Incremental Deployment for a QKD Network

Luyao Luo Gongming Zhao

***Abstract*—**

Index Terms*—*Quantum Network.

I. Introduction

Secure information exchange via quantum networks has been proposed, studied, and validated since 1980s [5, 17, 18, 34, 37, 47, 58] and many experimental studies have demonstrated that long-distance secrete sharing via quantum networks can become successful in reality, such as the DARPA quantum network [18], SECOQC Vienna QKD network [37], the Tokyo QKD network [47], and the satellite quantum network in China [58]. A quantum network (also called a quantum Internet) is an interconnection of quantum processors and repeaters that can generate, exchange, and process quantum information [8, 10, 24, 56]. It transmits information in the form of quantum bits, called qubits, and stores qubits in quantum memories. Quantum networks are not meant to replace the classical Internet communication. In fact, they supplement the classical Internet and enable a number of important applications such as quantum key distribution (QKD) [5, 17, 40], clock synchronization [25], secure remote computation [7], and distributed consensus [15], most of which cannot be easily achieved by the classical Internet.

With the help of quantum key distribution, it supports highly secure communication between two terminals. With this significant advantage, many users, including corporations and governments, are expecting to deploy a QKD network. As there has already deployed a traditional data communication network, a primary task for applying a QKD network is how to efficiently deploy a QKD network while integrating with the current data network. To be compatible with the traditional data network, we will deploy a set of quantum relay devices and configure a certain number of qubits on each relay. When users expect to deploy a QKD network, they should consider the following factors and constraints:

1) Since QKD network just provides quantum key for communication, it should be applied with data networks. As a result, the deployment of QKD networks should be integrated with the deployment of traditional networks.
2) With a limited budget, users may ask to deploy a small number of trusted quantum relays and activate a small number of qubits, so that each pair can transmit the data with required quantum key.
3) Traffic dynamic is a common and important issue in a network.

These practical factors and constraints bring some challenges for designing an efficient QKD network. As deployment is an important issue for application of a QKD network, some studies [12] [13] have focused on this problem recently.

After deployment of QKD devices, it is another critical task to design an efficient routing mechanism in the newly deployed network.

This paper focuses on how to efficiently deploy a QKD network while considering all the above factors and constraints. We study the incremental deployment strategy, which will preserve the benefits of legacy systems. To make this possible, we address two critical technical challenges: the incremental deployment and multi-commodity flow routing in a network.

The rest of this paper is organized as follows. Section 2 discusses the related works on the deployment and routing problems under the QKD network paradigm. Section 3 gives some preliminaries. In section 4, we define the incremental deployment problem, and propose an approximate algorithm to deal with this challenge. The experimental results are presented in Section 6. We conclude the paper in Section 7.

II. Preliminaries

A. QKD Network Topology Model

In QKD network we mainly consider two kinds of deployment: quantum relay depolyment and quantum device depolyment. That is, how to choose proper locations to establish relays and how to place enough quantum devices to satisfy endpoints communication requirements. Let $N = \{n_1, n_2, ..., n_m\}$ represent the QKD endpoint set, where $m = |N|$ is the number of endpoints. $Q = \{q_1, q_2, ..., q_k\}$ represents the relay set, where $k = |Q|$ is the number of relays. Let y_{n_i} and y_{q_j} denote the number of quantum devices in endpoint n_i and relay q_j. And E denotes the set of link, such as optical fibers, between endpoints and relays. We perform QKD between two nodes(endpoints or relays) through a quantum channel built on a physical link of E.

To achieve the security requirements of relays, we can not choose random places to deploy relays in practice. Based on the existing traditional network topology, the locations of relays can coincide with some key locations(such as core switches) in a candidate location set, which is denoted as $P = \{p_1, p_2, ..., p_s\}$, with $s = |P|$. That's to say, relay set Q is a subset of candidate location set P, and each pair of endpoints can perform QKD through some relays in Q with quantum channel length constraint satisfied.

Notation	Explanation
C_r	The cost for deploying a quantum relay
C_d	The cost for deploying a quantum device
R_0	The secret key generation capacity of a quantum device
$R(t,p)$	The secret key generation rate needed per unit of throughput from position p to next hop through path t
$w(f)$	Average communication throughput demand of S-D pair f

TABLE I: Some notations used in this paper

B. Problem Statement

In this section, we propose the problem statement of QKD network topology control (QNTC). Under the metropolitan area network, some organizations will deploy some quantum key distribution endpoints(QKD Endpoint) for secure communications and each pair of endpoints may have QKD requirement. Direct communication may be not supported because QKD endpoints can be far apart, so we need to deploy some trusted QKD relays to implement quantum key distribution. To achieve the security requirements of trusted relays, we can not choose random places to deploy relays. Based on the existing traditional network topology, the locations of trusted relays can coincide with some key locations(such as core switches) in a candidate location set P. QKD network topology control(QNTC) relies on the deployment of trusted relays and optical fibers to connect these QKD endpoints.

We assume that each endpoint pair needs at least one quantum channel, so there must be at least one available path for each endpoint pair that satisfies quantum constraints. There are mainly two constraints. One is the quantum resource capacity of a relay, which leads to the limited number of quantum channels passing through this relay. The other one is the link length constraint, that is, the length of any link must not exceed a given threshold, otherwise the quantum signals will not be measured correctly.

Considering the high price of quantum devices, we want to deploy as few devices as possible. However, due to the concurrency requirement(that is, the set of paths are contention-free while all the S-D pairs communicate at the same time), the number of quantum devices in each relay should be enough to satisfy the path width requirement for each pair.

The QNTC problem decides how to choose candidate positions to deploy quantum relays and how to deploy quantum devices in each relay. and our goal is to reduce the network deployment cost.

C. Notations and definitions

We summarize the some notations used in the rest of this paper in Table 1 and make the following explanations.

III. QKD Network Deployment

This section first defines the problem of quantum network topology Control(QNTC) and then gives an approximation algorithm to solve this problem.

A. Problem Definition for QNTC

In this section, we give the problem definition for QKD network topology control (QNTC). We assume that the cost for each quantum relay is denoted as α and the cost for quantum devices for each quantum channel passing through each relay is denoted as β. Our goal is to minimum the total cost while deploying the whole QKD network, which is consist of the cost for relays and the cost for devices.

We use x_i to denote that whether to deploy relay at position $p_i \in P$ ($x_i = 1$) or not ($x_i = 0$). And y_i denotes the number of quantum channel passing through position $p_i \in P$. Since there should be at least one quantum channel between each S-D pair, there must be at least one available path(includes relays and endpoints) where the physical length of each link is less than L_{max}. For any S-D pair f, we can find all the candidate paths that satisfy the link length constraint, denoted as T_f. z_j^f denotes that if we use the candidate path t_j^f chosen from T_f or not. w_f denotes the quantum bandwidth(or the generated qubit rate) required by S-D pair f.

$$\min \quad C_r \sum_{p\in P} x_p + C_d \sum_{p\in P} y_p$$

$$S.t. \begin{cases} \sum_{t\in T_f} z_f^t = 1, & \forall f \in F \\ z_f^t \le x_p, & \forall p \in t, \forall t \in T_f, \forall f \in F \\ \sum_{f\in F} \sum_{p\in t: t\in T_f} z_f^t \cdot w_f \cdot R(t,p) \le y_p \cdot R_0, & \forall p \in P \\ x_i \in \{0,1\} \\ y_i \in \{1,2,...\} \\ z_f^t \in \{0,1\} \end{cases} \tag{1}$$

The first set of equalities denotes each S-D pair f should select one path for quantum key distribution. The second set of inequalities means if a S-D pair f chooses a path t_j^f, then all the positions on this path should deploy a relay(x_i = 1). The third set of inequalities indicates the total secret key generation rate required on position p can not exceed its capacity.

To find candidate path set T_f for each f, firstly we construct a graph with given node set consists of endpoint set N and candidate location set P. Then we add a link between any two nodes if the physical length of link between them is less than L_{max}. After construction, we can run an algorithm on this graph to get multiple paths T_f for each S-D pair f. Here we use a recursive method, called DFSPP to search a permisssible path set T_f for each pair f. The algorithm is described in Alg.III-A Considering that there are exponential number of permissible paths in the network, we only care for the shortest path(use hop count as length) to reduce the time complexity of our algorithm. For each pair f, we take the source endpoint s and the destination endpoint d as the first

Algorithm 1 DFSPP(s,d,t,g):Depth First Search on Permissible Paths

```
if (s = d) then
    add the path to permissible path set
    return
if (t = g) then
    return
compute a node set $H(s, d)$
for each node $u \in H(s, d)$ do
    set node u as $visited$
    DFSPP(u,d,t+1,g)
    set node u as $unvisited$
```

two parameters of DFSPP. The parameter t is the current length of path, initially 0 and g is the maximun hops we could afford. DFSPP initially set g as the minimum hops from s to t in the legacy network and gradually increase it until we get a suitable path set.

Theorem 1: The quantum key distribution network control problem is NP-hard.

We can prove that the typical Steiner Minimum Tree with Minimum Steriner Point(SMT-STP) problem is a special case of our problem. Considering each pair of endpoints as a S-D pair and ignoring the quantum device cost, our problem becomes a steiner minimun tree with minimum steiner point problem. Since such a problem is NP-hard, our problem is NP-hard.

B. Problem Complexity Analysis

The QNTC problem remains challenging even when some constraints are relaxed as mentioned before. We will introduce some special cases in literature to show the universality of our problem.

Steiner Minimum Tree with Minimum Steriner Point with length constraints problem: Given a network topology G={D,E} and a set of point S, we use as few points as possible in point set D so that every point in S is connected.

Shortest Path Selection problem: Given a network topology G={D,E}, and a set of S-D pair F, we explore the shortest path for each $f \in F$. In this problem, we use hops to present the length of each path.

The main difference from the above two problem to our problem is that our problem is a joint problem of point selection and path selection. However, in steiner minimum tree problem, we only require one available path each S-D pair and in shortest path selection problem, we only search for the shortest path to use as less devices as possible. Typical algorithms for SMT never consider the path lengths for S-D pairs. But in our problem, the path length impacts the number of devices used, so we need to consider both relay selection and path selection at the same time.

This suggests that the QNTC problem is substantially harder than the SMT-MSP problem described above, and thus a different method is needed to find a tight approximation ratio. In the next section, we present a method with approximation performance guarantees.

C. Rounding-Based Deployment Algorithm for QKD Network

In this section, we propose a rounding-based deployment algorithm for the QKD network topology control problem, described in Alg.**??**. Due to the difficulty of the QNTC problem, the first step obtains the fractional solution for the relaxed QNTC problem. In the second step, we choose one feasible path for each S-D pair using randomized rounding method. Finally. we deploy both relays and devices of all candidate positions based on the paths we choose.

Algorithm 2 RQTC:Rounding-based QKD Network Topology Control

```
Step 1: Solving the Relaxed QNTC Problem
Construct a linear optimization program in Eq.13
Obtain the optimal solutions $\widetilde{z}_f^t$
Step 2: Selecting Paths for S-D pairs
Derive an integer solution $\hat{z}_f^t$ by randomized rounding
for each S-D pair $f \in F$ do
    for each feasible path $t \in T_f$ do
        if $\hat{z}_f^t = 1$ then
            Appoint a feasible path t for S-D pair f
Select quantum relays and deploy quantum devices according to path selection
```

In the first step, the algorithm constructs the relaxed linear problem of Eq.1, and solves the relaxed program Eq.13. More specifically, the QNTC problem assumes that each S-D pair perform QKD through one feasible path. By relaxing this assumption, the path of each f is permitted to be splittable. We formulate the following linear Program as LP_1.

$$\min \quad C_r \sum_{p \in P} x_p + C_d \sum_{p \in P} y_p$$

$$S.t. \begin{cases} \sum_{t \in T_f} z_f^t = 1, & \forall f \\ z_f^t \leq x_p, & \forall p \in t, \forall t, \forall f \\ \sum_{f \in F} \sum_{p \in t : t \in T_f} z_f^t \cdot w_f R(t, q) \leq y_p R_0, & \forall p \\ z_f^t \geq 0 \end{cases} \tag{2}$$

Note that the variables x_p, y_p, z_f^t are fractional in Eq.13 and we can solve it with a linear program solver in linear time. The optimal solution for this linear problem LP_1 is denoted by $\widetilde{x}_p, \widetilde{y}_p$ and $\widetilde{z}_f^t$. We use C to present the final result of cost $C_r \sum_{p \in P} x_p + C_d \sum_{p \in P} y_p$, so the optimal result is denoted by $\widetilde{\lambda}$. Since LP_1 is a relaxation of the QNTC problem, $\widetilde{\lambda}$ is a lower bound result of QNTC. Next, using the randomized rounding method, variable $\hat{z}_f^t$ is set to 1 with the probability of $\widetilde{z}_f^t$ while satisfying $\sum_{t \in T_f} z_f^t = 1, \forall f \in F$. It means that the S-D pair f selects path t if $\hat{z}_f^t = 1$.

D. Approximation Performance Analysis

We analyze the approximate performance of the proposed RQTC algorithm. We define variable α as follows:

$$\alpha = min\{\frac{\widetilde{y}_p \cdot R_0}{w_f \cdot R(t,p)}, p \in t, t \in T_f\}\} \tag{3}$$

Since RQTC is a randomized algorithm, we compute the expected relay numbers along with the quantum relays.

We first give two famous theorems for the following probability analysis.

Lemma 2: (Chernoff Bound):Given n independent variables:$x_1, x_2, ...x_n$, where $\forall x_i \in [0,1]$.Let $\mu = \mathbb{E}[\sum_{i=1}^{n} x_i]$. Then $\mathbf{Pr}[\sum_{i=1}^{n} x_i \geq (1+\epsilon)\mu] \leq e^{\frac{-\epsilon^2\mu}{2+\epsilon}}$, where ϵ is an arbitrarily positive value.

Lemma 3: (Union Bound):Given a countable set of n events:$A_1, A_2, ..., A_n$, each event A_i happens with possibility $\mathbf{Pr}(A_i)$.Then $\mathbf{Pr}(A_i \cup A_2 \cup ... \cup A_n) \leq \sum_{i=1}^{n} \mathbf{Pr}(A_i)$

Quantum Relay Number. Considering that if any path t through p is selected, $\hat{x}_p$ is set to 1.We give the defination of a random variable as follows:

Definition 1: For each $p \in P$ and each $f \in F$, a random varialbe $\xi_{p,f}$ is defined as:

$$\xi_{p,f} = \begin{cases} 1, with\ probability\ of\ \widetilde{z}_f^t \\ 0, otherwise \end{cases} \tag{4}$$

Since $\xi_{p,f}$ are independent variables, we have:

$$\begin{aligned} \mathbb{E}[\sum_{f \in F} \xi_{p,f}] &= \sum_{f \in F} \mathbb{E}[\xi_{p,f}] \\ &= \sum_{f \in F} \sum_{p \in t: t \in T_f} \widetilde{z}_f^t \\ &\leq L \cdot \widetilde{x}_p \end{aligned} \tag{5}$$

Combining Eq.5 and Lemma 2, assume ϵ is an arbitrary positive value, it follows:

$$\mathbf{Pr}[\sum_{f \in F} \frac{\xi_{p,f}}{\widetilde{x}_p} \geq (1+\epsilon)L] \leq e^{\frac{-\epsilon^2 L}{2+\epsilon}} \tag{6}$$

Now we assume that:

$$\mathbf{Pr}[\sum_{f \in F} \frac{\xi_{p,f}}{\widetilde{x}_p} \geq (1+\epsilon)L] \leq e^{\frac{-\epsilon^2 L}{2+\epsilon}} \leq \frac{\Phi}{n} \tag{7}$$

where Φ is the function of network-related variables and $\Phi \to 0$ when the network size $n = |P|$ grows. The solution for Eq.7 is:

$$\epsilon \geq \frac{log\frac{n}{\Phi} + \sqrt{log^2\frac{n}{\Phi} + 8Llog\frac{n}{\Phi}}}{2L}, n \geq 2 \tag{8}$$

We give the approximation performance as follows.

Theorem 4: The proposed RQTC algorithm achieves the approximation factor of $L \cdot \frac{3logn}{\alpha} + 3$ for quantum relay number.

Proof 5: Set $\Phi = \frac{1}{n^2}$,Eq.7 is transformed into:

$$\mathbf{Pr}[\sum_{f \in F} \frac{\xi_{p,f}}{\widetilde{x}_p} \geq (1+\epsilon)L] \leq \frac{1}{n^3}, where\ \epsilon \geq \frac{3logn}{L} + 2 \tag{9}$$

By applying Lemma 3, we have

$$\begin{aligned} &\mathbf{Pr}[\bigvee_{p \in P} \sum_{f \in F} \frac{\xi_{p,f}}{\widetilde{x}_p} \geq (1+\epsilon)L] \\ &\leq \sum_{p \in P} \mathbf{Pr}[\sum_{f \in F} \frac{\xi_{p,f}}{\widetilde{x}_p} \geq (1+\epsilon)L] \\ &\leq n \cdot \frac{1}{n^3} = \frac{1}{n^2}, \epsilon \geq \frac{3logn}{L} + 2 \end{aligned} \tag{10}$$

Apparently we have $\hat{x}_p) \leq \sum_{f \in F} \xi_{p,f}$, then $\mathbf{Pr}[\bigvee_{p \in P} \frac{\hat{x_p}}{\widetilde{x}_p} \geq (1+\epsilon)L] \leq \frac{1}{n^2}, \epsilon \geq \frac{3logn}{L} + 2$

Then the approximation factor for quantum relay number of the algorithm is $L(\epsilon+1) = L(\frac{3logn}{\alpha} + 2)$

Quantum Device Number. We first give the approximate factor of the total secret bit throughput of each relay. The first step of the algorithm will derive a fractional solution for z_f^t ,x_p and y_p for the QNTC problem by linear program. Using the randomized rounding method, for each S-D pair $f \in F$, only one path in T_f will be chosen as its default route. Thus, the key generation rate of relay p from S-D pair f is defined as a random variable $y_{p,f}$ as follows:

Definition 2: For each $p \in P$ and each $f \in F$, a random varialbe $\theta_{p,f}$ is defined as:

$$\theta_{p,f} = \begin{cases} w_f \cdot R(t,p)/R_0, with\ probability\ of\ \sum_{p \in t: t \in T_f} \widetilde{z}_f^t \\ 0, otherwise \end{cases} \tag{11}$$

According to the definition, the expected quantum device number in location p is:

$$\begin{aligned} \mathbb{E}[\sum_{f \in F} \theta_{p,f}] &= \sum_{f \in F} \mathbb{E}[\theta_{p,f}] \\ &= \sum_{f \in F} \sum_{p \in t: t \in T_f} \widetilde{z}_f^t \cdot w_f \cdot R(t,p)/R_0 \\ &\leq \widetilde{y}_p \end{aligned} \tag{12}$$

Combining Eq.12 and definition of α, we have:

$$\begin{cases} \frac{\theta_{p,f} \cdot \alpha}{\widetilde{y}_p} \in [0,1] \\ \mathbb{E}[\sum_{f \in F} \frac{\theta_{p,f} \cdot \alpha}{\widetilde{y}_p} \leq \alpha] \end{cases} \tag{13}$$

Then, by applying Lemma 2, assume ϵ is an arbitrary positive value. It follows :

$$\mathbf{Pr}[\sum_{f \in F} \frac{\theta_{p,f} \cdot \alpha}{\widetilde{y}_p} \geq (1+\epsilon)\alpha] \leq e^{\frac{-\epsilon^2 \alpha}{2+\epsilon}} \tag{14}$$

Now we assume that:

$$\mathbf{Pr}[\sum_{f \in F} \frac{\theta_{p,f}}{\widetilde{y}_p} \geq (1+\epsilon)] \leq e^{\frac{-\epsilon^2 \alpha}{2+\epsilon}} \leq \frac{\Phi}{n} \tag{15}$$

where Φ is the function of network-related variables and $\Phi \to 0$ when the network size $n = |P|$ grows. The solution for Eq.15 is:

$$\epsilon \geq \frac{log\frac{n}{\Phi} + \sqrt{log^2\frac{n}{\Phi} + 8\alpha log\frac{n}{\Phi}}}{2\alpha}, n \geq 2 \tag{16}$$

We give the approximation performance as follows.

Theorem 6: The proposed RQTC algorithm achieves the approximation factor of $\frac{3logn}{\alpha} + 3$ for quantum device numbers.

Proof 7: Set $\Phi = \frac{1}{n^2}$,Eq.15 is transformed into:

$$\mathbf{Pr}[\sum_{f \in F} \frac{\theta_{p,f}}{\widetilde{y}_p} \geq (1+\epsilon)] \leq \frac{1}{n^3}, where\ \epsilon \geq \frac{3logn}{\alpha} + 2 \tag{17}$$

By applying Lemma 3, we have

$$\begin{aligned}&\mathbf{Pr}[\bigvee_{p\in P}\sum_{f\in F}\frac{\theta_{p,f}}{\widetilde{y}_p}\geq(1+\epsilon)]\\ &\leq\sum_{p\in P}\mathbf{Pr}[\sum_{f\in F}\frac{\theta_{p,f}}{\widetilde{y}_p}\geq(1+\epsilon)]\\ &\leq n\cdot\frac{1}{n^3}=\frac{1}{n^2},\epsilon\geq\frac{3logn}{\alpha}+2\end{aligned}\tag{18}$$

Then the approximation factor for quantum device number of the algorithm is $\epsilon+1=\frac{3logn}{\alpha}+2$

Approximation Factor for the Cost of deployed QKD Network. We have discussed the approximation factor for both quantum relay and quantum device numbers. With Theorem 4 and Theorem 6, we can easily give the following result:

Theorem 8: The proposed RQTC algorithm achieves the approximation factor of $\min(L(\frac{3logn}{L}+3),\frac{3logn}{\alpha}+3)$ for total cost C.

IV. k-SPLITTABLE ROUTING IN DEPLOYED QKDN

In this section, We first discuss scenarios where the quantum key rate requirements between endpoints change dynamically. Then we give solutions in these cases and propose an approximation routing algorithms.

A. Dynamic Scenarios in QKD Network

In practice, the S-D pair set F may changes and the path width requirement w_f for each S-D pair f may changes too. When meeting these dynamic scenarios in QKD network, the number of quantum channels may exceed the capacity of a relay. At this time, we should re-select the path for every S-D pair, try to satisfy the quantum device constraint of each relay. However, considering devices are portable, when the quantum device constraints can not be satisfied anymore, we should update the quantum devices for each relay.

There are mainly three dynamic scenarios that will break constraints in QKD network. The first one is the change of S-D pair F. Considering that there are two quantum endpoints which were not communicating before want to perform quantum key distribution, we need to specify a path for them. The second is that the bandwidth required of one S-D pair f changes. Under such circumstances, the default path for f may need to be changed cause the quantum capacity of used relays may be not enough. The third dynamic scenario is the failure of quantum devices. Once the device used fails, f may needs to select another path if the quantum bandwidth is not satisfied anymore. In more serious cases, a certain relay may fail completely or become untrustworthy due to some force majeure. At this time, we need to update the route as soon as possible, otherwise it will make QKD invalid for a long time, because the time to restore the relay can not be established.

B. k-Splittalbe Routing in a QKD network

After deploying a QKD network, an efficient routing mechanism will help to improve the network performance, such as high throughput of secret bits. We studies the throughput maximization problem of QKD by k-splittable routing(k-TMQN). Given an integer k ≥ 1, we assume that each S-D pair transmit secret bits through at most k different paths. The splittable routing scheme helps to improve the QKD network throughput by increasing the utilization of the quantum devices. However, the arbitrarily splittable scheme will lower the security level and increase the management difficulty of the QKD network.

Efficient route selection depends on the current QKD network workload. We should dynamiclly update the routing schema so as to adapt the dynamic scenarios as stated before. Thus, we care for the current S-D pair set F' instead of the long-term S-D pair set F in section 3. The set F' changes accordingly with the QKD network system running. Each S-D pair $f\in F$ will transmit secret bits through at most k permissible paths from source to destination. Suppose that we can transmit secret traffic load of $\lambda\cdot w(f)$ for each S-D pair f, where λ is the throughput factor and $w(f)$ is the secret traffic demand of f. The k-TMQN problem will select at most k permissible paths for each S-D pair f, and allocate a feasible secret bit bandwidth on each path. Let $c(q)$ denote the capacity of relay q. To avoid congestion, the load on each relay should not exceed its capacity. The object is to maximize the network throughput factor λ.

We give the formulation of the k-TMQN problem as belows.

$$\max\quad\lambda$$

$$S.t.\begin{cases}\sum_{t\in T_f}z_f^t\leq k, & \forall f\in F\\ \sum_{t\in T_f}x(t)\geq\lambda\cdot w(f), & \forall f\in F\\ \sum_{f\in F}\sum_{q\in t:t\in T_f}z_f^t\cdot x(t)\cdot R(t,q)\leq c(q), & \forall q\in Q\\ z_t^f\in\{0,1\} & \forall f,t\end{cases}\tag{19}$$

Note that x(t) denotes the allocated bandwidth through each path t. The first set of inequalities ensures that each S-D pair at most select k paths. The second set of inequalities means that the achievable throughput of each S-D pair f should not be less than $\lambda\cdot w(f)$, where λ is throughput factor. The third set of inequalities represents that the total traffic load on relay q should not exceed its key generation capacity.

C. Algorithm for the k-TMQN problem

In this section, we design an algorithm for QKD k-splittable routing with quantum device capacity constraints after the QKD network topology is decided in section 3. To deal with routing challenges in a QKD, the algorithm explores a permissible path set for each flow firstly, and selects at most k permissible paths for routing between each pair of source and destination using randomized rounding mechanism in the second step. We proposed the approximate algorithm, called RSRQ(k) in the following part.

In the first step, we use a recursive method to search a permissible path set T_f for each S-D pair f.

In the second step, we constructs a linear program as a relaxation for the k-TMQN problem. As k-TMQN assumes that the traffic of each S-D pair will only go through at most k permissible paths, we formulate the following linear program LP_2 by relaxing this assumption. That is , the traffic of each f can be forwarded through a set of permissible paths arbitrarily. The equation (20) formulates this linear program.

$$\max \quad \lambda$$
$$S.t. \begin{cases} \sum_{t\in T_f} x(t) \geq \lambda \cdot w(f), & \forall f \in F \\ \sum_{f\in F}\sum_{q\in t: t\in T_f} x(t)\cdot R(t,q) \leq c(q), & \forall q \in Q \\ x(t) \geq 0, \forall t \end{cases} \tag{20}$$

The first set of inequalities denotes that the throughput of each S-D pair f should not be less than $\lambda \cdot w(f)$, where λ is the throughput factor and $w(f)$ is the secret bit rate demand between each S-D pair. The second set of inequalities ensures that the total traffic load on each relay q should not exceed its key generation capacity $c(q)$. The third set of inequalities means that the traffic on any path is non-negative.

Since LP_2 is a linear program, we can solve it in polynomial time with a linear program solver. Assume that the optimal solution of LP_2 is $\widetilde{x}$ and the result of throughput factor is denoted as $\widetilde{\lambda}$. Since LP_2 is a relaxtion of the k-TMQN problem, $\widetilde{\lambda}$ is an upper-bound of throughput factor for k-TMQN.

After solving the linear problem, we choose at most k permissible paths from $T_f^{'}$ for each S-D pair f with $x(t) > 0$ and allocate a feasible bandwidth $\hat{x}(t)$ for each selected path t. The detailed steps are described in Alg.(**??**)

The algorithm selects at most k permissible paths for each S-D pair and allocate feasible secret bandwidth of each path. For each S-D pair f, if $|T_f^k| \leq k$, we just choose these paths for f, and set $\hat{x}(t) = \widetilde{x}(t)$ accordingly. If not, we divide $|T_f^k|$ into two sets, $T_f^{'}$ and$T_f^{''}$.To construct the set $T_f^{'}$, we set $k' = k$ and $w'(f) = \widetilde{\lambda}w(f)$ initially and then select a path with $x(t) \geq \frac{w'(f)}{k'}$ and update $\hat{x}(t) = \widetilde{x}(t), w'(f) = \widetilde{\lambda}w(f) - \sum_{t\in T_f^{'}} \widetilde{x}(t), k' = k - |T_f^{'}|$ in each iteration unless we can not find any path which meets the condition. After that, the second path set $T_f^{''}$ is determined to be $T_f^k - T_f^{'}$. For each path $t \in T_f^{''}$, we use another variable $y(t) = \frac{k'\cdot\widetilde{x}(t)}{w'(f)} = \frac{k'\cdot\widetilde{x}(t)}{\sum_{t\in T_f^{''}}\widetilde{x}(t)}$ ti to denote the weight of each path. In this way, the weight of each path follows $0 < y(t) < 1$ and $\sum_{t\in T_f^{''}} y(t) = k'$. We solve the min-max knapsack problem by placing each path into k' knapsacks so as to minimize the total weight of all paths in each knapsacks. In the i_{th} knapsack, the set of paths is denoted as T_f^i, and the total weight of the i_{th} knapsacks is denoted as $z_i = \sum_{t\in T_f^i} y(t)$. At last, we choose one path from each path set T_f^i and allocate a feasible bandwidth $\hat{x}(t) = \frac{w'(f)z_i}{k'}$. Path t with has weight $y(t)$ will be chosen with probability $\frac{y(t)}{z_i}$. For the other paths in the set, we set $\hat{x}(t) = 0$.

Algorithm 3 RSRQ(k):Rounding-based k-Splitttable Routing in a QKD Network

```
Step 1: Initialization
for each S-D pair $f \in F$ do
    Calculate the permissible path set $T_f$
Step 2: Solving the Relaxed Problem
Construct a linear optimization program $LP_2$ in Eq.20
Obtain the optimal solutions $\widetilde{x}(t)$
Step 3: Selecting Paths for S-D pairs
for each S-D pair $f \in F$ do
    $k' = k$ and $w'(f) = \widetilde{\lambda}w(f)$
    while $max\{x(t)\} \geq \frac{w'(f)}{k'}$ do
        Put path $t$ to set $T_f^{'}$ with $x(t) \geq \frac{w'(f)}{k'}$
        Set $\hat{x}(t) = \widetilde{x}(t), w'(f) = \widetilde{\lambda}w(f) - \sum_{t\in T_f^{'}} \widetilde{x}(t), k' = k - |T_f^{'}|$
    $T_f^{''} = T_f^k - T_f^{'}$
    for each $t \in T_f^{''}$ do
        $y(t) = \frac{k'\cdot\widetilde{x}(t)}{w'(f)} = \frac{k'\cdot\widetilde{x}(t)}{\sum_{t\in T_f^{''}}\widetilde{x}(t)}$
    Put all paths into k' kanpsacks with min-max weight. The ith path set in knapsack i is denoted as $T_f^i$;
    for each knapsack i do
        $z_i = \sum_{t\in T_f^i} y(t)$
        Randomly choose a path $t \in T_f^i$ for S-D pair $f$ with probability $\frac{y(t)}{z_i}$
        $\hat{x}(t) = \frac{w'(f)z_i}{k'}$
```

We can give the following lemma.

Lemma 9: After random rounding selection, each S-D pair f has at most k permissible paths with throughput $\widetilde{\lambda}\cdot w(f)$.

Proof 10: If $|T_f^k| \leq k$, the algorithm simply choose those paths for f. Otherwise, we divide the set into two sets. Firstly we decide set $T_f^{'}$ in a loop operation. After that, the remaining bandwidth that needs to be provided and the number of paths that can be forwarded is decided as $w'(f) = \widetilde{\lambda}w(f) - \sum_{t\in T_f^{'}} \widetilde{x}(t)$ and $k' = k - |T_f^{'}|$, respectively. Then we choose k' paths from $T_f^{''}$, with total throughput:

$$\sum_{t\in T_f^{''}} \hat{x}(t) = \sum_{T_f^i} \frac{w'(f)z_i}{k'} = \frac{w'(f)}{k'} \sum_{t\in T_f^{''}} y(t) = w'(f) \tag{21}$$

So the total throughput of paths in both $T_f^{'}$ and $T_f^{''}$ is:

$$\sum_{t\in T_f^{''}} \hat{x}(t) + \sum_{t\in T_f^{'}} \hat{x}(t) = w'(f) + \sum_{t\in T_f^{'}} \widetilde{x}(t) = \widetilde{\lambda}\cdot w(f). \tag{22}$$

D. Algorithm Performance Analysis

We analyze the approximate performance of the proposed RSRQ(k) algorithm in this section. Like the previous proof, we calculate the expected key generation rate on each quantum relay and bound the probability that the key generation capacity will be violated. After solving the linear program

LP_2, we get a fractional solution of the relaxed k-TMQN problem. By randomized rounding method, for each path $t \in T_f$, its required key generation rate on relay q is denoted as a random variable $\theta_{f,t,q}$. Thus, the required key generation rate on relay q for S-D pair f is defined as a random varialbe $\theta_{f,q} = \sum_{t \in T_f} \theta_{f,t,q}$. For each S-D pair f, if $t \in T_f^{'}$, then $\theta_{f,t,q} = \widetilde{x}(t)R(t,q)$. If $t \in T_f^{''}$, we have expected value of $\theta_{f,t,q}$, that is, $\mathbb{E}[\theta_{f,t,q}] = \frac{y(t)}{z_i} \cdot \frac{w'(f)z_i}{k'} \cdot R(t,q) = \frac{w'(f)y(t)}{k'} \cdot R(t,q) = \widetilde{x}(t) \cdot R(t,q)$. We also define $\theta_{f,q}^{''} = \sum_{t \in T_f^{''}} \theta_{f,t,q}$. Obviously, $\mathbb{E}[\theta_{f,q}^{''}] = \sum_{q \in t: t \in T_f^{''}} \widetilde{x}(t) \cdot R(t,q)$. δ is denoted as $max\{\sum_{f \in F}[\sum_{q \in t: t \in T_f^{''}} \widetilde{x}(t) \cdot R(t,q)]/c(q), \forall q \in Q\}$. Obviously $0 \leq \delta \leq 1$. The expected secret key generation rate on each relay is:

$$
\begin{aligned}
\mathbb{E}[\sum_{f \in F} \theta_{f,q}^{''}] &= \sum_{f \in F} \mathbb{E}[\theta_{f,q}^{''}] \\
&= \sum_{f \in F} \sum_{q \in t: t \in T_f^{''}} \widetilde{x}(t) \cdot R(t,q) \\
&\leq \delta \cdot c(q) \leq c(q)
\end{aligned} \tag{23}
$$

We also have the following inequalities:

$$
\begin{cases}
\frac{\theta_{f,q}^{''}}{\delta \cdot c(q)} \in [0,1] \\
\mathbb{E}[\sum_{f \in F} \frac{\theta_{f,q}^{''}}{\delta \cdot c(q)}] \leq 1
\end{cases} \tag{24}
$$

We would like to expect the total secret key generation rate of each relay q wouldn't exceed the key generation capacity $\delta \cdot c(q)$ by a factor of 1+ϵ, where ϵ is an adjustable parameter.

$$
\mathbf{Pr}[\bigvee_{q \in Q} \sum_{f \in F} \frac{\theta_{f,t}^{''}}{\delta \cdot c(q)} \geq (1+\epsilon)] \leq \Phi \tag{25}
$$

where Φ is the network-related function(such as the number of quantum relays N, number of links, etc) and $\Phi \to 0$ when the QKD network size grows. By applying Lemma 3, we have the relaxation of Eq.(25):

$$
\begin{aligned}
&\mathbf{Pr}[\bigvee_{q \in Q} \sum_{f \in F} \frac{\theta_{f,t}^{''}}{\delta \cdot c(q)} \geq (1+\epsilon)] \\
&\leq \sum_{q \in Q} \mathbf{Pr}[\sum_{f \in F} \frac{\theta_{f,t}^{''}}{c(q)} \geq \delta \cdot (1+\epsilon)] \leq \Phi
\end{aligned} \tag{26}
$$

To satisfy the Eq.(IV-D), we could have the following inequality:

$$
\mathbf{Pr}[\sum_{f \in F} \frac{\theta_{f,t}^{''}}{\delta \cdot c(q)} \geq (1+\epsilon)] \leq e^{-\frac{\epsilon^2}{2+\epsilon}} \leq \frac{\Phi}{N} \tag{27}
$$

where N is the number of quantum relays $|Q|$in deployed QKD network. Then we can give the following lemma and corresponding proof.

Lemma 11: The proposed rounding-based algorithm guarantees that the total key generation rate on any relay will not exceed the key generation capacity of the fractional solution by a factor of $\delta(3 + 2logN) + 1$ with high probability.

Proof 12: Set $\epsilon = 2 + 2logN$. By applying lemma 2, it follows:

$$
\begin{aligned}
&\mathbf{Pr}[\sum_{f \in F} \frac{\theta_{f,t}^{''}}{c(q)} \geq \delta \cdot (3 + 2logN)] \\
&\leq e^{-\frac{(2logN+2)^2}{4+2logN}} \\
&\leq e^{-2logN} = \frac{1}{N^2}
\end{aligned} \tag{28}
$$

Combining both path set $T_f^{'}$ and $T_f^{''}$, we have

$$
\begin{aligned}
&\mathbf{Pr}[\sum_{f \in F} \frac{\theta_{f,t}}{(q)} \geq \delta \cdot (3 + logN) + 1] \\
&\leq \mathbf{Pr}[\sum_{f \in F} \frac{\theta_{f,t}^{''}}{c(q)} \geq \delta \cdot (3 + 2logN)] \\
&\leq \frac{1}{N^2}
\end{aligned} \tag{29}
$$

If we set $\Phi = \frac{1}{N}$, Eq.(25) is guaranteeed with $\alpha = 2{+}2logN$ and $\Phi = \frac{1}{N}$

Like the proof of quantum key generation capacity constraints, we can give the approximation factor of λ. Let λ' be the final result of our algorithm. According to lemma 11, with respect to the key generation capacity constraints, we obtain the following inequality with high probability.

$$
\frac{\lambda'}{\widetilde{\lambda}} \geq \frac{1}{\delta \cdot (3 + 2logN) + 1} \tag{30}
$$

As a result, we have the theorem:

Theorem 13: The proposed RSRQ(k) algorithm archives approximation factor of $\frac{1}{\delta \cdot (3+2logN)+1}$ for k-TMQN problem, where N is the number of quantum relays of the deployed QKD network and $0 \leq \delta \leq 1$.

E. Discussion

We discusss the approximation factor of our algorithm in two special cases. If $k = 1$, that's to say, we only perform QKD through one path for each S-D pair. In this case, it follows that $\sum_{t \in T_f^{''}} \widetilde{x}(t) = \widetilde{\lambda} w(f)$ and $\delta = 1$. Thus, our algorithm achives approximate performance of $O(\frac{1}{logN})$. If we can perform QKD through arbitrarily paths, it follows $\sum_{t \in T_f^{''}} \widetilde{x}(t) = 0$ and $\delta = 0$. In this case, our algorithm can get the optimal result for arbitrarily splittable routing.

However, there are chances that the quantum devices in some relays cannot satisfy all the requirements of S-D pairs.That is, the thoughput factor $\lambda' \leq 1$. In such situations, we do need to update the quantum devices in those relays. The addition of quantum devices takes a relatively long time, Besides, we can incrementally update the quantum devices in each relay at a regular time to cope with the increased demand for quantum secret communications in the QKD network. For example, we can update the devices in each relay monthly, based on previous data in recent month.

V. Performance Evaluation

VI. Related Works

VII. Conclusion

References

[1] J. Y. Yen, “Finding the K Shortest Loopless Paths in a Network,” *Management Science*, vol. 17, no. 11, pp. 712–716, July 1971.