are useful for applications in a time unit, will be more crucial than throughput for real applications. Our literature analysis also tells that 1) **Application-aware routing that considers multiple QoS metrics and purification jointly is a missing part of the literature on quantum networks.** 2) **QoS solutions in classical networks [6], [7] is different from quantum networks** in that, metrics in classical networks like delay or bandwidth are either concave or additive. However, metrics in quantum networks like goodput and fidelity are not additive nor linear which requires a more complex design in routing algorithms. To address the limitations of previous work, we proposed a new routing framework called application-aware QoS Quantum entanglement Routing (Q2R). Our contributions to this work are as follows:

- We analyze different applications in quantum networks and identify key QoS metrics. By modeling these applications' behaviors, we investigate what services are needed from quantum networks regarding delivered entanglements.
- We are the first to formulate the QoS-aware entanglement routing problem that takes latency, number of delivered entanglements, fidelity, and purification into consideration.
- We design a QoS routing framework, Q2R, including a QoS routing process and scheduling process that can meet heterogeneous QoS requirements of the requests from multiple users concurrently.
- Results of evaluations show that our approach can significantly improve goodput compared to previous work.

II. Background

A. Qubits and Quantum State

Qubit is the fundamental unit in quantum computing and quantum networks. Properties such as entanglement and super-

979-8-3503-6677-8/24/$31.00 ©2024 IEEE
DOI 10.1109/QCNC62729.2024.00052

position make qubits different from classical bits. One classical bit can have values of either 1 or 0, but a qubit can be in the superposition state of 1 and 0. Such a state can be written mathematically as follows: $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$, $|\Psi\rangle$ denotes a quantum state and both $|\alpha|$ and $|\beta|$ are complex number. Upon the measurement on a qubit, the quantum state of the qubit will become either state $|0\rangle$ or state $|1\rangle$ with probability $|\alpha|^2$ or $|\beta|^2$. An important feature of the quantum state is entanglement. An example of entanglement can be expressed as follows: $\frac{1}{\sqrt{2}}\left(|0\rangle_A|0\rangle_B + |1\rangle_A|1\rangle_B\right)$. The measurement resulting in x on qubit A will always result in another outcome state y on qubit B.

1) Entanglement Swapping: In quantum networks, entanglement swapping is crucial and can be used to entangle distant nodes. Essentially, entanglement swapping involves the transfer of entanglement between two quantum systems that do not share a direct entangled link, via an intermediary system that is entangled with both. As shown in Fig. 1, Alice and Bob each share one qubit entangled with one qubit in the middle repeater. Then, Bell-State-Measurement (BSM) is performed on the two qubits in the repeater node, eventually resulting in the entangled qubits in Bob and Alice. Qubits in distant nodes can be entangled through multiple hops of repeaters [4].

2) Fidelity: Fidelity represents the 'closeness' of two quantum states and can be considered entanglement quality. Most operations and states are imperfect, resulting in the real state having differences from the desired state, quantified by fidelity. The fidelity of entanglement can be affected by many factors, like initial fidelity when entanglement is generated and imperfect operations. While entanglement swapping can prolong the entanglement, it also makes fidelity decrease. Suppose two pairs of entangled states in Werner state [8] have fidelity F_1 and F_2, the fidelity F_{after} after swapping can be computed as:

$$F_{after} = F_1 F_2 + \frac{(1-F_1)(1-F_2)}{3} \qquad (1)$$

3) Entanglement Purification: Entanglement purification is a technique to overcome noise and loss in quantum channels. After entanglement generation or swapping, the generated states might be imperfect and cannot be used for quantum applications. Entanglement purification takes a collection of low-fidelity pairs of qubits and, through local operations and classical communication, produces a smaller number of high-fidelity pairs. As shown in Fig. 1, two entangled pairs with fidelity 0.90 are consumed to generate an entangled pair with fidelity 0.95. The well-known BBPSW protocol [9] is a symmetric purification protocol. It takes entangled pairs with similar fidelity F. The fidelity after purification can be computed as follows:

$$\frac{F^2 + \frac{1}{9}(1-F)^2}{F^2 + \frac{2}{3}F(1-F) + \frac{5}{9}(1-F)^2} \qquad (2)$$

III. QoS Analysis of Quantum Applications

We analyze and model various quantum network applications, abstracting their behaviors to generalize their Quality of Service (QoS) requirements. We also analyze the traffic patterns of these applications. This analysis addresses two primary questions: 1) What metrics are crucial for different quantum applications? 2) Based on these metrics, what are the specific requirements for entanglement delivery needed to meet these demands?

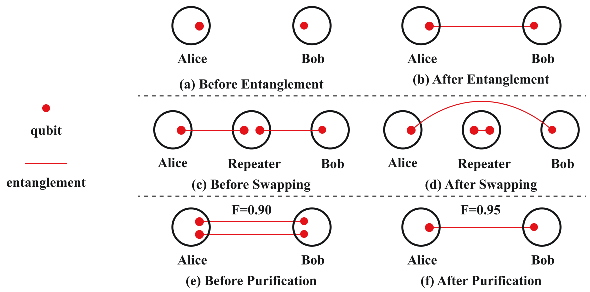

Fig. 1: Entanglement and entanglement swapping.

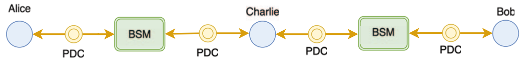

Fig. 2: Three nodes Entanglement Based QKD Chain

A. Quantum Key Distribution (QKD)

QKD, as a typical quantum application, has been well-studied in recent decades. The main idea is to encode a message onto photons and transmit photons to complete the process of negotiating the key between the two parties. A series of protocols have been proposed. They are mainly categorized into entanglement-based QKD protocols [10], [11] and non-entanglement-based QKD protocols.

For a QKD protocol, one important metric to quantify the performance is the key generation rate, which refers to the rate at which secure cryptographic key bits are generated and verified between two parties, and the key generation rate can be considered as one QoS metric in the quantum network setting. We give the relation of key generation rate with different parameters in different protocols. For one-way communication, the secret key generated given by Shor and Preskill's proof is:

$$R_{\text{sec}} = R_{\text{sift}}\left[1 - \kappa H_2(\text{QBER}) - H_2(\text{QBER})\right] \qquad (3)$$

R_{sift} is the sifted key rate that refers to the number of successful detection events per second where two adjacent nodes in a QKD chain (e.g., Alice and Charlie in Fig. 2) independently and correctly choose the same measurement basis, leading to compatible and potentially secure key bits [12]. R_{sift} measures the throughput of potentially usable raw key bits for both parties. The second term for calculating R_{sec} describes the effect of privacy amplification where $H_2(X)$ is the binary Shannon entropy function, which is less relevant in the entanglement distribution of quantum networks.

B. Distributed Quantum Computing

Distributed Quantum Computing (DQC) aims to overcome the scalability limitations of current quantum hardware by distributing a single quantum circuit across multiple Quantum Processing Units (QPUs) [13]. The initial phase of DQC involves qubit allocation, which assigns physical qubits to each QPU [14]. This aspect, however, is beyond the scope of this work. Consider the example shown in Fig. 3(a), where a simple quantum circuit is naively distributed over three QPUs. The

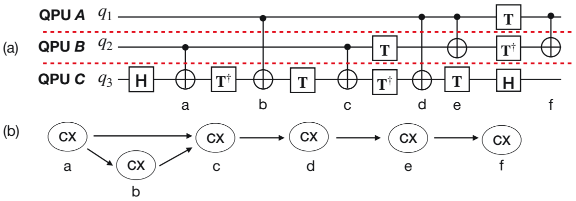

Fig. 3: (a) Example DQC Circuit (b) Remote Gate DAG

execution of this circuit must respect the dependency of the original design, meaning a two-qubit gate on qubits q_0 and q_1 can only be executed after all preceding gates on these qubits are completed. We use a Directed Acyclic Graph (DAG) to represent the computation process. This DAG includes only inter-QPU gates and is depicted in Fig. 3(b). Each node within this DAG represents a remote gate operation, and each edge indicates the dependency between these gates.

One critical QoS metric in DQC is the *Job Completion Time* (JCT), which significantly increases due to the time-intensive nature of remote operations and the resultant quantum state decoherence. Minimizing JCT not only enhances DQC performance but also reduces user wait times and improves network throughput. To minimize JCT, it is imperative to ensure that high-fidelity EPR pairs are prepared ahead of each gate's start time. Otherwise, we will have to wait until the EPR pair is generated, which will result in the congestion of all the following remote gates. Besides JCT, one important QoS metric that evaluates the performance and reliability of executing one quantum circuit is fidelity. Previous work on single QPU quantum computing also defines a metric called Probability of Successful Trial (PST) [15] which heavily relies on fidelity. When we extend PST in DQC environments, the success rate is also highly related to fidelity – detailed formulation skipped due to page limit.

C. Quantum Distributed Systems

Examples of quantum distributed systems include Quantum Byzantine Agreement [16] and Quantum Secret Sharing [17]. These applications require the engagement of multiple parties and will require the preparation of a multi-party entanglement state distributed to each party. For example in Quantum Secret Sharing, Alice needs to prepare a three-particle GHZ state $|\Psi\rangle_{\text{GHZ}} = \frac{|000\rangle+|111\rangle}{\sqrt{2}}$, and shares with each of Bob and Charlie one particle from a GHZ triplet. We can see in such an application that a multi-party entanglement GHZ state is needed, which is much harder to prepare than a simple Bell State. And in these applications, the type of entanglement quantum network provides is also different. The key metrics in such applications are the time and quality of such entanglement states, or the fidelity of multi-party entanglement.

D. Quantum Network Traffic Pattern

From previous analysis and modeling, we can see different quantum applications have different QoS metrics. In terms of entanglement delivered, their requirements also differ in the type, rate, and quality of entanglement. These applications also differ in traffic patterns. In the entanglement-based QKD scheme, there will be a one-to-one, fixed-rate, lower bound threshold on fidelity flow between one pair of source-destination pairs. In practical use cases, users who require QKD services may not be adjacent, and the intermediate node in the quantum network may also be required to engage in the entanglement delivery process(with entanglement swapping). As shown in Fig. 4, a source Node B and destination Node E are executing a QKD application, which requires 5 EPRs per time slot, from time slot 1 to time slot 5. On the other hand, distributed quantum computing applications will require a closely located set of network nodes. Different from QKD applications, DQC is much more latency-sensitive, which means a certain number of EPR pairs must be generated in a certain time slot. The traffic pattern of DQC tends to be a collection of changing requirements of EPR pairs. As shown in Fig. 4, one typical sequence of DQC entanglement flow will be 2 EPR pairs between Node F and Node H at time slot 1, 4 EPR pairs between Node E and Node F at time slot 2, 4 EPR pairs between Node I and Node H at time slot 3 and 1 EPR pair between Node E and Node I at time slot 4. Fig. 4, also shows a quantum secret-sharing application that is executed by Node A, D, and G.

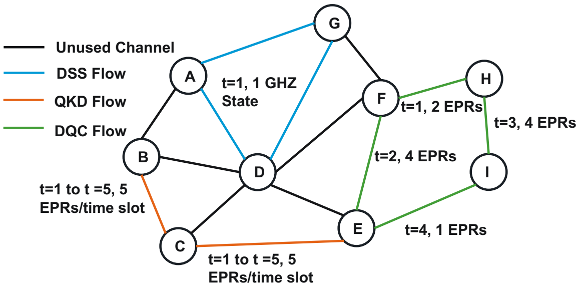

Fig. 4: An example of traffic pattern of different quantum applications in a quantum network, green line denotes

IV. Quantum QoS Routing Framework

One approach to achieving QoS in quantum networks is using QoS-aware routing protocols. Of course, other approaches also exist, such as designing transport and flow control/scheduling protocols. In this paper, we focus the study on routing.

A. Motivation of QoS Routing in Quantum Network

As shown in our analysis and model of different quantum applications, different quantum network applications require different QoS metrics. Quantum applications fall into two categories as delineated by their traffic patterns [18]: Create and Keep (CK) and Measure Directly (MD). CK entails the simultaneous creation of a limited quantity of entangled pairs stored for a duration, while MD involves the immediate generation and measurement of numerous entangled pairs. These differences introduce challenges to the design of all stacks in quantum networks. For example, previous quantum routing work [4], [19] aiming to maximize the overall throughput of the network may not fully meet user requirements due to the inclusion of low-fidelity or late entanglement pairs that do

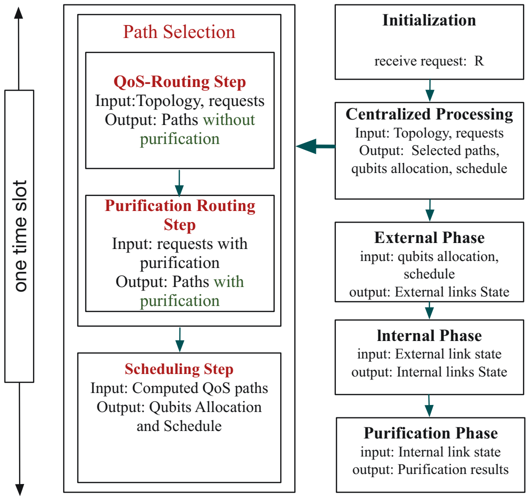
one time slot
Path Selection
QoS-Routing Step
Input:Topology, requests
Output: Paths without purification
Purification Routing Step
Input: requests with purification
Output: Paths with purification
Scheduling Step
Input: Computed QoS paths
Output: Qubits Allocation and Schedule
Initialization
receive request: R
Centralized Processing
Input: Topology, requests
Output: Selected paths, qubits allocation, schedule
External Phase
input: qubits allocation, schedule
output: External links State
Internal Phase
input: External link state
output: Internal links State
Purification Phase
input: Internal link state
output: Purification results

Fig. 5: Phases in one-time slot.

not benefit applications. These applications exhibit differing error tolerance levels and have distinct minimum fidelity requirements that require us to design fidelity-guarantee routing algorithms. In addition, recent work [20] demonstrates a multi-node quantum network using solid-state qubits. The experiment yields an average fidelity of 0.551 across all outcome states and a heralding rate of $1/40$ s^{-1}, showing that basic quantum network operations remain challenging and resource-intensive. Consequently, there is a pressing need to develop innovative protocols that can efficiently utilize resources within quantum networks, aiming to fulfill the requirements of various requests while minimizing resource consumption and maintaining high-performance levels. These considerations inspire our work: A system that can meet requirements for requests with different quality of services such as quality, throughput, and latency, and possess the mechanisms to coordinate among concurrent requests. In this work, we propose a QoS-aware Quantum Routing (Q2R) framework, which can first find feasible paths that meet multiple constraints and later use a scheduling algorithm to coordinate between completing requests and avoid starvation.

B. Q2R Design Overview

1) User's Request: . From the previous section's introduction on different application and their models. Their request for entanglement will at least include the following information: 1) Source-destination pair, which denotes which two nodes require entanglement. 2) Throughput per second denotes the number of entangled pairs generated in this time slot. 3) Minimum Fidelity denotes a minimum fidelity requirement so quantum network applications can run correctly.

2) Five Phase Model: As illustrated in Fig. 5, our QoS framework operates in a time-slotted model. Each time slot comprises five distinct phases: initialization, centralized processing, external, internal, and purification phases. The heart of our design is the centralized processing phase, executed by the central controller which comprises path selection and scheduling steps. The rest of the phases are carried out within the respective quantum nodes. The workflow of our framework is as follows:

Initialization: The central controller begins by updating the previously stored network topology and quantum node information, including the number of qubits and the fidelity of different operations. Afterward, it updates the global information as each user sends its requests.

Centralized Processing The centralized processing takes place within the central controller. Upon receiving user requests, the controller selects one or multiple paths for each request, allocates resources, and schedules requests. Like previous QoS works in data centers and computer networks, our centralized processing functions in a routing-and-scheduling manner, considering both processes jointly. The controller first runs the designed **QoS Routing Algorithm** we proposed to find feasible paths for requests that don't require purifications, including the allocation of qubits along the path. For requests that can't meet the fidelity constraint, a simple heuristic finds path-maximizing delivered entanglement pairs and determines the purification strategy. Following path selection, we find multiple paths and corresponding qubits allocations. These requests then enter the scheduling step to determine which request will be processed, which path will be used, and how many qubits will be consumed along the path in the current time slot.

External Phase After decision-making, the central controller sends its decisions to the corresponding nodes. The nodes first generate entanglement pairs with their adjacent nodes, the number of which is determined by the previous steps. After generating entangled pairs, the adjacent nodes will communicate with their adjacent nodes to exchange state information about their external links.

Internal Phase Nodes perform entanglement swapping based on external link state information. After the swapping along the path, results for requests that don't need purification are sent back to the central controller. Requests needing purification proceed to the purification phase.

Purification Phase Multiple entanglement pairs may be generated between the source and destination pair after the internal phase. Based on the delivered pair, the source and destination nodes perform the BBPSW protocol to purify the entangled pairs, and the final result is returned to the central controller.

The framework allows for pipelining operations. For example, while nodes are generating entanglement with adjacent nodes, the central controller processes requests for subsequent slots. Once the nodes complete their tasks and synchronization, they can immediately process requests for the next time slot.

3) QoS Routing Problem Formulation: This section introduces our path selection step and proposed scheduling step in detail. First, we formulate our problem. Given an arbitrary network topology $G = \langle V, E \rangle$, and a list of requests R where each request r_i can be denoted by one tuple (D_i,N_i,F_i,T_i). Here D_i is one source-destination pair (s_i, d_i) and $s_i, d_i \in V$, N_i is a positive number representing the number of entangled pair, F_i is a number between 0 and 1 representing the minimum fidelity of this request, and T_i is maximum latency in the number of time slots. In fact, for other metrics related to fidelity, such as the gate success rate, F_i can be replaced by these metrics. Our quantum network model is heterogeneous, the number of qubits on one quantum node u can be different from others, and each edge $e \in E$ between two quantum nodes consists of quantum channels which are constrained by the number of available qubits on two end nodes. One request r_i to be fulfilled is defined by the deadline T_i, at least N_i entangled pairs that have a fidelity larger than F_i have been generated. The objective can then be to maximize the fulfilled requests in the network.

C. QoS Routing

1) Routing Metrics: The major goal of the routing step is to find a feasible path that can meet the QoS requirements for each request; specifically, in the path selection step, we want to meet the throughput requirement and fidelity requirement as much as possible. The throughput metric is a widely used expected throughput (EXT) [4]. Suppose a path includes p nodes. Then the number of swapping is $|p| - 1$. Suppose the entanglement fidelity on the i-th edge is F_i, the final end-to-end fidelity is

$$\frac{1}{4} + \frac{3}{4}\prod_{i=1}^{n}\left(\frac{4F_i - 1}{3}\right) \tag{4}$$

2) QoS Routing Step: : The goal of the QoS routing step is to find one or multiple paths, such that the expected throughput of selected path $E_P > N$ and the fidelity of this path $F_P > F$. We have several observations: The objective of the QoS routing step is different from previous works in that previous works mainly focus on maximizing a single metric, such as expected throughput, and can be considered as a 'best-effort' method, and our objective is to find paths that can meet the constrain. QoS routing problems with additive metrics in classical networks have been proven to be NP-complete [6]. Our metrics are neither additive nor linear, making solving QoS routing problems in a quantum network setting even more challenging. We proposed our heuristic $Q2R - MC$. The pseudocode is summarized in the Algorithm. 1. The main techniques in our algorithm are the following concepts: 1) non-linear path functions: We used the idea of linear length function in that we composite our two metrics as follows:

$$l(P) = \left(\left(\frac{E_P}{N^i}\right)^q + \left(\frac{F_P}{F^i}\right)^q\right)^{1/q} \tag{5}$$

This non-linear function can help us determine the selected paths' quality and reduce the search space. If the length of the current path is smaller than 1 and we know any path uses this sub-path will not meet the constraints. The second technique we use is 2) The principle of dominated path. A path Q is said to be dominated path by a path P if $\mathbf{w_i(P)} \leq \mathbf{w_i(Q)}$, in this case, a path with subpath Q can never perform better than a path with subpath P. Besides these two techniques, we also need to keep k paths on each node that meet constraints on the current node. This is due to the nature of the non-linearity of our length: subpaths of the longest paths are not necessarily the longest. Using these techniques, our algorithm works in a Dijkstra fashion, and we maintain a priority queue that follows the highest length first polic and each element is a tuple of $< Path, E_p, F_p, length >$: Our algorithm first initializes empty states and enqueues the source node(line 1 - line 5), it also removes those nodes if their available qubits are fewer than the demands of the request(line 6). Then it extracts the path with the highest length in the queue(line 8). It checks if the current node is the destination, and whether it meets the fidelity requirement and throughput requirement(line 7 - line 11). Then it will scan its neighbors of the current node. It will first compute the new expected throughput and fidelity of the new path(line 19 - line

Algorithm 1: Pseudocode of Q2R-MC

```
   input : $G = \langle V, E\rangle, r, e, F_E, f, k, n$
   output: An array of feasible path Path where each
           element in it is $\langle p, ext, f\rangle$
 1 $Counter \leftarrow 0$
 2 $result \leftarrow$ an array of path, initialize to $\varnothing$
 3 $Path \leftarrow$ an array of $n$ elements, each is a list of paths,
     all set to $\varnothing$
 4 $q \leftarrow$ priority queue, highest length first
 5 $q.enqueue(\langle D.src, +\infty, 1.0, 1.0\rangle)$
 6 $V \leftarrow V.remove(v.Q_u < r.N)$
 7 while $q$ is not empty do
 8     $\langle p, ext, f, l\rangle \leftarrow$q.dequeue()
 9     $u \leftarrow p.last$
10     if $u$ = $dst$ then
11         $\langle p, ext, F_u\rangle \leftarrow$ Construct path with prev, E and F
12         if $ext \geq r.N$ and $f \geq r.F$ then
13             result.add($\langle p, ext, f\rangle$ )
14             if $result.size$=$n$ then
15                 return result
16         end
17     end
18     for $v \in$ neighbors of $u$ do
19         if $v$ in $p$ then continue ;
20         $E_{new} \leftarrow e(p + v, W)$
21         $F_{new} \leftarrow f(F_u, F_E[u, v])$
22         $l \leftarrow length(E_{new}, E_{new})$
23         check if new path $p + v$ is dominated
24         if $l \geq 1$ and new path not dominated then
25             if $counter[v] < k$ then
26                 $Path[v].add(\langle p + v, E_{new}, F_{new}, l\rangle)$
27                 $counter[v] + +$
28             else
29                 $\langle p', E'.F', l'\rangle, j \leftarrow$path in queue with
                     minimum length to v
30                 if $l > l'$ then
31                     Path[v][j] = $\langle p + v, E_{new}, F_{new}, l\rangle)$
32                     Replace in queue old path
                         $\langle p', E'.F', l'\rangle$ with new path
                         $\langle p + v, E_{new}, F_{new}, l\rangle)$
33                 end
34             end
35         end
36     end
37 end
```

23). Then it checks if the new path is dominated by the existing path for the current node(line 23). If not dominated and the number of the stored path on this node is smaller than the tunable size k, it will add the new path with corresponding information to the queue(line 24- line 27). Otherwise, it will replace one path stored in the current node if the length value exceeds the old length(line 28 - line 32).

3) Purification Routing Step: : After the QoS routing step, we will find one or multiple feasible paths for each request to meet the constraint. The entangled pairs of these paths can be generated with entanglement generation and entanglement swapping. However, other requests remain that can't meet fidelity constraints, and thus we need to perform entanglement purification. We adopt a Swap and Purify scheme, which means we will first generate an entangled pair between the source-destination pair and then perform entanglement purification. In this work, we use the well-known BBPSW protocol [9] which is a symmetric purification protocol taking entangled pairs with the same fidelity. It will need to sacrifice multiple low-fidelity entangled pairs to purify entangled pairs. Thus we need to find one path that maximizes the expected throughput. In this step, we use extended Dijsktra's algorithm [4] in Q-CAST, which can maximize the expected throughput between single source-destination pairs. Expected throughput determines the average number of entangled pairs that one path can deliver. Based on the maximal expected throughput, we evaluate how many entangled pairs can be used to purify and whether it can meet the fidelity constraint. If such a path exists, the central controller will add this to the previously computed QoS routing paths and will use extra entangled pairs between source and destination to perform entanglement purification.

4) Scheduling Step: : After the path selection step, we can identify one or more paths for each request that meet the constraints. Our scheduling algorithm aims to determine the optimal allocation of these requests. For example, in the current time slot, we need to decide which request should be processed and which paths should be used. The scheduling step is motivated by several factors: 1) The cost of end-to-end entanglement is high, and given the limited number of available qubits on each node, we aim to use resources in the quantum network efficiently. 2) Previous work [4] adopted a greedy method to schedule requests, always selecting the path that can maximize the expected throughput for a single source-destination pair. However, this is not the optimal global solution since the paths computed at the routing step may compete, and we can take advantage of multiple paths in our framework to avoid competition and use resources more efficiently. 3) Simple scheduling methods such as Early-Deadline-First and Shortest-Job-First do not work well in our setting, as they do not consider per-request information. Instead, we formulate the scheduling problem into a simple optimization problem. Let the P_{ij} denote the j-th path for request r_i, and the corresponding W_{ij} denote the number of qubits used along path P_{ij}, let x_{ij} denotes a binary variable that equals 1 if the j-th path of request i is chosen. The optimization problem is formulated as follows:

$$\text{Maximize:} \quad \sum_i \sum_j x_{ij} \tag{6a}$$

Subject to:

$$\sum_i \sum_j W_{ij} \cdot x_{ij} \le C_{(u,v)} \quad \forall (u,v) \in E, \text{ if } (u,v) \in P_{ij} \tag{6b}$$

$$\sum_i \sum_j 2 \cdot W_{ij} \cdot x_{ij} \le Q_u \quad \forall u \in N, \text{ if } u \in P_{ij} \tag{6c}$$

$$x_{ij} \in \{0,1\} \quad \forall i,j \tag{6d}$$

Each computed path P_{ij} from the previous path selection is a feasible path, and selecting path P_{ij} contributes to the completion of request r_i. We want to maximize this contribution across the network, so we choose $\sum_i \sum_j x_{ij}$ as our objective to be maximized. Constraint 6b enforces the qubit constraint on the number of quantum links on each edge, and constraint 6c ensures that the use of qubits does not exceed the node capacity. Here we only have $\sum_i n_i$ variables where n_i is the number of candidate paths of request r_i, and this is a relatively small scale integer linear programming problem which can be solved efficiently by modern solvers.

V. Evaluation

A. Evaluation Methodology

Network Topology. We do not assume any specific topology, and the network topologies are randomly generated using the Waxman model [21]. The network topology is distributed to 100 KM by 100 KM units square and each unit is 1km. For edge generation, we follow the previous work and use the Waxman model [21] where the distance of each node is at least $\le \frac{50}{\sqrt{|V|}}$. The elementary entanglement success rate is determined by $p_c = e^{-\alpha L}$ where α is a systematic parameter. Given E_p and after network generation, the value α is searched to make average success rate to be $E_p \pm 0.01$ The number of each quantum node is randomly picked from 5 to 10, and the quantum channel on each edge is randomly picked from 3 to 7.

Request parameter. Each user's request is denoted by a tuple (D, T, N, F). We generate users' requests as follows: src-dst pair is picked randomly from the topology. The latency requirement T is uniformly picked from 1 to 5. The throughput requirement N is uniformly picked from 1 to 5. The fidelity requirement is randomly picked from 0.75 to 0.99.

Model of Quantum Operations and Devices To maintain generality, we do not specify any particular physical implementation of quantum networks. Instead, we abstract entanglement swapping and entanglement generation as probabilistic physical processes varying from 0.5 to 0.9.

Default Parameter. In our default setting, the number of quantum nodes is set to 50. The entanglement pair success rate is set to 0.6 and the entanglement swapping success rate is set to 0.9. The average degree of network is 6. For each set, 10 different random networks are generated and we simulate 50 time slots on each of the networks. At each time slot, we generate 20 requests.

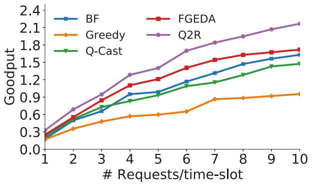

Fig. 6: Relative Goodput with different number of requests ($n = 20$)

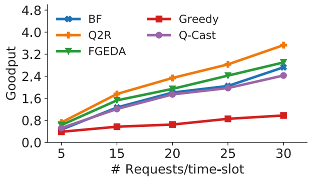

Fig. 7: Goodput with different number of requests ($n = 50$)

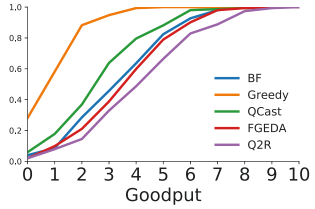

Fig. 8: CDf of Goodput with default setting

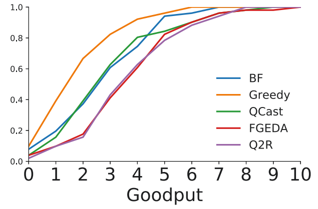

Fig. 9: CDf of Goodput with $n = 150$

Comparison Scheme. The main performance metric we compare in this work is goodput, defined as the fulfilled request per unit of time, which eventually satisfies the application QoS requirements and is used by the applications. To better illustrate relative performance, in the experiments of varying quantum operations, we set Q-CAST algorithm [4] as the baseline and compare the relative performance of our proposed algorithms. Besides these metrics, we also compare the actual running time of different algorithms. We compare the following scheme:

- Q2R: The proposed Q2R framework.
- BF: The BF method is inspired by the classical Bellman-Ford QoS routing algorithm [22]. It searches feasible paths in a Bellman-Ford fashion; here, we used it as a baseline for the classical QoS routing algorithms, and we also selected the path with the highest expected throughput at each time slot.
- Q-CAST [4]: extended Dijkstra's algorithm without but adopts a greedy scheduling strategy that aims to maximize the overall network throughput but does not take fidelity into consideration.
- FGEDA: a heuristic that is based on Q-CAST where we examined the fidelity constraint after we found a path.
- Greedy: always using the paths with the fewest number of hops to establish entanglement.

B. Evaluation on QoS Routing Algorithm

This section evaluates our proposed QoS routing algorithm Q2R-MC with other routing algorithms.

1) Goodput: Our main results are summarized in Fig. 6 and Fig. 7, we vary the number of requests from 1 to 10 with $|V|$ is 20 and requests from [5,10,15,20,25,30] with $|V|$ is 50. We can see from both figures that Q2R performs the best in terms of goodput, with $n = 20$ Q2R-BF can achieve at most 60% improvement compared to Q-CAST and overall 40% improvement, and with $n = 50$, Q2R can achieve 52% improvement. Basically, we can see our proposed algorithms outperform Q-CAST. Greedy algorithms show the lowest performance since they only take a minimum number of hops as a metric. Q-CAST performs better than the Greedy algorithm but performs worse than our proposed algorithm for the following reasons: 1) Q-CAST is still a greedy-fashioned method that aims to maximize the overall throughput but fails to perform per-request service, and this also results in its not utilizing the network resources efficiently. 2) The second reason is that its design did not take fidelity into consideration and thus failed to meet fidelity requirements. In Fig. 8 and Fig. 9, we plot the CDF of Goodput under different node sizes, and the results are calculated in terms of fulfilled requests(frs). From Fig. 8, we can see similar results as previously. From Fig. 9, we can see almost all algorithms degrade, and our proposed method still outperforms other methods. The reason is that when the network scales to 150 nodes, the random generation source-destination pair will also cause the number of hops to increase. Considering the exponential decay of both fidelity and expected throughput, we can see it will be harder for all the algorithms to succeed.

2) Algorithm Running Time: We evaluate the running time of our proposed Q2R-MC and other baselines. We generate 1000 experiments, each including 10 randomly generated requests, and summarize the per-request processing time. As shown in I, Q2R-BF runs slowly among all three algorithms due to the high complexity of BellMan-Ford fashioned algorithms, and both Q2R-MC and FGEDA have a much faster speed. We can conclude that, although running slow with a larger network, Q2R-BF can provide more precise results. Considering the long running time, Q2R-BF can be used to provide ground-truth baselines but can't be used in practical use. Q2R-MC, compared with the other two algorithms, can provide a good tradeoff between running time and accuracy and thus can be practical and efficient in future quantum networks.

	30	50	100	200
Q2R-BF	160.5ms	505.1ms	2151.3ms	7976.9ms
Q2R-MC	15.1ms	31.7ms	42.6ms	65.7ms
FGEDA	9.9ms	13.5ms	25.8ms	36.6ms

TABLE I: Algorithm Running Time for One Request

C. Evaluation on Scheduling Method

We evaluate our proposed scheduling method. Compared schemes are classical network scheduling methods and scheduling methods from previous work, which are as follows:

- Shortest Job First (SJF): SJF prioritizes the processing of requests that require the smallest number of entangled pairs, effectively processing the 'easiest' requests first.
- First-In-First-Out (FIFO): FIFO processes requests in the order they are received, adhering to a strict queue discipline.
- Earliest Deadline First (EDF): This method prioritizes urgency, always processing the request with the nearest deadline first.

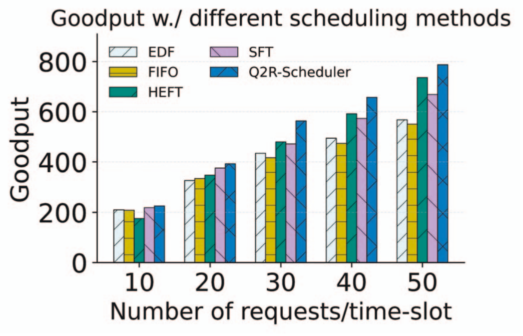

Fig. 10: Goodput vs. request per time-slot

- Highest Expected Throughput First (HEFT): Adopted from Q-CAST [4], which always selects the path with the highest expected throughput.
- Q2R-Schedule: The scheduling method proposed in our framework.

We evaluate how well scheduling methods can resolve competition among requests with limited resources, and vary the number of requests per time slot and the number of qubits per node. As in Fig. 10, we evaluate different scheduling schemes under different requests per time slot. We vary the number of requests sent to the system from 10 to 50. We can see our proposed scheduling scheme consistently outperforms other scheduling methods. Among all the different scheduling methods, FIFO performs the worst since it only schedules in a First-in-First-out fashion, and does not take any request information into consideration. All the other scheduling methods take request information into consideration, and HEFT performs better than other schemes.

VI. Related Work

In designing future quantum networks, addressing the entanglement routing problem is crucial. The entanglement routing problem was first introduced in [3], highlighting the key differences between routing in classical and quantum networks, and proposed a multi-path routing approach. Shi and Qian present the detailed network model of entanglement routing and introduce two routing protocols to maximize expected throughput [4]. Chakraborty *et al.* formulate routing as a multi-commodity flow problem and fidelity is considered by limiting the number of hops [2]. Recently, a QoS framework for the quantum network was proposed in [23] which aims to generate high-quality entanglements for a specific quantum link layer of nitrogen-vacancy (NV) centers in diamond. It explicitly states that routing is "out of the scope" and does not consider generating end-to-end paths that provide QoS, while our work is a routing framework focusing on finding a feasible path that can meet application requirements.

VII. Conclusion

This work studies QoS requirements for quantum networks with various applications. We propose an application-aware QoS routing framework called Q2R that includes a scheduling step and a QoS routing step to meet throughput and fidelity constraints. For the scheduling step, we formulate it as an optimization problem, and for the routing step, we propose two heuristic algorithms: Q2R-MC. Simulation results show that Q2R outperforms existing algorithms regarding the number of fulfilled requests.

Acknowledgment

The authors were partially supported by NSF Grants 1750704, 2114113, and 2322919, and DoE Grant DE-SC0022069. We thank the anonymous reviewers for their comments.

References

[1] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki, "Quantum entanglement," *Reviews of modern physics*, vol. 81, no. 2, p. 865, 2009.

[2] K. Chakraborty *et al.*, "Entanglement distribution in a quantum network: A multicommodity flow-based approach," *IEEE Transactions on Quantum Engineering*, vol. 1, pp. 1–21, 2020.

[3] R. Van Meter, T. Satoh, T. D. Ladd, W. J. Munro, and K. Nemoto, "Path selection for quantum repeater networks," *Networking Science*, vol. 3, no. 1, pp. 82–95, 2013.

[4] S. Shi and C. Qian, "Concurrent entanglement routing for quantum networks: Model and designs," in *SIGCOMM*, 2020, pp. 62–75.

[5] S. Wengerowsky, S. K. Joshi, F. Steinlechner, H. Hübel, and R. Ursin, "An entanglement-based wavelength-multiplexed quantum communication network," *Nature*, vol. 564, no. 7735, pp. 225–228, 2018.

[6] Z. Wang, "On the complexity of quality of service routing," *Information Processing Letters*, vol. 69, no. 3, pp. 111–114, 1999.

[7] R. A. Guerin, A. Orda, and D. Williams, "Qos routing mechanisms and ospf extensions," in *GLOBECOM 97. IEEE Global Telecommunications Conference. Conference Record*, vol. 3. IEEE, 1997, pp. 1903–1908.

[8] R. F. Werner, "Quantum states with einstein-podolsky-rosen correlations admitting a hidden-variable model," *Physical Review A*, vol. 40, no. 8, p. 4277, 1989.

[9] H.-J. Briegel, W. Dür, J. I. Cirac, and P. Zoller, "Quantum repeaters: the role of imperfect local operations in quantum communication," *Physical Review Letters*, vol. 81, no. 26, p. 5932, 1998.

[10] A. K. Ekert, "Quantum cryptography based on bell's theorem," *Physical review letters*, vol. 67, no. 6, p. 661, 1991.

[11] C. H. Bennett, G. Brassard, and N. D. Mermin, "Quantum cryptography without bell's theorem," *Physical review letters*, vol. 68, no. 5, p. 557, 1992.

[12] A. Scherer, B. C. Sanders, and W. Tittel, "Long-distance practical quantum key distribution by entanglement swapping," *Optics express*, vol. 19, no. 4, pp. 3004–3018, 2011.

[13] C. Monroe, R. Raussendorf, A. Ruthven, K. R. Brown, P. Maunz, L.-M. Duan, and J. Kim, "Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects," *Physical Review A*, vol. 89, no. 2, p. 022317, 2014.

[14] Y. Mao, Y. Liu, and Y. Yang, "Qubit allocation for distributed quantum computing," in *INFOCOM 2023*. IEEE, 2023, pp. 1–10.

[15] S. S. Tannu and M. K. Qureshi, "Not all qubits are created equal: A case for variability-aware policies for nisq-era quantum computers," in *Proceedings ASPLOS*, 2019, pp. 987–999.

[16] M. Ben-Or and A. Hassidim, "Fast quantum byzantine agreement," in *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, 2005, pp. 481–485.

[17] M. Hillery, V. Bužek, and A. Berthiaume, "Quantum secret sharing," *Physical Review A*, vol. 59, no. 3, p. 1829, 1999.

[18] A. Dahlberg *et al.*, "A link layer protocol for quantum networks," in *Proceedings of ACM SIGCOMM*, 2019, pp. 159–173.

[19] Y. Zhao and C. Qiao, "Redundant entanglement provisioning and selection for throughput maximization in quantum networks," in *INFOCOM 2021*. IEEE, 2021, pp. 1–10.

[20] M. Pompili *et al.*, "Realization of a multinode quantum network of remote solid-state qubits," *Science*, vol. 372, no. 6539, pp. 259–264, 2021.

[21] B. M. Waxman, "Routing of multipoint connections," *IEEE journal on selected areas in communications*, vol. 6, no. 9, pp. 1617–1622, 1988.

[22] X. Yuan, “On the extended bellman-ford algorithm to solve two-constrained quality of service routing problems,” in *Proceedings Eight International Conference on Computer Communications and Networks (Cat. No. 99EX370)*. IEEE, 1999, pp. 304–310.

[23] M. Skrzypczyk and S. Wehner, “An architecture for meeting quality-of-service requirements in multi-user quantum networks,” *arXiv preprint arXiv:2111.13124*, 2021.