IEEE INFOCOM 2024 - IEEE Conference on Computer Communications | 979-8-3503-8350-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/INFOCOM52122.2024.10621359

Quantum BGP with Online Path Selection via Network Benchmarking

Maoli Liu[†1], Zhuohua Li[†*1], Kechao Cai[2], Jonathan Allcock[3], Shengyu Zhang[3], John C.S. Lui[1]
[1]*The Chinese University of Hong Kong,* [2]*Sun Yat-sen University,* [3]*Tencent Quantum Laboratory*
{mlliu, zhli, cslui}@cse.cuhk.edu.hk, caikch3@mail.sysu.edu.cn, jonallcock@tencent.com, shengyuzhang@gmail.com

***Abstract*—Large-scale quantum networks with thousands of nodes require topology-oblivious routing protocols to realize. Most existing quantum network routing protocols only consider the *intra-domain* scenario, where all nodes belong to a single party with complete topology knowledge. However, like the classical Internet, quantum Internet will likely be provided by multiple quantum Internet Service Providers (qISPs). In this paper, we consider the *inter-domain* scenario, where the network consists of multiple subnetworks owned by mutually untrusted parties without centralized control. Under this setting, previously proposed quantum entanglement routing policies, which rely on the network topology knowledge, are no longer applicable. We propose a Quantum Border Gateway Protocol (QBGP) for efficiently routing entanglement across qISP boundaries. To guarantee high-quality information transmission, we propose an algorithm named *online top-K path selection*. This algorithm utilizes the *information gain* introduced in this paper to adaptively decide on measurement parameters, allowing for the selection of *high-fidelity* paths and accurate fidelity estimates, while minimizing costs. Additionally, we implement a quantum network simulator and evaluate our protocol and algorithm. Our evaluation shows that QBGP effectively distributes entanglement across different qISPs, and our path selection algorithm increases the network performance by selecting high-fidelity paths with much lower resource consumption than other methods.**

***Index Terms*—Quantum Networks, Entanglement Routing**

I. Introduction

Quantum networks transmit quantum information (called *quantum bits*, or *qubits*) between separated quantum systems, enabling applications like quantum cryptography [1] and quantum key distribution (QKD) [2] that are impossible with classical networks alone. Quantum networks can also help to perform distributed quantum computation [3] by connecting multiple small quantum computers together. The unique features of quantum mechanics [4] make the design of quantum networks very different from classical networks. For example, similar to classical bits, qubits can be transmitted via physical quantum links (e.g., optical fibers or free space). But simply constructing quantum networks via physical links is unrealistic because the loss of quantum coherence fundamentally limits the successful transmission rate, which decays exponentially with the length of any physical quantum links. Moreover, due to the no-cloning theorem [5], an arbitrary qubit can neither be copied for re-transmission nor be amplified to eliminate noise. Therefore, quantum information is susceptible to corruption via inevitable *decoherence* during the transmission. As a result, the *store-and-forward* transmission scheme used in classical packet switching networks is no longer applicable to quantum networks. Instead, people propose *entanglement-based networks*, which leverage quantum mechanics and rely on trusted intermediate nodes (called quantum *repeaters*) to help construct end-to-end *entanglement links* (or called *virtual links*) between source and destination nodes. Quantum entanglement can be regarded as resources that can be consumed for transmitting qubits by a process known as quantum *teleportation* [6]. Many experiments [7]–[10] have demonstrated that entanglement-based quantum networks can be realized in practice.

The ultimate goal of quantum networks is to realize large-scale and long-distance quantum communication. To achieve this, different types of protocols and hardware are needed. In particular, *quantum entanglement routing* protocols need to be executed at each routing node in order to construct desired end-to-end quantum entanglement in the network. Briefly speaking, an entanglement routing protocol takes a routing request as input, which is a source-destination (Alice-Bob) pair. The protocol is responsible for finding a path from Alice to Bob in the network. Then each repeater along the path is scheduled to generate entanglement with its neighbors and then performs *entanglement swapping* [6], [11] operations. Finally, an end-to-end virtual link is established between Alice and Bob.

To realize long-distance entanglement routing, many protocols [12]–[20] have been proposed. However, all of them focus on the intra-domain scenario, i.e., all nodes are operated by a single quantum Internet Service Provider (qISP). This implies that there is a centralized server that maintains topology information, sends updates to all nodes, and synchronizes all the nodes on a global clock. Similar to the classical Internet, we envision that in the future, the global quantum Internet service will be provided by multiple qISPs around the world. In this case, there is no centralized server for maintaining network information. Each network provider only knows the topology of its own sub-network, and will not reveal the topology to other network providers for privacy concerns. Classical Internet solves this problem by using the Border Gateway Protocols (BGP), which exchange reachability information among different Autonomous Systems (ASes), and make routing decisions at boundary routers. Thus, it is necessary to come up with a quantum analog of BGP, which

[†]Maoli Liu and Zhuohua Li are co-first authors. *Zhuohua Li is the corresponding author. The work of John C.S. Lui was supported in part by the RGC SRFS2122-4S02. The work of Kechao Cai was supported in part by NSF China under Grant No. 62202508 and by Shenzhen Science and Technology Program (Grant No. 202206193000001, 20220817094427001).

achieves *inter-domain* routing while overcoming the difficulties of realizing entanglement-based quantum networks.

To make routing decisions, quantum networks need some metrics to determine the quality of a virtual link. People usually quantify the quality using the so-called *fidelity* [21], a value from 0 to 1, measuring how well a quantum channel preserves quantum information. To provide fidelity guarantees, many existing works try to design entanglement routing algorithms that use shortest paths [16], perform entanglement purification [19], or speed up entanglement generation [22], etc. However, only considering path lengths is insufficient because the fidelity of a virtual link is not solely determined by its path length due to probabilistic factors like noisy swapping operations and decoherence during qubit storage. Moreover, all of these efforts assume that all nodes have sufficient knowledge about the whole network, e.g., the network topology and the current entanglement states. So these methods do not work under the inter-domain scenario. Another way to make routing decisions is to measure the average fidelities of each path in the routing table using topology-oblivous techniques such as *network benchmarking* [23]. Unfortunately, directly using network benchmarking in routing protocols will consume a large number of quantum resources.

In this work, we tackle the problems mentioned above. First, we propose *Quantum Border Gateway Protocol (QBGP)*, the quantum analog of the classical Border Gateway Protocol, enabling inter-domain entanglement routing. QBGP is a distributed protocol that runs asynchronously in each boundary repeater (called *QBGP speaker*), which establishes end-to-end quantum entanglement using only local information in its routing table (details in Section IV-B). Second, we formulate the path selection and fidelity estimation problem as a *best-K arms identification* problem [24] and design an online top-K path selection algorithm based on network benchmarking, aiming to find high-fidelity AS paths in a routing table and give accurate fidelity estimates while minimizing quantum resource consumption. We introduce the concept of *information gain* for better choices of parameters for the network benchmarking subroutine in our algorithm (details in Section IV-C).

To evaluate our protocol and algorithm, we implement a packet-based event-driven quantum network simulator, which simulates packet-switching in the classical channel and qubit transmission in the quantum channel simultaneously. Our evaluation results show that QBGP can flexibly and effectively establish end-to-end virtual links across different qISPs. Our path selection algorithm increases the network performance by selecting high-fidelity paths while consuming much fewer quantum resources than other methods.

We summarize our contributions as follows.

- We propose QBGP, an inter-domain entanglement routing protocol for quantum networks provided by multiple qISPs, which does not require a centralized server to control all the routers or update topology information.
- We design an online top-K path selection algorithm to help QBGP to select paths with higher fidelities and we also provide a theoretical analysis. Our algorithm leverages online learning techniques and the information gain to reduce quantum resource consumption.
- We implement a packet-based and event-driven quantum network simulator to evaluate our protocol and algorithm. The results demonstrate their effectiveness. The source code is available online (See Section VII), which can be the basis of other research in the future.

II. Background

A. Quantum Networks

1) Quantum Networks: A quantum network consists of *quantum nodes* and *quantum links*. Each quantum node is equipped with a *quantum processor* that can perform quantum operations and measurements, and store a certain number of qubits in its *quantum memory*. A pair of quantum nodes are called "neighbors" if they are connected by a quantum link, which can be any quantum channel such as optical fibers or free-space links. Quantum links are equipped with a quantum source used for generating quantum entanglement. In addition, all quantum nodes can transmit classical information to each other via the classical Internet.

2) Quantum Entanglement and Teleportation: Quantum entanglement is a phenomenon where the joint state of multiple qubits cannot be factored into a product of states belonging to each qubit separately. One typical example is the Einstein-Podolsky-Rosen (EPR) state [25]. Direct entanglement between neighboring nodes can be established, for example, by placing a quantum source in the middle and bidirectionally distributing entangled photon pairs through optical fibers.

Quantum teleportation is a procedure where a qubit from the source node is recreated in the destination node (while destroying the original qubit) by consuming an entanglement link between two nodes. Provided that the source and the destination share an entanglement link, teleportation can be performed over arbitrary lengths. Establishing long-distance entanglement between two quantum nodes can be done by using *quantum repeaters* as relays, stitching over multiple short-distance entanglement links into a long-distance one. This is called *entanglement swapping* [6], [11], as shown in Figure 1. A quantum repeater node (R) is placed midway between the source (S) and the destination (D). If R shares two entangled pairs with S and D respectively (Figure 1 (a)), it can perform entanglement swapping, which teleports the entangled qubit half shared with S to D, leading to an entanglement shared between S and D (Figure 1 (b)).

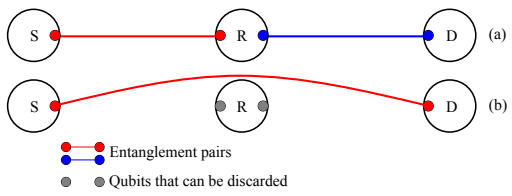


Fig. 1. Entanglement swapping.

B. BGP and Inter-domain Routing

The classical Internet is a network of networks provided by multiple Internet Service Providers (ISPs), and Autonomous Systems (ASes) are sub-networks that make up the whole Internet. Specifically, an AS is a collection of network infrastructures controlled by a single administrative entity or domain that has a unified routing policy. Inside an AS, an intra-domain routing protocol is defined so that all nodes can communicate with each other and derive the network topology of the AS. Then based on the topology, routers can calculate the shortest path to the destination. Typical intra-domain routing protocols used in classical Internet are OSPF [26] or RIP [27], etc.

Different ASes also interconnect via dedicated links to support inter-AS communications. Each AS is identified by an Autonomous System Number (ASN) and announces its routing information to other ASes via the Border Gateway Protocol (BGP). The boundary routers that connect to another AS, called *BGP speakers*, launch BGP peering sessions and exchange routing information. BGP represents a destination using a *network prefix*, which is an aggregation of IP addresses (e.g. 10.20.30.0/24). A BGP advertisement consists of a list of network prefixes, and the corresponding *AS paths* that can reach the prefixes. An AS path is a sequence of ASes along the path, identified by ASNs. Upon receiving BGP advertisements, the BGP speakers select the most preferred AS path based on several criteria, such as distance and peer relationship. Then the selected AS path is updated to their routing tables, and advertised to their neighbors. When propagating the advertisements, BGP speakers will prepend their ASNs to the AS path. Eventually, all the BGP speakers store an AS path for each prefix.

Inter-domain routing is necessary because it creates the backbone of the whole network, enabling different ISPs to build large-scale networks while keeping the topology inside each AS private. Without inter-domain routing, it is difficult to negotiate the routing policies between different ISPs. Moreover, as the network scale increases, exchanging routing tables and computing shortest paths will quickly become infeasible, and packets will either be lost or take a long time to be delivered.

C. Network Benchmarking

1) Noise and Average Fidelity: The inevitable noise in the quantum realm is often modeled by a *noise channel* [4]. One typical example is the *depolarizing channel*, where the input state stays intact with probability p, or becomes a maximally mixed state otherwise. p is called the *depolarizing parameter*. *The average fidelity* of the depolarizing channel with parameter p is $(1+p)/2$ [4], capturing how well a link preserves an arbitrary state sent through it. Since the average fidelity of end-to-end virtual links established along a given path indicates the quality of that path, we also call it "path fidelity" for convenience. In this paper, fidelity, path fidelity and average fidelity are equivalent when there is no ambiguity.

2) Network Benchmarking: Under the Markovian assumption [23], quantum links established along the same path at different times always correspond to the same noise. The idea of network benchmarking is to first transform a quantum channel to a depolarizing channel with the same fidelity via *channel twirling* [28], then measure the depolarizing parameter and deduce the fidelity of the original channel.

The details of network benchmarking are as follows. Assuming a path P between Alice and Bob, Alice first chooses a bounce length set $\mathcal{M}$. For each $m \in \mathcal{M}$, as shown in Algorithm 1, Alice generates a qubit and performs m bounces with Bob (channel twirling), applies a final operation (Line 10), and measures the outcome (Line 11). A *bounce* above involves Alice applying a random Clifford operation [29] to a state and teleporting it to Bob (Line 6, 7), and Bob then applying an operation and teleporting it back (Line 16, 17). Alice repeats this for each m multiple times and obtains its average measurement result b_m. Finally, Alice uses regression with the model $b_m = Ap^{2m}$ to estimate p of the twirled channel and calculates the average fidelity $(1+p)/2$.

Algorithm 1: Network Benchmarking with Fixed Bounces [23]

Input: Path ID between Alice and Bob P, bounce length m
Output: b

1 **Function** `AliceBenchmarkingProtocol(`P, m, *bob*`)`**:**
2 Choose two random operation sequences of length m from the Clifford operations: $\{G_{A,1}, \ldots, G_{A,m}\}$, $\{G_{B,1}, \ldots, G_{B,m}\}$
3 Send signal and $\{G_{B,1}, \ldots, G_{B,m}\}$ to *bob*
4 Generate an initial quantum state ρ
5 **for** $i = 1, 2, \ldots, m$ **do**
6 $\rho \leftarrow G_{A,i}(\rho)$
7 Request entanglement via path P and teleport ρ to *bob*
8 Request entanglement via path P and receive the quantum state ρ from *bob*
9 Choose a final operation G_A
10 $\rho \leftarrow G_A \circ \left(\bigcirc_{i=1}^{m} G_{B,i} \circ G_{A,i}\right)^{-1}(\rho)$
11 Measure the state ρ and obtain the outcome b

12 **Function** `BobBenchmarkingProtocol(`*alice*`)`**:**
13 Receive signal and $\{G_{B,1}, \ldots, G_{B,m}\}$ from *alice*
14 **for** $i = 1, 2, \ldots, m$ **do**
15 Receive the quantum state ρ from *alice*
16 $\rho \leftarrow G_{B,i}(\rho)$
17 Teleport ρ to *alice*

III. Design

A. Main Ideas

Our design contains two protocols for (1) inter-domain reachability propagation (Section IV-A) & entanglement routing (Section IV-B) and (2) benchmarking end-to-end virtual links (Section IV-C). They cooperate and mutually improve each other. On the one hand, when the network is initialized, or when its topology changes, QBGP propagates reachability information across the network and helps each router to maintain multiple paths for each destination. Then for the same destination, the entanglement routing protocol can distribute entanglement along different paths, leading to multiple end-to-end virtual links with different fidelities. Users can use these links to perform our online top-K path selection algorithm. On the other hand, our path selection algorithm identifies top-K high-fidelity links, which help the source node to reorder its routing table, so it will select better paths for future requests.

To illustrate, we give a motivating example in Figure 2, where the network consists of five network providers corresponding to AS_1 to AS_5. Each AS has several boundary routers that connect to other ASes. As for the internal nodes in each AS, we assume that they are connected, and thus we omit the internal topology. The example shows three paths (denoted in red, blue, and green) from the source (S) to the destination (D). The red path and the blue path have the same AS path in the routing table ($AS_1 \rightarrow AS_2 \rightarrow AS_3$), but the red path may have better fidelity since it goes through fewer internal nodes in AS_2. Our online top-K path selection algorithm can help us to identify and utilize high-fidelity paths.

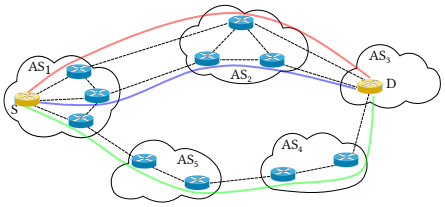


Fig. 2. A network topology consisting of 5 ASes.

B. Network Models and Assumptions

We mimic the structure of the classical Internet and assume that our quantum network has the following components.

1) Autonomous Systems: An Autonomous System (AS) is a collection of quantum nodes controlled by a single administrative entity. Within an AS, the network operator defines a single intra-domain routing policy, through which the quantum nodes can establish entanglement with each other. Previous studies have proposed numerous intra-domain entanglement routing protocols. Here, we do not assume any specific protocol is being used, and different ASes may even use different routing protocols. Instead, we abstract out the details inside ASes, and our inter-domain protocol only invokes the intra-domain protocol defined in each AS as a sub-protocol.

2) Quantum Repeaters and QBGP Speakers: Quantum repeaters are intermediate nodes that work as relays to convert multiple small entanglement links into longer-distance end-to-end entanglement links. The number of qubits that can be stored in the repeater's quantum memory is called "capacity". QBGP speakers are quantum repeaters that are located at the boundary of ASes and run QBGP. Each AS may own multiple QBGP speakers, and each speaker has a direct connection with other speakers in another AS. When performing inter-domain routing across different ASes, a QBGP speaker is responsible for receiving and propagating reachability information and works as a repeater for handling routing requests.

IV. Algorithms

A. Quantum Border Gateway Protocols

We adapt the classical BGP for quantum networks. Compared to classical networks, one significant property of quantum networks is that quantum communication is probabilistic. This contrasts with the deterministic nature of classical BGP, which defines a series of rules to decide a single, optimal path for packet forwarding. Such a strategy is not applicable in quantum networks since the deterministic path selection rule may not always yield high-quality paths. Additionally, sending traffic across only one path may lead to network congestion due to the limited capacity of quantum repeaters and is vulnerable to link failures. To address these challenges, QBGP adaptively changes its preference over multiple paths using our top-K path selection algorithm instead of using deterministic rules for single-path selection. The benchmarking results are stored in the source node's routing table, which guides the source node to choose either one or multiple high-fidelity paths.

We emphasize that QBGP can maintain the beneficial properties of the classical BGP, such as loop prevention and routing policies based on business relationships. However, our primary focus lies in exploring the foundational mechanisms of the QBGP algorithm. Therefore, while examining these properties is practically significant, it falls outside the main scope of our current study and will not be extensively discussed.

B. Inter-Domain Entanglement Routing Protocol

After QBGP makes routing decisions, the inter-domain entanglement routing protocol is executed to distribute entanglement along the selected AS path. Our protocol differs from existing works in two aspects: (1) Most existing works require that the global network topology is known by all nodes. In this case, a path can be first discovered from the source to the destination. Then all nodes along the path are notified to reserve required resources and operate for this request in parallel. This implicitly requires a centralized server to control and synchronize all the repeaters along the path. On the contrary, our protocol is oblivious to the global network topology so it does not construct the path in advance. Instead, each node only knows what the next hop is by searching the address in its routing table, and the request will be forwarded hop-by-hop. (2) In most existing works, when failures happen (e.g., due to decoherence), the centralized control can schedule a retry until it succeeds or the request expires. On the contrary, our protocol does not have centralized control, so when failures happen, we need to backtrack the nodes in the path and clean up their resources.

We present the main procedure of our protocol in Algorithm 2. The protocol is executed on each speaker asynchronously and expressed as multiple callback functions. Since there is no centralized control, the execution of each callback function is only triggered upon receiving a corresponding event.

1) Request Forwarding and Entanglement Generation: The routing protocol starts when the network receives a routing request (source-destination pair), where the source can be any node inside the network, and the destination is an identifier (like an IP address). Since we focus on inter-domain routing, we assume that the source and the destination belong to different ASes. We also assume that every request has a unique identifier, such that the protocol knows which resources are reserved for which requests. As shown in function `HandleRoutingRequest`, upon receiving a request, the current node first uses `GetNextHop` to find the next-hop

Algorithm 2: Inter-Domain Entanglement Routing

```
Input: Current speaker: node, Request message: request
// Execute upon receiving a routing request
Function HandleRoutingRequest(node, request):
    next_hop ← GetNextHop(request)
    if node.qmemory is not full then
        if next_hop.ASN == node.ASN then
            Generate entanglement between node and next_hop
              via intra-domain protocol
        else Generate entanglement directly
        request.path.append(node.ID)
        ForwardRequest(next_hop, request)
    else return Failure(request)
// Execute upon receiving a qubit
Function HandleQubit(node, qubit, request):
    next_hop ← GetNextHop(request)
    if node.qmemory is not full then
        node.qmemory[request].append(qubit)
        if len(request.path) == 2 then
            return Success(request)
        else if len(node.qmemory[request]) == 2 then
            m ← BellMeasure(node.qmemory[request])
            node.qmemory[request].clear()
            ForwardMeasurement(next_hop, m)
    else return Failure(request)
// Execute upon receiving a measurement result
Function HandleMeasurement(node, m, request):
    next_hop ← GetNextHop(request)
    if node is the destination of request then
        node.meas_result.append(m)
        if len(node.meas_result) == len(request.path) - 1 then
            qubit ← node.qmemory[request].pop()
            Correction(qubit, node.meas_result)
    else ForwardMeasurement(next_hop, m)
```

speaker according to its own routing table, then tries to establish entanglement with it. At the same time, it forwards the request to the next hop via the classical channel. Note that every intermediate node will append its identifier in the request message so that succeeding nodes know the number of hops the request has gone through.

The entanglement generation towards the next-hop speaker differs depending on whether the current speaker and the next-hop speaker are within the same AS. If the next-hop speaker has the same AS number (ASN) as the current speaker does, the intra-domain routing protocol is executed based on the routing policies defined in the corresponding AS. Otherwise, the two boundary speakers must be directly connected via a quantum link, so the quantum source can be triggered to distribute entanglement. No matter how entanglement is generated, upon receiving a qubit, function `HandleQubit` is executed, where the speaker stores the received qubit and its associated request identifier. When the entanglement links for both directions (upstream and downstream) are established for a request, i.e., there are two qubits associated with a request, the node will perform Bell measurement on these two qubits. The measurement results will be forwarded to the next-hop speaker in order to complete the swapping procedure.

2) Buffering Measurement Results and Swapping: A speaker executes function `HandleMeasurement` when it receives a measurement result. According to the associated request identifier, the node first determines whether it is the destination node of this request. If it is not the destination, the measurement result will be forwarded to the next hop. Otherwise, it will buffer the measurement result for the request until all the measurement results are received from the intermediate nodes, i.e., when the number of measurement results equals the number of preceding repeaters minus one. Then it performs the correction procedure to complete the entanglement swapping. Note that the order of the measurement results does not matter, so the measurement results only need to specify which request it belongs to, and no other classical communication is required.

3) Success/Failure Notification: Since there is no centralized control, it is important to notify successful / failed requests via the classical channel in order to clean up resources reserved in all the intermediate nodes. For all the intermediate repeaters, the cleanup procedure is straightforward because after they perform the Bell measurements, their work is done and they can immediately discard their qubits for the corresponding request. To notify the success of a request, when the destination node finishes the correction procedure, it sends an acknowledgment to the source node, meaning that the entanglement is ready. On the other hand, when a request fails (e.g., when some node's quantum memory is run out), the request will not be forwarded. Furthermore, all the preceding repeaters (stored in the request message) are notified so that they can release their reserved resources for the request.

C. Online Top-K Path Selection Algorithm

To reliably transmit quantum information, one naïve way is to measure the fidelity of each path respectively using *network benchmarking* (Section II-C) and choose high-fidelity paths for communication. However, this would incur significant costs because network benchmarking consumes the same amount of resources for each path, regardless of their fidelity. It is wasteful to precisely benchmark low-fidelity paths that are unsuitable for quantum transmission. To address this problem, we design an online learning algorithm, which adaptively learns each path's fidelity and discards inferior paths as early as possible.

1) Problem Formulation: Given a source-destination pair, suppose the routing table $\mathcal{L}$ of the source node contains L AS paths to the destination, $\mathcal{L} = \{1, 2, \ldots, L\}$. The fidelity of path i, denoted by f_i, can be written as $f_i = (p_i + 1)/2$, where p_i is the depolarizing parameter of the twirled channel [4]. If applying the vanilla network benchmarking for each path i, it collects data $\{b_m\}_{m\in\mathcal{M}}$ and fits it $b_m = A_i p_i^{2m}$ to get the estimate of p_i. Selecting the bounce length $m \in \mathcal{M}$ is important as it determines both the benchmarking cost and the accuracy of fidelity estimation. For a bounce length m, network benchmarking consumes $2m$ entangled pairs, and the Fisher information we obtain by observing b_m is $F(p_i, m) = 32{A_i}^2 m^2 p_i^{(4m-2)}$ [23]. Thus, the average Fisher information for each entanglement is $I(p_i, m) \coloneqq F(p_i, m)/2m = 16{A_i}^2 m p_i^{(4m-2)}$. We call $I(p_i, m)$ the *information gain*, which measures the amount of information for each entanglement given the bounce length m. A higher information gain can lead to a greater reduction of uncertainty given the same cost. We find that there exists an optimal bounce length such that the information gain is

maximized, i.e., $m_i{}^* = \arg\max_{m \in \mathcal{M}} I(p_i, m)$ and this can help us to choose optimal bounce lengths from $\mathcal{M}$, so as to minimize the quantum resource consumption.

We formulate the path estimation and selection problem as a best-K arms identification problem. Let $\mathcal{L} = \{1, \ldots, L\}$ be the set of L arms, where an arm corresponds to an AS path in the routing table. Each arm $i \in \mathcal{L}$ is associated with a random variable P_i, representing the depolarizing parameter of the twirled channel from path i. The mean of P_i is denoted by $p_i = \mathbb{E}[P_i]$, which is unknown beforehand. Without loss of generality, we assume that $P_i \in [0, 1]$ for any $i \in \mathcal{L}$ and $p_1 \geq \cdots \geq p_K > p_{K+1} \geq \cdots \geq p_L$. Pulling arm i at round t yields a reward $p_i(t)$, which is drawn independently from the distribution of P_i. Here, pulling arm i means that we choose a suitable bounce length m_i and get $p_i(t)$. Our goal is to find the top-K arms with the highest mean values from set $\mathcal{L}$.

2) Algorithm Design: Our online top-K path selection algorithm is described in Algorithm 3. Let $\mathcal{L}(t)$ be the remaining paths in $\mathcal{L}$, $\hat{p}_i(t)$ be the empirical mean of arm i, and $\text{UCB}_i(t)$/$\text{LCB}_i(t)$ (Line 5, 6) be the upper/lower confidence bound of $\hat{p}_i(t)$ at round t. Initially, we have no knowledge of the AS paths in the routing table $\mathcal{L}$. Therefore, we start an initialization phase to gain initial information on each path. We select a set of bounce lengths $\mathcal{M}_{\text{init}}$ and a repetition number N_{rep}. For each path $i \in \mathcal{L}$, we apply Algorithm 1 for each $m \in \mathcal{M}_{\text{init}}$ for N_{rep} times to get the average $b_{i,m}$ (Line 1) and get $\hat{p}_i$ and A_i via regression (Line 2). Note that $\mathcal{M}_{\text{init}}$ is of small size, so $\hat{p}_i$s are not precise enough in this phase. Next, we launch an online exploration phase. For each round t, the algorithm first calculates $\text{UCB}_i(t)$ and $\text{LCB}_i(t)$ for each arm $i \in \mathcal{L}(t)$, and finds the best $K - |\mathcal{L}_{\text{good}}|$ arms with the highest upper confidence bounds in $\mathcal{L}(t)$, denoted by $\mathcal{H}(t)$. Then the algorithm identifies "bad" and "good" arms (Line 9, 12), and moves the identified arms to $\mathcal{L}_{\text{bad}}$ and $\mathcal{L}_{\text{good}}$, respectively. The algorithm stops if it identifies K good paths; Otherwise, the algorithm pulls each arm $i \in \mathcal{L}(t)$, i.e., determines a bounce length m_i by maximizing the information gain $I(\hat{p}_i(t), m)$(Line 18), calls Algorithm 1 (Line 19) and obtains an observation $p_i(t)$ using the aforementioned function (Line 20), and then updates $\hat{p}_i(t+1)$ (Line 21). For practical purposes, we also set two thresholds h_1 and h_2 and update A_i to avoid wasting too many resources.

3) Complexity Analysis: We analyze the sample complexity of Algorithm 3. We define gaps Δ_i for each path $i \in \mathcal{L}$: $\Delta_i = p_i - p_{K+1}$ if $i \leq K$ and $\Delta_i = p_K - p_i$ if $i \geq K$. We have the following attractive properties.

Theorem 1. *For any $\delta > 0$, with probability at least $1 - \delta$, Algorithm 3 finds the best K paths with the sample complexity $\mathcal{O}\left(\sum_{i=1}^{L} \Delta_i^{-2} \log\left(\frac{L}{\Delta_i \delta}\right)\right)$.*

Remark. Theorem 1 shows that our algorithm can quickly find high-fidelity paths especially when some links have large fidelity gaps. However, the sample complexity for uniformly benchmarking each path using the vanilla network benchmarking at an accuracy of ϵ is $\mathcal{O}(L\epsilon^{-2}\log(\frac{L}{\epsilon\delta}))$, which leads to a large quantum resource consumption. We refer interested readers to Appendix A for the proof.

Algorithm 3: Online Top-K Path Selection Algorithm

Input: $\mathcal{L}$, $\mathcal{M}_{\text{init}}$, N_{rep}, $\mathcal{M}_{\text{loop}}$, L, K, δ, h_1, h_2
Output: high-fidelity paths and fidelity information

1 $b_{i,m} \leftarrow$ `Average` (*Algorithm 1*(i, m), N_{rep}), $\forall i \in \mathcal{L}, m \in \mathcal{M}_{\text{init}}$
2 A_i, $\hat{p}_i \leftarrow$ `Regression` ($\mathcal{M}_{init}$, $\{b_{i,m}\}_{m \in \mathcal{M}_{init}}$) for $i \in \mathcal{L}$
3 $\hat{p}_i(1) \leftarrow \hat{p}_i$ for $i \in \mathcal{L}$; $\mathcal{L}(1) \leftarrow \mathcal{L}$; $\mathcal{L}_{\text{good}} \leftarrow \varnothing$; $\mathcal{L}_{\text{bad}} \leftarrow \varnothing$
4 **for** $t = 1, 2, \cdots$ **do**
5 $\quad \text{UCB}_i(t) \leftarrow \hat{p}_i(t) + \sqrt{\log(4Lt^2/\delta)/(2t)}$ for $i \in \mathcal{L}(t)$
6 $\quad \text{LCB}_i(t) \leftarrow \hat{p}_i(t) - \sqrt{\log(4Lt^2/\delta)/(2t)}$ for $i \in \mathcal{L}(t)$
7 $\quad \mathcal{H}(t) \leftarrow K - |\mathcal{L}_{\text{good}}|$ paths in $\mathcal{L}(t)$ with the highest UCBs
8 $\quad$ **for** $i \in \mathcal{L}(t) \setminus \mathcal{H}(t)$ **do**
9 $\quad\quad$ **if** $\min_{j \in \mathcal{H}(t)} \text{LCB}_j(t) > \text{UCB}_i(t)$ **then**
10 $\quad\quad\quad \mathcal{L}_{\text{bad}} \leftarrow \mathcal{L}_{\text{bad}} \cup \{i\}$
11 $\quad$ **for** $i \in \mathcal{H}(t)$ **do**
12 $\quad\quad$ **if** $\text{LCB}_i(t) > \max_{j \in \mathcal{L}(t) \setminus \mathcal{H}(t)} \text{UCB}_j(t)$ **then**
13 $\quad\quad\quad \mathcal{L}_{\text{good}} \leftarrow \mathcal{L}_{\text{good}} \cup \{i\}$
14 $\quad \mathcal{L}(t) \leftarrow \mathcal{L}(t) \setminus (\mathcal{L}_{\text{good}} \cup \mathcal{L}_{\text{bad}})$
15 $\quad$ **if** $|\mathcal{L}_{\text{good}}| \geq K$ or $|\mathcal{L}_{\text{bad}}| \geq L - K$ **then return** $\mathcal{L}_{\text{good}}$
16 $\quad$ **if** $\min_{i \in \mathcal{H}(t)} \text{LCB}_i(t) > h_1$ or $\max_{i \in \mathcal{L}(t) \setminus \mathcal{H}(t)} \text{UCB}_i(t) < h_2$ **then return** $\mathcal{L}_{\text{good}} \cup \mathcal{H}(t)$
17 $\quad$ **for** $i \in \mathcal{L}(t)$ **do**
18 $\quad\quad m_i \leftarrow \arg\max_{m \in \mathcal{M}_{\text{loop}}} I(\hat{p}_i(t), m)$
19 $\quad\quad b_{i,m_i}(t) \leftarrow$ Algorithm 1(i, m_i)
20 $\quad\quad p_i(t) \leftarrow (b_{i,m_i}(t)/A_i)^{1/(2m_i)}$
21 $\quad\quad \hat{p}_i(t+1) \leftarrow (\hat{p}_i(t) * t + p_i(t))/(1+t)$
22 $\quad\quad$ Update A_i via regression
23 $\quad \mathcal{L}(t+1) \leftarrow \mathcal{L}(t)$

V. Performance Evaluation

A. Simulator Implementation

We implement a simulator based on an off-the-shelf quantum network simulation framework called NetSquid [30]. Our implementation is packet-based and event-driven, i.e., we do not assume the existence of a global controller, and each node can only communicate with each other via sending/receiving messages, meaning that we can simulate distributed algorithms. Our simulator is asynchronous, i.e., each router individually runs its protocol, receiving messages and responding accordingly. So instead of waiting for all the routing requests to arrive in each time slot and then executing the routing algorithms, our simulator runs each protocol's callback functions immediately once receiving the triggering events. This enables us to simulate distributed "online" protocols. All the experiments were done on a machine with a 3.70 GHz Intel Xeon E5-1630 v4 CPU and 32GB RAM, running Linux (kernel 5.15.88).

B. Methodology

1) Network Topology and Traffic Simulation: We evaluate our protocols in both synthetic and real-world AS-level network topology datasets. For synthetic topology, we use the generation algorithm by Elmokashfi et al. [31], such that the generated graph satisfies several AS-level topological characteristics such as the hierarchical structure and strong clustering. For the real-world dataset, we use the data collected by the University of Oregon Route Views Project [32], which contains 6,474 nodes (ASes) and 13,895 edges. To get real-world topologies with different sizes, we sample subgraphs with specific sizes from

the real-world dataset by uniformly sampling nodes until we can induce a connected subgraph from these nodes. Both the synthetic and real-world datasets are AS-level topologies and do not contain the topology of the BGP speakers, so we randomly generate speakers for each AS by defining a parameter that specifies the average number of neighbors a speaker connects. As a result, the number of speakers in each AS is proportional to the degree of the AS in the AS-level graph.

To launch the QBGP routing information propagation, we randomly generate network addresses and assign them to random ASes. Then, these ASes will send QBGP announcements to their neighbors and propagate the reachability information until convergence. We simulate traffic by randomly choosing source-destination pairs, where the source can be any node in the network, and the destination is one of the propagated addresses. The source-destination pairs are emitted by the source node as routing requests, following the Poisson distribution with a fixed parameter. Since we focus on the inter-domain scenario, we exclude all the requests where the source and the destination are within the same AS.

2) Performance Metrics: We quantify the network performance using *throughput* and *goodput*, respectively. We define network *throughput* as the number of successfully generated end-to-end virtual links divided by the elapsed time, and we define network *goodput* as the summation of the fidelity of all the successfully generated end-to-end virtual links divided by the elapsed time. Note that *throughput* does not consider the fidelity of each link and only measures the "connectivity capability" of the protocol, while *goodput* quantifies the quality of the connections. On the one hand, we use *throughput* to evaluate the performance of QBGP since it is responsible for connecting quantum nodes across different ASes and it is fidelity agnostic. On the other hand, we use *goodput* to show that the noise introduced by qubit transmission and measurement significantly impacts the quality of the network service, and our *online top-K path selection* algorithm can effectively select high-fidelity paths so as to improve the network performance. In addition, we use the number of bounces to quantify the quantum resource consumption.

C. Evaluation Results

1) Performance of QBGP: To demonstrate the performance of QBGP, we vary the network scale and resources to evaluate how our protocol performs when facing different amounts of traffic. As shown in Figure 3, when the number of requests is relatively small, the throughput increases because there are sufficient quantum resources in each repeater. When the number of requests continues to increase, the throughput gradually converges because the capacity of some repeaters has run out, and the network is congested. We also vary the number of ASes and the capacity of each speaker, and the results show that a larger network size/capacity leads to a higher carrying capacity. Overall, our evaluation shows that QBGP performs entanglement routing effectively across different ASes.

[1] "ebits/s" refers to the rate of transmitting entangled bits (ebits) per second.

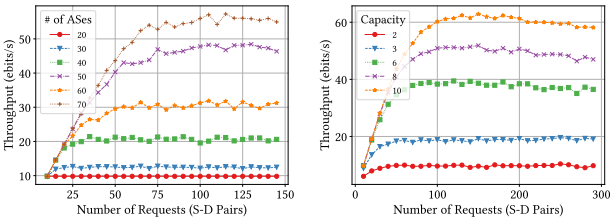
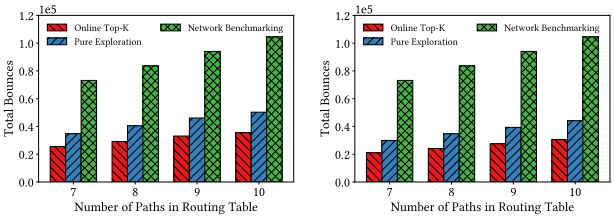


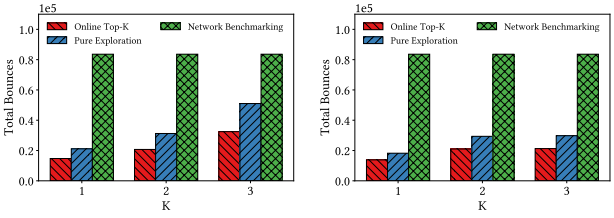
Fig. 3. Throughput v.s. number of ASes and capacity.[1]

2) Performance of Online Top-K Path Selection: We demonstrate the performance of our online top-K path selection algorithm by evaluating the quantum resource consumption and fidelity estimation accuracy. We compare our algorithm with the vanilla network benchmarking [23] and an online pure exploration algorithm, which is similar to Algorithm 3, but randomly chooses bounce lengths m_i from $\mathcal{M}_{\text{loop}}$ at each round t instead of maximizing the information gain (Line 18). We randomly select a source-destination pair from the network, then pick K good paths from L paths in the source node's routing table by respectively applying the three algorithms. In Figure 4, we fix $K = 3$ and vary the number of paths L; and in Figure 5, we keep $L = 8$ and vary the value of K. The results show that our path selection algorithm always consumes fewer bounces. To show our path selection algorithm estimates fidelities accurately, we fix $K = 3$ and $L = 6$ and apply the three algorithms to the path set of a randomly chosen source-destination pair on the synthetic topology network. All the algorithms identify K good paths correctly, and the estimated fidelity results are listed in Table I. The evaluation demonstrates that our online top-K path selection algorithm performs well from both accuracy and cost perspectives.



(a) Synthetic topology (b) Real-world topology

Fig. 4. Bounces v.s. number of paths in the routing table.



(a) Synthetic topology (b) Real-world topology

Fig. 5. Bounces v.s. K.

3) Necessity of Selecting High-Fidelity Paths: Here, we use the depolarizing noise model to depict the noise introduced by

TABLE I
ESTIMATED FIDELITY

	Path 1	Path 2	Path 3	Path 4	Path 5	Path 6	Avg. Error (%)
Ground Truth Fidelity	0.9399	0.9728	0.9719	0.9679	0.9575	0.9588	N/A
Network Benchmarking	0.9411	0.9731	0.9696	0.9671	0.9578	0.9577	1.04
Online Top-K (Ours)	0.9390	0.9720	0.9710	0.9673	0.9556	0.9578	1.05
Pure Exploration	0.9412	0.9720	0.9710	0.9668	0.9568	0.9576	1.04

the quantum channel and the measurement operation. We show the impact of noise in quantum networks and emphasize the importance of selecting good paths for generating high-fidelity virtual links in Figure 6 and Figure 7.

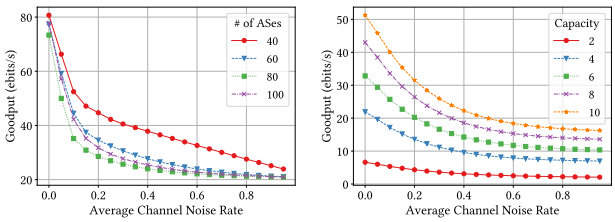


Fig. 6. Goodput v.s. channel noise.

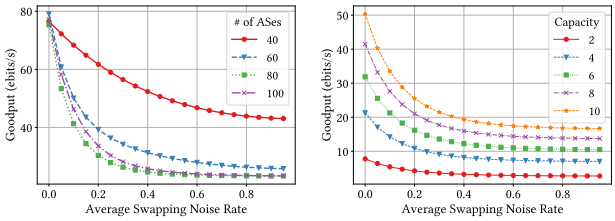
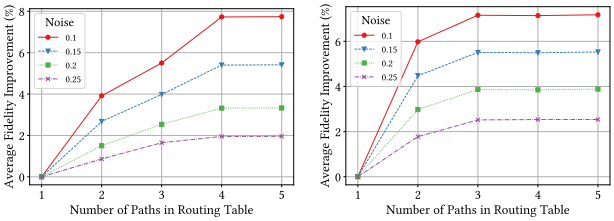


Fig. 7. Goodput v.s. measurement noise.

The goodput drops significantly if the noise is large, especially when the network has a large scale. This is because the longer distance a qubit is transmitted and the more swapping is made, the more coherence will be lost. When the noise is small, large networks usually can achieve higher goodput. However, as the noise increases, the fidelities of the generated end-to-end virtual links are affected, and eventually, users with long distances can hardly establish virtual links, and the goodput becomes even lower than a small network.

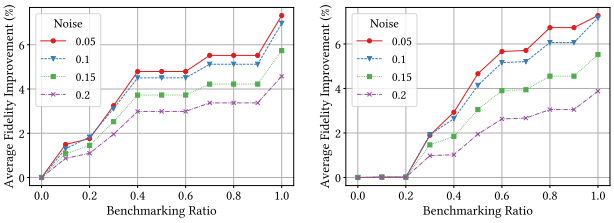
4) Network Performance Improvement by Online Top-K Path Selection: To demonstrate the effectiveness of our online top-K path selection algorithm, we randomly select source-destination pairs and compare the network performance of the default shortest path first routing policy and our path selection algorithm respectively.

We first show the improvements in the average fidelity of the virtual links. We vary the number of paths stored in the routing table for each QBGP speaker and evaluate the fidelity improvements, as shown in Figure 8. The noise means the depolarizing rate of the quantum channels. As the number of paths increases, our algorithm has more candidate paths to explore, meaning that it has more chances to discover a high-fidelity path, so the average fidelity of the generated virtual links also increases. We also fix the number of paths in the routing table and vary the number of source-destination pairs that we benchmark. We call the fraction of benchmarked pairs as "benchmarking ratio". Figure 9 shows that the more source-destination pairs our algorithm explores, the larger fidelity improvement we gain.



(a) Synthetic topology (b) Real-world topology

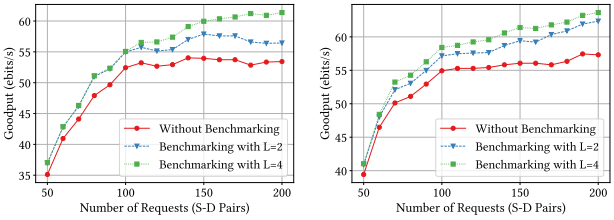
Fig. 8. Fidelity increase v.s. number of paths.



(a) Synthetic topology (b) Real-world topology

Fig. 9. Fidelity increase v.s. benchmarking ratio.

Then we show the network performance improvements in terms of goodput. We vary the number of paths in the routing table and the benchmarking ratio respectively. As shown in Figure 10, compared to the default shortest path first policy (i.e., without benchmarking), our online top-K path selection algorithm can improve the network goodput by more than 15% when the number of requests is large. Figure 11 shows that given a fixed number of paths, exploring more paths leads to better network performance. Finally, selecting high-fidelity paths is essential especially when the traffic is heavy because some requests may fail due to congestion, and choosing high-fidelity paths can sustain the service quality as much as possible.

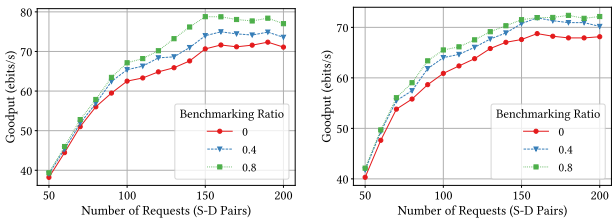


(a) Synthetic topology (b) Real-world topology

Fig. 10. Goodput v.s. number of paths in the routing table.

VI. RELATED WORK

The entanglement routing problem is an active research area. Many existing works [13], [14], [33]–[35] focus on theoretical analysis and only consider specific network topologies such as grid, ring, star, or diamond. Recent studies focus more on general network topologies. [36] adapts Dijkstra's shortest



(a) Synthetic topology (b) Real-world topology

Fig. 11. Goodput v.s. benchmarking ratio.

path algorithm in quantum repeater networks. [16] provides a mechanism to recover from link failures. [37] considers fidelity by setting a cutoff time to discard qubits that take too long to be distributed. [17], [19] use quantum purification to make sure that the fidelity is above a threshold. However, all of them rely on the knowledge of global network topologies and hence do not work in the inter-domain scenario. There are also some works that do not require global network topology. [14] proposes a distributed algorithm such that each router provides local best efforts routing, but without considering fidelity. [12] proposes a greedy algorithm based on the small-world phenomenon, but it requires a specific type of topology.

To assess quantum link quality, [38] proposes a protocol for estimating the fidelity of entangled pairs, but it does not consider state preparation or measurement errors. [39] presents quantum link verification protocols with a focus on fault diagnostics. Network benchmarking [23] estimates quantum link fidelity in a topology-oblivious way. However, it is designed to measure the fidelity of a single link and incurs high costs if directly applying it to routing decisions.

The best arm identification problem with fixed confidence has been extensively studied [40]–[42], aiming at identifying the best arm using as few samples as possible. A line of research extends the problem to the best-K arms identification problem [43]–[45], which identifies top K arms.

VII. Conclusion

In this paper, we proposed Quantum Border Gateway Protocol (QBGP) and an online top-K path selection algorithm to achieve entanglement routing under the inter-domain scenario and provide high-quality quantum communication by selecting high-fidelity paths. Our evaluation showed that our QBGP could efficiently establish virtual links among different qISPs, and our path selection algorithm improved the network performance while consuming much fewer resources than other methods.

The authors have provided public access to their code and/or data at https://zenodo.org/doi/10.5281/zenodo.10444190.

Appendix

A. Proof of Theorem 1

First, we introduce the following lemma, which can be proved by Hoeffding's inequality. For convenience, we define $U(t,L,\delta) := \sqrt{\log\left(4Lt^2/\delta\right)/(2t)}$.

Lemma 1. *Let $X_1,\dots,X_L$ be L independent random variables with $X_i \in [0,1]$ almost surely for $i \in [L]$. Then for any $\delta > 0$, with probability at least $1-\delta$, we have*

$$\left|\tfrac{1}{t}\textstyle\sum_{s=1}^{t}(X_{i,s}-\mu_i)\right| \le U(t,L,\delta), \quad \forall t \ge 1, \forall i \in [L].$$

Proof of Theorem 1. Denote the set of paths with the top-K highest fidelities by $\text{TOP}_K = \{1,\dots,K\}$. At time t, if Algorithm 3 does not stop, we have $|\mathcal{L}_{\text{good}}| < K$ and $|\mathcal{L}_{\text{bad}}| < L-K$. Denote $k(t) = |\mathcal{L}_{\text{good}}|$. $\mathcal{H}(t)$ is the set of paths with the top $K-k(t)$ highest empirical means in $\mathcal{L}(t)$. Define an event $\zeta = \{\forall i \in [L], \forall t \ge 1, |\frac{1}{t}(\sum_{s=1}^{t} X_{i,s} - \mu_i)| \le U(t,L,\delta)\}$. We claim that if ζ holds, the algorithm will not make mistakes, i.e., the top-K paths will be in $\mathcal{L}_{\text{good}}$, while non-top-$K$ paths will be in $\mathcal{L}_{\text{bad}}$. We discuss the following two cases.

Case 1. Suppose that path i is moved into $\mathcal{L}_{\text{bad}}$ at time t. For any $j \in \mathcal{H}(t)$, we have

$$\hat{p}_j(t) - U(t,L,\delta) > \hat{p}_i(t) + U(t,L,\delta). \tag{1}$$

On the event ζ, $|\hat{p}_i(t) - p_i| \le U(t,L,\delta)$ holds for all $i \in \mathcal{L}(t)$. Therefore, we have

$$p_j + U(t,L,\delta) \ge \hat{p}_j(t), \quad p_i - U(t,L,\delta) \le \hat{p}_i(t). \tag{2}$$

(1) and (2) imply that $p_j > p_i$ for all paths $j \in \mathcal{H}(t)$. Since the algorithm has already identified $k(t)$ good arms at time t, it only needs to identify the best $K-k(t)$ paths in $\mathcal{L}(t)$. If arm i is one of the best $K-k(t)$ paths in $\mathcal{L}(t)$, there must exist a certain path $j \in \mathcal{H}(t)$ and j does not belong to the best $K-k(t)$ paths, i.e., $p_i \ge p_j$. However, we have $p_j > p_i$ for all paths $j \in \mathcal{H}(t)$, which is a contradiction. Thus, arm i is not one of the best $K-k(t)$ paths in $\mathcal{L}(t)$, or equivalently, arm i does not belong to TOP_K.

Case 2. Suppose path i is moved into $\mathcal{L}_{\text{good}}$ at time t. Similarly, on the event ζ we have $p_i > p_j$ for all paths $j \in \mathcal{L}(t) \setminus \mathcal{H}(t)$, meaning that path i is one of the best $K-k(t)$ paths in $\mathcal{L}(t)$. So if the event ζ holds, a good path belonging to TOP_K will be put into $\mathcal{L}_{\text{good}}$, while a bad path which does not belong to TOP_K will be put into $\mathcal{L}_{\text{bad}}$. By Lemma 1, we have $\Pr\{\zeta\} \ge 1-\delta$. Therefore, with probability at least $1-\delta$, Algorithm 3 finds the top-K paths when it terminates.

Now we prove the sample complexity. According to the elimination and acceptance rule, one of the events that can remove a bad arm i from $\mathcal{L}(t)$ is

$$\min_{j\in\mathcal{H}(t)} \hat{p}_j(t) - U(t,L,\delta) > \hat{p}_i(t) + U(t,L,\delta). \tag{3}$$

Let path j be one of the TOP_K paths in $\mathcal{H}(t)$. On the event ζ, we have $\hat{p}_j(t) \ge p_j - U(t,L,\delta)$ and $\hat{p}_i(t) \le p_i + U(t,L,\delta)$. Note that $p_j - 2U(t,L,\delta) \ge p_i + 2U(t,L,\delta)$ implies that (3) holds with high probability. Since $\min_{j\in\text{TOP}_K} p_j - p_i = p_K - p_i = \Delta_i$, we have $\Delta_i \ge 4U(t,L,\delta)$. Thus, to remove a bad path i, the number of samples T_i should satisfy $T_i \le c\Delta_i^{-2}\log\big(L\log\left(\Delta_i^{-2}\right)/\delta\big)$, where c is a constant. Similarly, we can get the number of samples T_j needed to add a good path j to $\mathcal{L}_{\text{good}}$.

Therefore, the sample complexity of Algorithm 3 is $\mathcal{O}\left(c\sum_{i=1}^{L}\Delta_i^{-2}\log\left(\frac{L\log\left(\Delta_i^{-2}\right)}{\delta}\right)\right)$, which completes the proof of Theorem 1. □

REFERENCES

[1] A. K. Ekert, "Quantum cryptography based on bell's theorem," *Phys. Rev. Lett.*, vol. 67, pp. 661–663, Aug 1991.

[2] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," in *Proceedings of the International Conference on Computers, Systems & Signal Processing*, 1984, pp. 175–179.

[3] J. I. Cirac, A. K. Ekert, S. F. Huelga, and C. Macchiavello, "Distributed quantum computation over noisy channels," *Phys. Rev. A*, vol. 59, pp. 4249–4254, Jun 1999.

[4] M. A. Nielsen and I. Chuang, "Quantum computation and quantum information," 2002.

[5] W. K. Wootters and W. H. Zurek, "A single quantum cannot be cloned," *Nature*, vol. 299, no. 5886, pp. 802–803, 1982.

[6] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, "Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels," *Phys. Rev. Lett.*, vol. 70, pp. 1895–1899, Mar 1993.

[7] C. Elliott, D. Pearson, and G. Troxel, "Quantum cryptography in practice," in *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '03, 2003, p. 227–238.

[8] W. Chen, Z.-F. Han, T. Zhang, H. Wen, Z.-Q. Yin, F.-X. Xu, Q.-L. Wu, Y. Liu, Y. Zhang, X.-F. Mo, Y.-Z. Gui, G. Wei, and G.-C. Guo, "Field experiment on a "star type" metropolitan quantum key distribution network," *IEEE Photonics Technology Letters*, vol. 21, no. 9, pp. 575–577, 2009.

[9] S. Wang, W. Chen, Z.-Q. Yin, H.-W. Li, D.-Y. He, Y.-H. Li, Z. Zhou, X.-T. Song, F.-Y. Li, D. Wang, H. Chen, Y.-G. Han, J.-Z. Huang, J.-F. Guo, P.-L. Hao, M. Li, C.-M. Zhang, D. Liu, W.-Y. Liang, C.-H. Miao, P. Wu, G.-C. Guo, and Z.-F. Han, "Field and long-term demonstration of a wide area quantum key distribution network," *Opt. Express*, vol. 22, no. 18, pp. 21 739–21 756, Sep 2014.

[10] C. Elliott, "The darpa quantum network," in *Quantum Communications and cryptography*. CRC Press, 2018, pp. 91–110.

[11] M. Żukowski, A. Zeilinger, M. A. Horne, and A. K. Ekert, ""event-ready-detectors" bell experiment via entanglement swapping," *Phys. Rev. Lett.*, vol. 71, pp. 4287–4290, Dec 1993.

[12] L. Gyongyosi and S. Imre, "Decentralized base-graph routing for the quantum internet," *Phys. Rev. A*, vol. 98, p. 022310, Aug 2018.

[13] M. Pant, H. Krovi, D. Towsley, L. Tassiulas, L. Jiang, P. Basu, D. Englund, and S. Guha, "Routing entanglement in the quantum internet," *npj Quantum Information*, vol. 5, no. 1, p. 25, Mar. 2019.

[14] K. Chakraborty, F. Rozpedek, A. Dahlberg, and S. Wehner, "Distributed routing in a quantum internet," 2019.

[15] K. Chakraborty, D. Elkouss, B. Rijsman, and S. Wehner, "Entanglement distribution in a quantum network: A multicommodity flow-based approach," *IEEE Transactions on Quantum Engineering*, vol. 1, pp. 1–21, 2020.

[16] S. Shi and C. Qian, "Concurrent entanglement routing for quantum networks: Model and designs," in *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication*, ser. SIGCOMM '20, 2020, p. 62–75.

[17] C. Li, T. Li, Y.-X. Liu, and P. Cappellaro, "Effective routing design for remote entanglement generation on quantum networks," *npj Quantum Information*, vol. 7, no. 1, pp. 1–12, Jan. 2021.

[18] A. Farahbakhsh and C. Feng, "Opportunistic routing in quantum networks," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, 2022, pp. 490–499.

[19] Y. Zhao, G. Zhao, and C. Qiao, "E2e fidelity aware routing and purification for throughput maximization in quantum networks," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, 2022, pp. 480–489.

[20] J. Li, M. Wang, K. Xue, R. Li, N. Yu, Q. Sun, and J. Lu, "Fidelity-guaranteed entanglement routing in quantum networks," *IEEE Transactions on Communications*, vol. 70, no. 10, pp. 6748–6763, 2022.

[21] M. A. Nielsen, "A simple formula for the average gate fidelity of a quantum dynamical operation," *Physics Letters A*, vol. 303, no. 4, pp. 249–252, 2002.

[22] M. Ghaderibaneh, H. Gupta, C. Ramakrishnan, and E. Luo, "Pre-distribution of entanglements in quantum networks," in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, sep 2022, pp. 426–436.

[23] J. Helsen and S. Wehner, "A benchmarking procedure for quantum networks," *npj Quantum Information*, vol. 9, no. 1, p. 17, 2023.

[24] S. Bubeck, T. Wang, and N. Viswanathan, "Multiple identifications in multi-armed bandits," in *International Conference on Machine Learning*. PMLR, 2013, pp. 258–265.

[25] A. Einstein, B. Podolsky, and N. Rosen, "Can quantum-mechanical description of physical reality be considered complete?" *Phys. Rev.*, vol. 47, pp. 777–780, May 1935.

[26] J. Moy, "OSPF Version 2," Internet Engineering Task Force, Request for Comments RFC 2328, Apr. 1998. [Online]. Available: https://datatracker.ietf.org/doc/rfc2328

[27] C. Hedrick, "Routing Information Protocol," Internet Engineering Task Force, Request for Comments RFC 1058, Jun. 1988. [Online]. Available: https://datatracker.ietf.org/doc/rfc1058

[28] E. Magesan, J. M. Gambetta, and J. Emerson, "Characterizing quantum gates via randomized benchmarking," *Physical Review A*, vol. 85, no. 4, p. 042311, 2012.

[29] D. Gottesman, "The heisenberg representation of quantum computers," *arXiv preprint quant-ph/9807006*, 1998.

[30] T. Coopmans, R. Knegjens, A. Dahlberg, D. Maier, L. Nijsten, J. de Oliveira Filho, M. Papendrecht, J. Rabbie, F. Rozpędek, M. Skrzypczyk, L. Wubben, W. de Jong, D. Podareanu, A. Torres-Knoop, D. Elkouss, and S. Wehner, "NetSquid, a NETwork Simulator for QUantum Information using Discrete events," *Communications Physics*, vol. 4, no. 1, pp. 1–15, Jul. 2021.

[31] A. Elmokashfi, A. Kvalbein, and C. Dovrolis, "On the scalability of bgp: The role of topology growth," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 8, pp. 1250–1261, 2010.

[32] U. of Oregon Route Views Project. (2023) University of oregon route views project. [Online]. Available: https://www.routeviews.org/routeviews/

[33] G. Vardoyan, S. Guha, P. Nain, and D. Towsley, "On the stochastic analysis of a quantum entanglement switch," *SIGMETRICS Perform. Eval. Rev.*, vol. 47, no. 2, p. 27–29, dec 2019.

[34] S. Pirandola, "End-to-end capacities of a quantum communication network," *Communications Physics*, vol. 2, no. 1, pp. 1–10, May 2019.

[35] M. Caleffi, "Optimal routing for quantum networks," *IEEE Access*, vol. 5, pp. 22 299–22 312, 2017.

[36] 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, Dec. 2013.

[37] W. Kozlowski, A. Dahlberg, and S. Wehner, "Designing a quantum network protocol," in *Proceedings of the 16th International Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '20, 2020, p. 1–16.

[38] L. Ruan, "Minimization of the estimation error for entanglement distribution networks with arbitrary noise," *Phys. Rev. A*, vol. 108, p. 022418, Aug 2023.

[39] M. Liu, J. Allcock, K. Cai, S. Zhang, and J. C. Lui, "Quantum networks with multiple service providers: Transport layer protocols and research opportunities," *IEEE Network*, vol. 36, no. 5, pp. 56–62, 2022.

[40] E. Even-Dar, S. Mannor, Y. Mansour, and S. Mahadevan, "Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems." *Journal of machine learning research*, vol. 7, no. 6, 2006.

[41] S. Mannor and J. N. Tsitsiklis, "The sample complexity of exploration in the multi-armed bandit problem," *Journal of Machine Learning Research*, vol. 5, no. Jun, pp. 623–648, 2004.

[42] K. Jamieson and R. Nowak, "Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting," in *2014 48th Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2014, pp. 1–6.

[43] S. Kalyanakrishnan, A. Tewari, P. Auer, and P. Stone, "Pac subset selection in stochastic multi-armed bandits." in *ICML*, vol. 12, 2012, pp. 655–662.

[44] M. Simchowitz, K. Jamieson, and B. Recht, "The simulator: Understanding adaptive sampling in the moderate-confidence regime," in *Conference on Learning Theory*. PMLR, 2017, pp. 1794–1834.

[45] H. Jiang, J. Li, and M. Qiao, "Practical algorithms for best-k identification in multi-armed bandits," *arXiv preprint arXiv:1705.06894*, 2017.