applied sciences

MDPI

Article

A Dynamic-Routing Algorithm Based on a Virtual Quantum Key Distribution Network

Lin Bi [1,2], Minghui Miao [1,2,*] and Xiaoqiang Di [2,3]

[1] School of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China; bilin7080@163.com
[2] Jilin Province Key Laboratory of Network and Information Security, Changchun 130022, China; dixiaoqiang@cust.edu.cn
[3] Information Center of Changchun University of Science and Technology, Changchun 130022, China
* Correspondence: mmh18702258900@163.com

Abstract: Quantum key distribution (QKD) is an encrypted communication technique based on the principles of quantum mechanics that ensures communication security by exploiting the properties of quantum states. Currently, the transmission efficiency of the QKD system is low. Trusted relay technology is used to solve this problem and achieve long-distance transmission. However, trusted relaying alone cannot decrypt the issues of poor link stability and the low utilization of key resources. To further optimize the system performance, we propose a dynamic routing algorithm. One of the improvement schemes includes the following: firstly, an adjustable-size quantum key pool (QKP) is designed, which can dynamically adjust the size of the refreshing pool according to the actual demand. Secondly, the utilization of key resources is improved by using the residual quantum key model to dynamically obtain the remaining key amount in the QKP and set the key amount threshold. We calculate the link-blocking probability and track the blocking intensity and blocking entry by combining the Poisson process, thus realizing the evaluation of the link stability. Finally, the number of remaining keys in the QKP and the link-blocking probability combine with the random wandering model as the basis of the route selection for the QKD dynamic routing algorithm to achieve efficient key path selection. We validated the algorithm by comparing it with other algorithms on the Mininet simulation platform, and the algorithm proved to have a better performance in terms of congestion avoidance, delay reduction, and improved QKD efficiency. This scheme provides a novel and efficient way to solve the problems in existing QKD systems. It effectively improves the transmission efficiency and strengthens the system's security by dynamically obtaining the critical volume, accurately evaluating the link state, and selecting the optimal critical path.

Keywords: quantum key distribution; adjustable-size key pool; remaining key quantity; link-blocking probability; dynamic routing algorithm

Citation: Bi, L.; Miao, M.; Di, X. A Dynamic-Routing Algorithm Based on a Virtual Quantum Key Distribution Network. *Appl. Sci.* **2023**, *13*, 8690. https://doi.org/10.3390/app13158690

Academic Editor: Marco Genovese

Received: 18 June 2023
Revised: 19 July 2023
Accepted: 21 July 2023
Published: 27 July 2023

1. Introduction

Quantum key distribution (QKD) is an encryption technology based on quantum communication, the purpose of which is to ensure that neither party has to worry about eavesdropping or tampering when exchanging keys [1–3]. It includes the QKD network architecture, quantum key pool (QKP), and quantum key relays to form a complete and secure quantum communication-based cryptosystem [4–7]. Quantum algorithms are exponentially faster than other algorithms because they can exploit properties such as quantum superposition and quantum entanglement to achieve parallel computation and information processing on some specific issues, thereby achieving exponential acceleration. However, there are still challenges to be addressed in this area to improve security and efficiency further. To this end, researchers have proposed various QKD routing algorithms for establishing reliable information transmission and efficient network topologies [8].

In domestic and international research, in 2014, Wang Xuan [9] proposed a QKD routing algorithm based on quality of service and experimentally validated it using software-defined networking (SDN) technology. In the same year, a novel QKD routing algorithm based on graph theory was proposed in the literature [10], which achieved fast routing via the construction of a topology graph, aiming to reduce communication delays and improve the network performance. In addition, the authors of reference [11] investigated an approach based on hierarchical routing algorithms to optimize packet transmission paths and reduce communication delays and critical resource consumption by improving routing algorithms. In 2016, the authors of reference [12] proposed a routing scheme combining a hybrid distance vector and shortest path algorithms to find feasible paths faster by optimizing the network topology and path selection. However, the literature needs to adequately discuss the possible delays between network nodes, which may affect the actual operation of the network. Furthermore, in 2020, the authors of reference [13] proposed an improved QKD routing algorithm that employs reinforcement learning ideas and applies them to routing decisions to improve the security and performance of the system. However, the literature did not compare and analyze it with other existing QKD routing algorithms; thus, its advantages and limitations among similar algorithms could not be determined. In 2023, the authors of reference [14] investigated the use of QKD routing algorithms to achieve security and privacy in resource allocation in resource-sharing environments, such as cloud computing. The study proposed a resource-allocation framework that can effectively support the distribution and management of resources. However, the study did not consider alternative key management strategies and poses security risks. Although mature QKD routing methods from various perspectives are available, further improvements and optimizations are needed in practical applications, especially in terms of cost reduction, improved communication quality, and critical distribution efficiency.

To solve the critical shortage problem, we designed a resizable QKP scheme through QKP technology. The system can adjust the key pool size according to the demand, reducing the cost and mitigating the risk of key shortage. Secondly, this paper uses Bayesian formulae to calculate the link-blocking probability based on the node-blocking chance. It uses a new dynamic routing algorithm that combines the remaining key quantity and the link-blocking opportunity to improve the communication quality and key transmission efficiency. The quantum bit error rate (QBER) is used to assess the quantum channel quality, the quantum bit emission rate is related to the distribution efficiency, and the quantum key generation rate reflects the efficiency and availability of the distribution process. A lower key-blocking probability indicates a higher distribution success rate, while a higher number of remaining keys means sufficient resources are available for subsequent key generation. By optimizing these parameters, a more efficient and reliable QKD can be achieved. In addition, with the dynamic routing algorithm, we can maximize the use of available essential resources and reduce the costs of network deployment and operation. Finally, we performed a theoretical analysis and an experimental comparison of the proposed method with the algorithm, and the results show that the algorithm can significantly improve the quality of the QKD system, save costs, and improve resource utilization relative to other algorithms.

The rest of this article is organized as follows: Section 2 first describes the SDN-based QKD network framework and trusted relay QKD network. Then, it introduces the QKP structure, the steps to design a tunable QKP, and the remaining key amount in the QKP. Finally, the probability of link blocking and the threshold of link blocking are calculated. Section 3 proposes a dynamic routing algorithm that combines the remaining key amount, the likelihood of link blocking, and the evaluation metrics for the algorithm. In Section 4, the experimental platform is introduced, and the simulation results of the algorithm comparison are analyzed. Finally, Section 5 summarizes the article.

2. Technical Foundation

2.1. QKD Network Framework Based on SDN

Combining SDN and QKD can mitigate the deployment difficulties of QKD networks. This is because SDN-based QKD networks offer the advantages of reduced costs, simplified management, and increased flexibility. Traditionally, complex quantum physical equipment must be deployed at each node in a QKD network, which incurs high costs and technical complexity [15]. However, in an SDN network, the control plane has been abstracted, and operations such as routing and forwarding can be implemented through software. At the same time, utilizing the flexibility of SDN, the key distribution system can be more easily managed and maintained, thereby improving its reliability and efficiency. A QKD network framework based on SDN is designed using a four-layer architecture [16–20]; each layer is defined as represented in Figure 1.

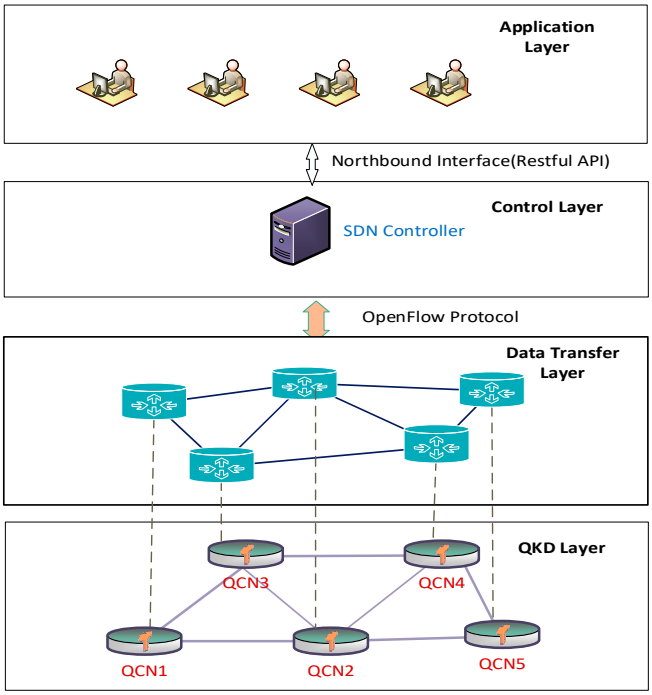

Figure 1. SDN-based QKD network architecture.

Application Layer: This layer manages the application programs in the QKD network. It can communicate with the network controller to request the generation, update, or destruction of keys.

Control layer: This layer is responsible for resource allocation and key management in the QKD network. It receives application requests and converts them into configuration commands for the underlying network. At this layer's core, the core of the entire SDN-based QKD architecture, the controller can obtain information about network topology and links, providing a basis for implementing dynamic routing algorithms. In addition, it can collect and analyze network and link status information supplied by data layer devices and grasp network topology, traffic load conditions, and other valuable data [21]. When problems occur, it can diagnose and take appropriate measures promptly to ensure the reliability and security of the network.

Data Transfer Layer: responsible for the transmission of keys and related information. In the SDN architecture, the data plane consists of programmable network switches and optical devices connected to the controller and processes transmission tasks according to its instructions. The data transfer layer needs to allocate independent channels for different users or applications based on the security policies of the control layer.

Quantum Layer: The QKD devices are located in this layer, and they generate keys between each other through specific QKD protocol BB84 [22,23], completing the negotiation of quantum keys. The newly developed quantum keys are recorded in the control layer with corresponding session information. On the other hand, the generated keys can be distributed to trusted target devices or objects, such as digital signature algorithms used for security authentication [24]. At the same time, this layer is responsible for tracking the usage of each key, including information such as validity period and consumption rate. When a key expires or reaches the predetermined consumption threshold, a message is sent to the control layer to notify it that the renegotiation or updating of the key information is required [25].

2.2. Relay-Based Networks

There is a practical distance limitation for QKD systems, i.e., key transmission distance. Moreover, the point-to-point QKD system is only suitable for communication between two nodes [26,27]. To solve these problems, trusted relay schemes are introduced. The advantages of trusted relaying are that longer communication distances can be achieved, complex topologies can be built, and high reliability is available. Therefore, trusted relaying schemes can be applied to various scenarios to meet different application needs [28–30]. In addition, QKD networks based on trusted relays employ a variety of security mechanisms to ensure the confidentiality and integrity of keys. These include using quantum channels to transmit quantum bits for key security and authentication mechanisms to verify the legitimacy of the communicating parties. QKD networks based on trusted relays extend the communication distance through trusted quantum relays and adopt appropriate security mechanisms to prevent attacks and eavesdropping [31].

In communication in trusted relays, both parties use the quantum channel for QKD protocol to generate a shared quantum key. The key transmits to a trusted relay node, which authenticates, processes, and forwards to the other communicating party using classical encryption algorithms. Once the receiver receives the key, it is again confirmed and verified using the QKD protocol to ensure the correctness and security of the key [32]. In this way, a secure communication connection establishes between the communicating parties. The trusted relay method (TRM) is a network security protocol [33] used to enhance communications' security and privacy. As users in different geographical locations, it allows Alice and Bob to establish secure communication even when there are intermediate nodes or attackers between the communicating entities. They can resort to trusted relay methods to address the problem of a potential attacker, Eve, who may listen in or interfere with the communication. The communication process is shown in Figure 2.

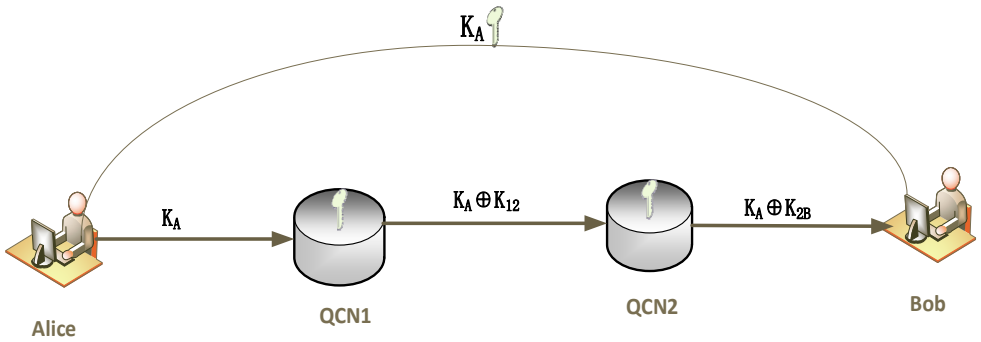

Figure 2. Key relay process in QKD network communication based on trustworthy relays.

Step 1: Alice transmits the quantum state to the relay node through the quantum channel, i.e., Alice and node 1 share the communication key K_A.

Step 2: Node 1 and node 2 share the link key K_{12}, and node 1 calculates $K_A \oplus K_{12}$ by One-Time Password (QTP) encryption, encodes and stores the measurement results, and transmits them to relay node 2.

Step 3: Node 2 receives $K_A \oplus K_{12}$, calculates $(K_A \oplus K_{12}) \oplus K_{12}$ to obtain K_A, and then calculates $K_A \oplus K_{2B}$ to transmit to the destination node Bob.

Step 4: Bob receives $K_A \oplus K_{2B}$ and computes $(K_A \oplus K_{2B}) \oplus K_{2B}$ to obtain the shared communication key K_A with Alice.

After steps (1–4), Alice and Bob can obtain the same communication key K_A. When Alice and Bob transmit data in the classical network, they can encrypt the shared data information using communication key K_A.

2.3. Design of QKP

Quantum keys are a very valuable resource and to manage and store them, a QKP is usually configured between each pair of neighboring nodes. i.e., a QKP is a repository containing multiple pre-generated quantum keys [34,35]. The sender and receiver use these keys to encrypt and decrypt the communication data during the QKD process.

Designing an adjustable QKP size requires a combination of factors. Firstly, there is a trade-off between security and availability. Larger key pools can meet more demands but also increase the security risk and resource management burden. Therefore, there is a need to balance security and availability to ensure that the critical pool is large enough and that the keys are effectively managed and protected [36]. Secondly, the speed of key generation is an important consideration. Generating quantum keys takes time and is algorithm and hardware-device-dependent. Larger key pools may take longer to generate enough keys. Therefore, when sizing the essential collection, the time required to create new keys must be considered to ensure that the necessary keys are available promptly. In addition, the rate of key consumption needs to be considered. Smaller key pools tend to be depleted, requiring frequent generation of new keys. Conversely, larger key pools reduce the need to generate new keys but may also increase the risk of crucial obsolescence. Therefore, there is a trade-off between the vital consumption rate and the key pool size to ensure that keys are always available and valid.

In summary, resizing a QKP is a complex problem that requires a combination of security, availability, key generation speed, and key consumption rate. Trade-offs and evaluations need to be made when making decisions to meet specific requirements and ensure system stability and security.

2.3.1. Definition of Parameters

The dynamically resizable QKP can generate and store keys and then resize them according to actual demand, thus avoiding the problem of wasted and insufficient keys. If there is currently more demand, then the size of the QKP needs to be increased, with more connections being transferred in parallel to satisfy more requests. Conversely, if demand decreases, then the size of the QKP can be reduced.

Definition 1: *Link key generation rate. The link key generation rate is the number of links i to link j keys established through the key negotiation protocol in a given period using quantum communication methods. Its equation can be expressed as.*

$$R_{ij} = \frac{N_{num}}{T_i} \tag{1}$$

where R_{ij} denotes the link key generation rate, N_{num} denotes the number of successfully negotiated link keys, and T_i denotes the total time required to negotiate the link for the key.

Definition 2: *Q_{ij} denotes the rate of key consumption from link i to link j, i.e., the number of keys consumed per unit of time, which can be calculated by Equation (2). Refers to the number of keys generated in the QKD network, and t denotes the real time used to create these keys.*

$$Q_{ij} = \frac{F_t}{t_i} \tag{2}$$

2.3.2. Design Steps

Assuming a QKP of size M_0, which needs to be enlarged or reduced to M_1, the following design steps are shown.

First, calculate the average utilization rate of the current QKP U. It is known that the generation rate of quantum keys is R_{ij} and the consumption rate of quantum keys is Q_{ij}. Then, the relationship between the three can be expressed by the following equation:

$$U = \frac{Q_{ij}}{R_{ij} + M_0} \tag{3}$$

The average usage of a QKP is equal to the ratio of the number of keys consumed from the QKP per unit of time to the sum of the number of new keys generated from the QKP per unit of time and the current QKP size. The size M_1 of the target pool is then calculated, giving:

$$M_1 = \frac{M_0}{U} \tag{4}$$

where U denotes the average number of times each QKP key is used. Therefore, the target key pool size needs to be set to the original key pool size divided by the average utilization to maintain a given average utilization. This ensures that the key pool can always meet the expected demand.

Finally, if $M_1 > M_0$ current capacity, the pool size is increased, and if $M_1 < M_0$ current capacity, the pool size is reduced.

Figure 3 shows a schematic diagram of the structure of the QKP, which consists of three parts, namely:

a. Quantum key sending and receiving device: responsible for generating and sending quantum keys and receiving and measuring keys sent from other devices.
b. The key storage device: responsible for storing the distributed quantum keys for subsequent use.
c. Quantum encryption application: uses the generated and stored keys to encrypt and decrypt the information to be transmitted. In this system, the QKP is deployed inside two communication nodes, and both relaying and encryption processing are implemented through a key control server that increases or decreases the capacity of the QKP according to the amount of service demand.

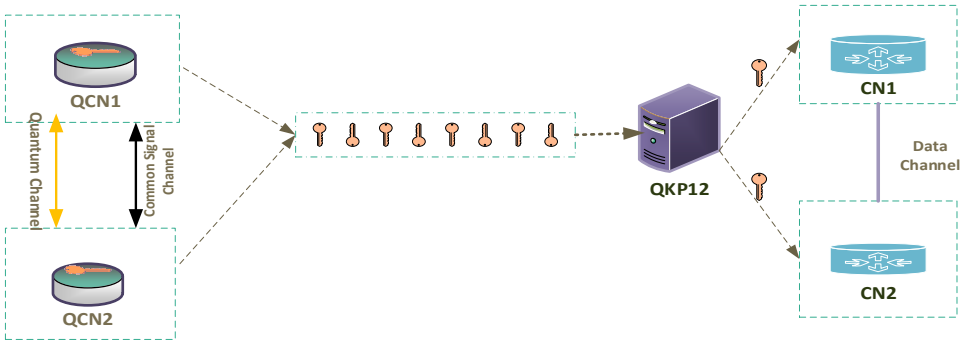

Figure 3. QKD network architecture.

Since most QKD systems spend a significant amount of time resources generating and storing quantum keys but only use them for a specific period, accumulating too many keys at other time nodes wastes limited resources [37]. By dynamically scaling up or down the capacity of a QKP, the system can more quickly change its power and respond to changing workload demands as needed. A fixed number of QKD systems may not ideally balance the relationship between key generation security and resource consumption. Scaling down the capacity of the QKP as required ensures a high level of security with minimal resources.

2.3.3. Remaining Key Volume

The number of remaining keys in a QKP can be used to evaluate the number of quantum keys available in the network, thus helping to select a better quantum communication path [38,39]. To maximize the saving of quantum keys and enable it to support as many users as possible, improve communication efficiency. Therefore, we defined the following equation to calculate the number of keys remaining in QKP.

Assume that K_0 denotes the initial amount of keys in the QKP, R_{ij} represents the rate of quantum key generation, and Q_{ij} indicates the rate of quantum key consumption. Introducing a mathematical calculus model to describe the change in the number of keys provides a more accurate and comprehensive analysis to predict the trend in the number of keys. Where $\int_{t_0}^{t} R_{ij} - Q_{ij}$ denotes the cumulative number of keys consumed in the interval from the time t_0 to t, the amount of keys K_{ij}^{t} in the remaining essential pool is indicated as follows:

$$K_{ij}^{t} = K_0 + \int_{t_0}^{t} R_{ij} - Q_{ij} \tag{5}$$

In QKD, by setting the critical value of the remaining quantum key in the QKP, the system's security and operational efficiency can be improved while preventing unauthorized access and attacks. Assuming that the critical value of the remaining key volume is K_{min}, this can be expressed using the following equation:

$$K_{min} = \frac{R_{ij} \times T}{Q_{ij}} \tag{6}$$

where T denotes the time interval between quantum key generation and consumption, the meaning of the formula is that the critical value of the remaining key amount is equal to the ratio of the quantum key generation rate to the consumption rate multiplied by the time interval. K_{min} can be used to determine whether the quantum key needs to be regenerated, and when the remaining essential amount K_{ij}^{t} is lower than the critical value K_{min}, we can trigger the regeneration process of the quantum key to ensure that there are enough remaining keys available before the critical distribution completes.

2.4. Blocking Probabilities and Thresholds

In QKD systems, link nodes are an essential component. This is because link nodes typically connect multiple quantum devices to exchange and process quantum information. And in the QKD framework, trusted relays and QKP link nodes are the basic units that form the quantum communication network. At the same time, QKP needs link nodes to ensure that the transmission and processing of quantum states during key distribution can be carried out smoothly. In summary, the contribution of link nodes to the QKD dynamic routing algorithm is to optimize the routing in the QKD network to improve the security and efficiency of the system. Figure 4 shows the link node topology.

Link nodes play an essential role in quantum communication networks, including quantum state transmission, quantum entanglement distribution, quantum channel establishment, quantum relaying, and error correction [40]. They must maintain the coherence and accuracy of the quantum states and ensure the reliable transmission of entangled quantum states. Link nodes must also establish dedicated quantum channels, employing appropriate quantum error correction and distribution protocols to resist noise and loss. In addition, link nodes can act as quantum relay stations, extending the communication distance and connecting multiple nodes. They also need to implement quantum error correction techniques to improve the reliability and fault tolerance of the overall network.

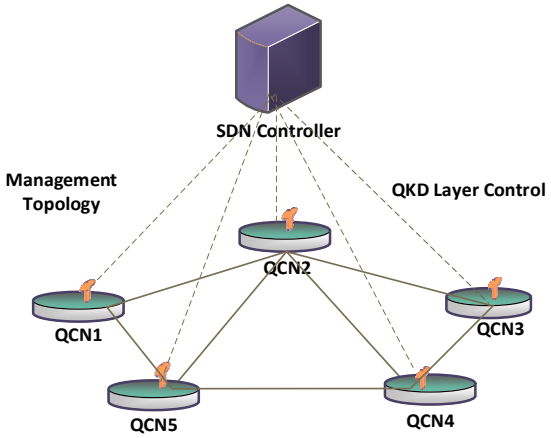

Figure 4. Link node structure diagram.

If a node blocks or crashes, it can affect the stability and availability of the entire system. Therefore, selecting nodes with a low probability of blocking can reduce the likelihood of system failure, increase system throughput and allow the system to better handle large volumes of requests and data. If specific nodes are prone to blocking, this may lead to an unbalanced load on the system, thus reducing its flexibility. Nodes in a QKD network are prone to blocking, usually because they have much higher incoming and outgoing traffic than other nodes. The blocking probability of each node can be calculated using the following formula:

$$P_i = \frac{X_i}{\sum_{j=1}^{n} X_j} \tag{7}$$

where P_i denotes the blocking probability of the node, and X_i represents that node's incoming and outgoing traffic. Using this formula, each node's blocking probability can be obtained to determine which nodes have a low blocking chance. The nodes with low blocking probability can be prioritized when making routing decisions, reducing network delays and blocking.

Suppose we know the blocking probability of a node in a network to be P_i and require the link probability P_{ij} from that node to another node, which can be calculated using the conditional probability formula:

$$P_{ij}(A|B) = \frac{P(A \cap B)}{P(B)} \tag{8}$$

$P_{\mathrm{ij}}(A|B)$ indicates the probability of event A under the premise of event B. For the problem in this paper, we can define event B as "all paths from the source node to the target node", and then $P(B)$ is the sum of the probabilities of all links from the source node to the target node. While event A can be defined as "a specific path from the source node to the target node", then $P(A \cap B)$ is the probability of the specific link.

In order to track the blocking level more accurately to ensure reliability and continuity of communication, we need to make the following assumptions: link-blocking events are independent, and their occurrence conforms to a Poisson process, i.e., the arrival times of link-blocking events follow an exponential distribution. Where the frequency of link-blocking events is λ, i.e., the average number of times a link-blocking event occurs per unit of time. The probability mass function of the Poisson distribution can be expressed as:

$$P(X = m) = \left(e^{\wedge}(-\lambda) \times \lambda^{\wedge} m\right)/m! \tag{9}$$

X denotes the number of link-blocking events per unit of time, and m represents the number of specific events. We can use the link-blocking probability to estimate the

likelihood that at least one link-blocking occurs per unit of time. Designated as $P(LB)$, it can be calculated as:

$$P(LB) = 1 - P(X = 0) = 1 - e^{\wedge}(-\lambda) \tag{10}$$

We can use the probability mass function $P(X = m)$ to calculate the probability of having 0 link blocks in unit time, namely $P(X = 0)$. This probability can be expressed as the $\hat{e}(-\lambda)$. Next, by calculating the complement of $P(X = 0)$, namely $1 - P(X = 0)$, the probability of at least one link block per unit of time is denoted as $P(LB)$. This probability can be expressed as $1 - \hat{e}(-\lambda)$.

In conclusion, we can estimate the probability $P(LB)$ of at least one link blocking per unit of time by combining the nature of the likelihood of node blocking, which can be used to evaluate the network performance and reliability and help in network planning, provisioning, and troubleshooting. Specifically, if $P(LB)$ is high, it indicates that the link-blocking event is more likely, and the availability and performance of the network may be affected. Measures can be considered to improve network performance and reliability. In addition, by setting a reasonable $P(LB)$ target value, it can be used as an indicator of network performance measurement and compared with the actual link-blocking situation to monitor and evaluate the quality of network services.

Also, it is important to note that the upper limit of the link's blocking probability, i.e., the link-blocking probability threshold, needs to be calculated based on the link capacity and load situation. Assuming a link capacity of C (the maximum number of quantum bits a communication link can transmit, i.e., the bandwidth.), the average arrival rate is ε and the average service time is $\frac{1}{\mu}$. The following equation can calculate the threshold of link blocking probability P_{max}:

$$P_{\max} = \frac{\varepsilon/C}{\frac{1}{\mu} + \lambda/C} \tag{11}$$

For example, assuming a link capacity of 1 Gbit/s, an average arrival rate of 500 Mbit/s, and an average service time of 10 ms in a QKD network, the threshold for the link blocking probability can be calculated using the following equation:

P_{max} = (500 Mbit/s/1 Gbit/s)/(1/10 ms + 500 Mbit/s/1 Gbit/s) = 0.00005. Therefore, the threshold value for the probability of link blocking is 0.005%. If the link-blocking probability exceeds this threshold, the measure must be taken to optimize network performance.

3. Routing Algorithms

3.1. Related Definitions

Definition 3: *Link weight calculation formula. The QKD dynamic routing algorithm uses weight to determine the optimal path in the network. The weight is usually expressed as a numerical value representing the path quality from the source node to the destination node. In a QKD network, each node periodically sends packets to its neighboring nodes to check the connectivity between them and exchange routing and weight information.*

In traditional networks, routing is usually selected based on distance, bandwidth and topology. In a QKD network, however, due to limited quantum key resources, there is a low utilization of network resources and uneven link load, etc. Therefore, the QKD routing algorithm evaluates the link weights and selects the best path based on the remaining quantum key resources on the link and the link congestion. Suppose there are n optional paths in the QKD dynamic routing algorithm, P_{ij} link blocking probability, and $K_{\mathrm{ij}}^{\mathrm{t}}$ the number of keys in the QKP, where the coefficient factors α and β are used to adjust the weights of the remaining quantum keys and link blocking probability and to determine their relative importance on a case-by-case basis. Thus, the values of these two factors can be flexibly adjusted according to actual requirements. W_i is denoted as the weight of the selected link:

$$W_i = \alpha \times K_{ij}^{t} + \beta \times P_{ij} \tag{12}$$

Firstly, since the calculation of link weights involves a linear combination of the remaining key quantity K_{ij}^{t} and the link-blocking probability P_{ij}, both α and β must be real numbers. Additionally, the values of α and β must satisfy the following conditions:

$$\begin{array}{l} 0 < \alpha < 1 \\ 0 < \beta < 1 \\ \alpha + \beta = 1 \end{array} \tag{13}$$

Definition 4: *Random walk modeling. The state of each node in the QKD network as a random variable, X_t for the node of the tth time step, assuming* $X\_t = (x\_\hat{t}1, x\_\hat{t}2, \ldots x\_\hat{t}n)$, *where n is the number of nodes. State transition probability matrix: defines the transition probability matrix P between node states, where* $P[i, j]$ *represents the transition probability from form i to state j. Value function for dynamic programming modeling: define* $V(X\_t)$ *represents the optimal target function value at time t state X_t. The state transfer equation according to the state transition probability matrix P and the random walk V value function is:*

$$V(X\_t) = max\{W(X\_t, u) + \sum [P(X\_t, u, X\_\{t+1\} \times V(X\_\{t+1\}))]\} \tag{14}$$

where $W(X\_t, u)$ *represents the gain obtained from selecting operation u under state X_t, and* $P(X\_t, u, X\_\{t+1\})$ *indicates the probability of transferring from state X_t to state* $X\_\{t+1\}$ *after operation u.*

The QKD dynamic routing algorithm uses the residual quantum key and link-blocking probability as weights and uses the value function $V(X\_t)$ to optimize resource allocation and routing.

The specific steps are described as follows:

1. At initial time t, the algorithm obtains the current network state X_t, including the remaining quantum key amount of each node and the blocking probability of the link.
2. The algorithm uses the value function $V(X\_t)$ to calculate the corresponding expected payoff of each possible operation, that is, through the iterative calculation of all possible operations u.
3. The algorithm selects the operation u with the maximum expected payoff and updates the network state to the state $X\_\{t+1\}$ in the next moment.
4. Repeat steps 2 and 3 until the algorithm converges or reaches the specified number of iterations.

3.2. Routing Algorithm Design

In the design of the trusted QKD network and the QKD routing algorithm, this paper sets the quantum key pool with adjustable size, that is, updates the size of the QKP corresponding to each link according to the link status and requirements and establishes the remaining quantum key mathematical model to ensure that there is always enough key to be used in the quantum key pool. Secondly, this paper uses the random walk model to monitor the blocking degree of each node, collects the blocking information of each node in the network, and estimates the link-blocking probability. The continuity and availability of QKD communication thus improves the performance of the QKD network. The procedure of the routing algorithm is shown in Figure 5.

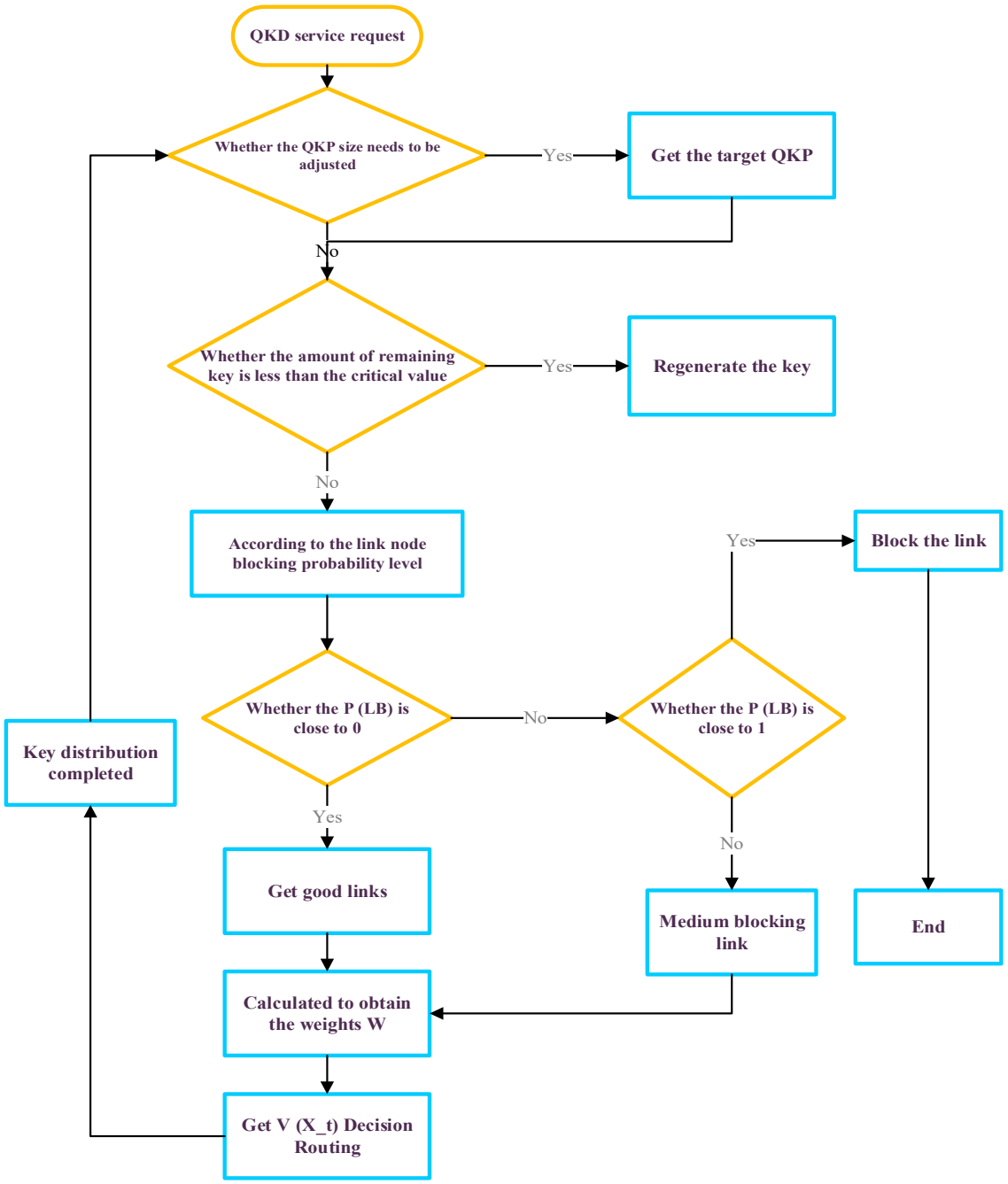

Figure 5. Algorithm optimization design flowchart.

3.3. Algorithm Evaluation Indicators

A comparison simulation with the classical Open Shortest Path First routing algorithm (OSPF) and the routing algorithm using the residual key-path-weighted routing algorithm (RKP) is carried out to more comprehensively evaluate and verify the performance of the algorithm proposed in this paper. Taking into account the QKD network characteristics, routing rules, and network quality, the following three evaluation metrics are designed:

(1) Average key utilization. Average critical consumption refers to the resource consumption required to establish and maintain a security key in a dynamic QKD routing method. Dynamic QKD routing methods can utilize resources more efficiently and achieve better system performance by reducing the average key consumption. It reflects the degree to which the allocated vital resources are used within a certain period

and then reflects whether the allocation and management of the key are reasonable. The calculation representation is shown in Formula (15):

$$C = \frac{Q_{ij}}{R_{ij}} ij \in P_{s,d} \tag{15}$$

(2) Blocking rate of key distribution operations. The key distribution blocking rate is the probability of key distribution failure due to channel conditions, network congestion, or other factors in a dynamic QKD routing method. The blocking rate of key distribution directly affects the system availability and the efficiency of key generation. By reducing the blocking rate of key distribution, the dynamic QKD routing method can improve the success rate of key generation, effectively reduce the risk of system interruption, and improve the efficiency of key generation. The key distribution blocking probability can directly reflect the blocking situation and the system performance and is an important index to measure the performance of the QKD system. Assuming the number of blocking service interruptions occurring due to insufficient collection link resources, N_b and N_k denote the total number of service requests. Then, the blocking rate M of the key distribution service is obtained, where the calculation is expressed as follows (16):

$$M = \frac{N_b}{N_k} \tag{16}$$

(3) Time delay of the algorithm. The algorithm time delay refers to the time required to compute the optimal routing and key distribution path in the dynamic QKD routing method. This index reflects the response speed and real-time performance of the dynamic QKD routing method. The lower algorithm time delay can enable the dynamic QKD routing method to adapt faster to changes in network topology and alterations in key distribution requirements, thus improving the flexibility and efficiency of the system. In QKD networks, transmission, processing, and waiting delays are usually combined into a total delay. Assuming that the routing algorithm sends data from the source node to the destination node through n intermediate nodes, the total delay of this route is as shown in (17). The average delay per link is (18):

$$T_t = \sum_{i=1}^{n+1} (t_d + t_p + t_w) \tag{17}$$

$$T_D = \frac{T_t}{N_r} \tag{18}$$

where t_d represents the transmission time between node i and node i+1, t_p represents the time node i caches, forwards and processes data, and t_w represents the time node i waiting for responses from other nodes. *n+1* includes the source and destination nodes. Assuming there are N_r links on a path, T_D means the average delay of each link.

4. Experiments

4.1. Simulation Environment Configuration

The simulation of quantum networks is implemented by integrating QKDsim into Mininet. Using a scripting language based on Python Version 2.7, the operating system is Ubuntu 18.04, Open vSwitch Version 2.5.7 is used as the virtual switch, and OpenFlow Version 1.3 is used as the southbound interface protocol, where the simulation simulates for a fixed time of 300 s. The number of services arriving during this period ranges from 50 to 200. In the experiments, assume that the bandwidth is set to 50 Mbps, the initial size of the key pool is 100 KB, the initial amount of keys contained in the key pool is 50 key volumes, and the number of successfully negotiated link keys is random, ranging from a

few tens to a few hundred, depending on the given service. The experimental simulation topology is shown in Figure 6.

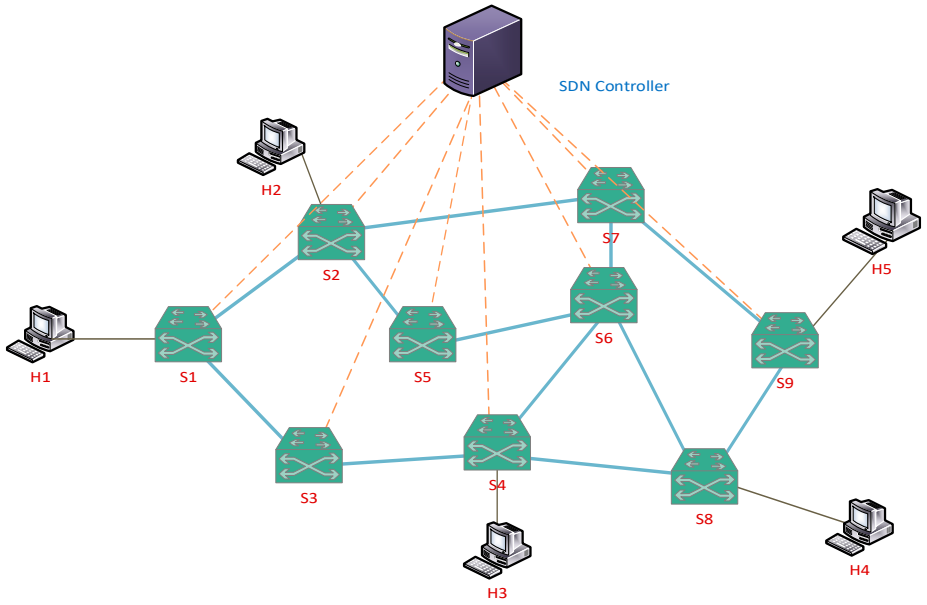

Figure 6. Topology of simulation experiment.

4.2. Simulation Results and Analysis

(1) Comparison of average usage of key resources

The average utilization rate of key resources for three algorithms with increasing business request volume is shown in Figure 7a, and the average utilization rate of keys fluctuates up and down as the number of business requests increases. This is because the actual number of key resources used will also increase correspondingly, while the number of generated keys remains relatively stable. Therefore, within a certain time period, the average utilization rate of keys will first increase until it reaches its peak, and then decrease in other time periods.

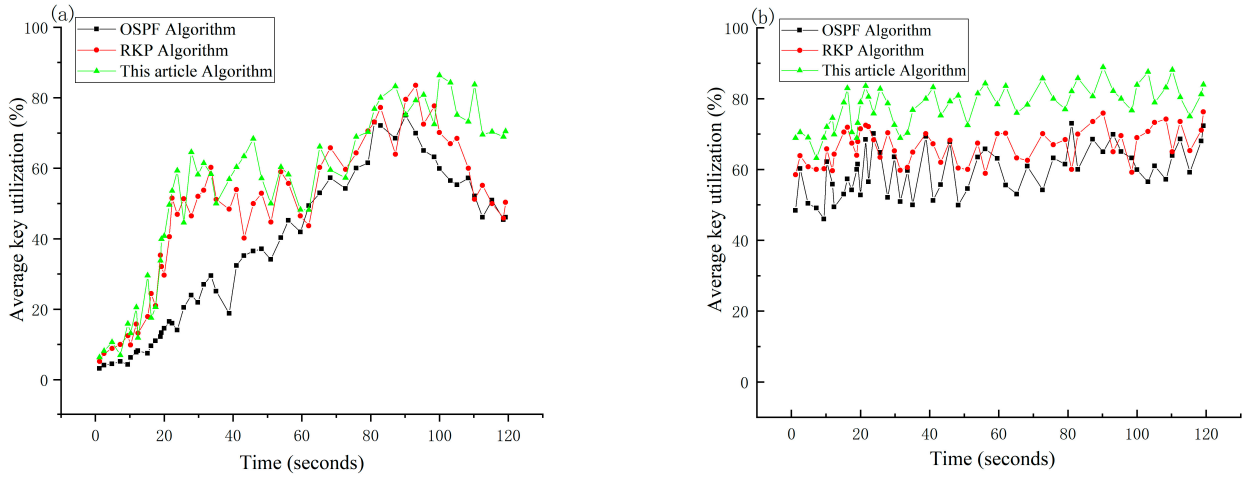

Figure 7. (**a**) Average utilization of key resources for three algorithms as business request volume increases; (**b**) Average utilization of keys when business request volume is constant.

As can be seen from Table 1, the algorithm has more significant values in terms of mean and standard deviation than the other two algorithms, thus indicating a more efficient use of key resources overall. As time t increases, the comparative graph of key resource

utilization efficiency for a given volume of service requests is shown in Figure 7. Figure 7b shows that the average utilization of key resources fluctuates and tends to stabilize. This is because when the business request volume is relatively stable, corresponding key resources can be configured based on actual needs, and proper scheduling and management can be carried out, thereby maintaining a steady utilization rate of key resources. However, fluctuations may be caused by factors such as noise in QKD systems and detector efficiency.

Table 1. Numerical calculation of the algorithm.

Algorithm	Mean Value	Standard Deviation	Least Value	Maximum Value	Median
OSPF Algorithm	35.54685	22.4593	3.2000	75.0000	35.8500
RKP Algorithm	47.38364	20.96505	5.2100	83.5000	51.19958
This article Algorithm	53.6791	23.25971	6.3400	86.3500	58.82423

In summary, the OSPF routing algorithm only focuses on path length, which may lead to a significant waste of key resources and a low utilization of these resources. However, the RKP routing algorithm and the routing algorithm proposed in this paper can better utilize existing key resources and avoid waste. Additionally, the routing algorithm proposed in this paper can more accurately estimate link conditions and dynamically adjust routes based on them, allowing for better utilization of key resources.

(2) Key distribution blocking probabilities

To evaluate the effectiveness and reliability of the algorithms more objectively, the comparative plots of the blocking probability of the key distribution service for the three algorithms with different service request volumes can be observed, as shown in Figure 8a–c. The volumes of 50/100/200 service requests are selected for comparison and analysis, respectively. The figure shows that the key blocking probability of the three algorithms is relatively flat when the service request volume is 50. However, when the number of service requests is 100 and 200 Erlang, the OSPF algorithm has a higher blocking rate than the other two because it does not fully consider the link situation. Its key distribution is blocking probability, mainly limited by bandwidth and reliability.

The RKP algorithm uses the remaining quantum key amount in the QKP on the routing path as the weight. It gives preference to the way with a more considerable remaining quantum key amount for key distribution, thus reducing the blocking probability of key distribution. Since this algorithm considers quantum resource utilization, it can effectively reduce the blocking possibility of key distribution. On the other hand, the algorithm in this paper considers the blocking probability of the link in addition to the number of remaining keys in the QKP. Combining the link conditions can reduce the key distribution blocking possibility more effectively.

(3) Delay comparison

The average key transmission delay comparison graph is shown in Figure 9. As the volume of service requests increases, more requests must wait for key resources to be allocated and utilized, and the key delay will increase accordingly. Initially, the OSPF algorithm outperforms the RKP algorithm and the algorithm in this paper because the OSPF algorithm reduces the complexity by selecting the shortest path for transmission, resulting in low latency transmission. However, at a later stage, the OSPF routing algorithm does not consider the status of the key pool of the QKD device, and therefore a longer waiting time may occur. The RKP algorithm, on the other hand, is based on a routing algorithm with the amount of remaining quantum keys in the quantum key pool as the weight.

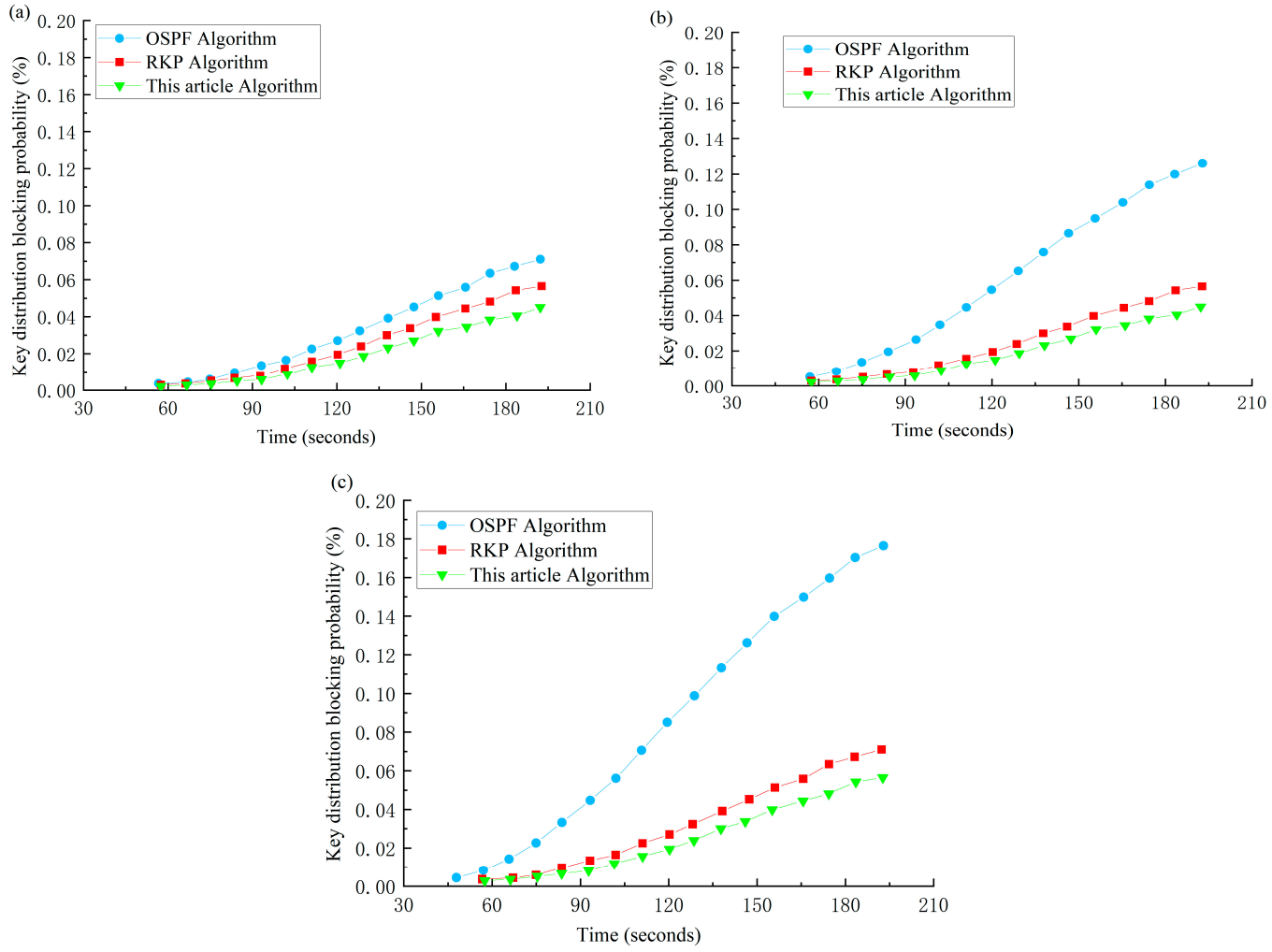

Figure 8. (**a**) Key blocking probability when traffic volume is 50 Erlang; (**b**) Traffic blocking probability when traffic volume is 100 Erlang; (**c**) Traffic blocking probability when traffic volume is 200 Erlang.

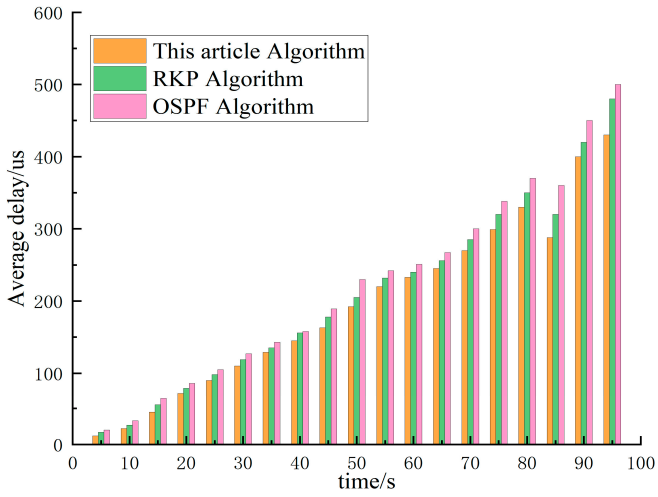

Figure 9. Comparison of average latency of key transmission.

In contrast, only the number of keys in the key pool is considered without considering the specific condition of the link. And thus, unreliable connections may be selected for routing in the case of poor link status. In the early preparation stage, the algorithm in

this paper takes some time to process and calculate the path, resulting in high latency. However, at a later stage, the algorithm in this paper considers the congestion situation. It can optimize the path selection, thus reducing the latency to a large extent.

5. Conclusions

Through the above method, the proposed routing algorithm can effectively reduce the waste of quantum key resources and improve the reliability and availability of networks. In this paper, the classical OSPF and RKP algorithms are compared and simulated, improving the utilization rate of quantum key resources and reducing network congestion on the whole. This algorithm provides a reference for the key distribution algorithm. Future research should continue to deepen the algorithm research comprehensively, improve the efficiency and security of the algorithm, and explore the performance and optimization strategy of the algorithm in different application scenarios.

Author Contributions: L.B.: conceptualization, methodology, formal analysis, survey, resources, data management, validation, writing—original draft, project management. M.M.: conceptualization, methodology, formal analysis, investigation, data management, experimental validation, writing—original draft, writing—review and editing. X.D.: software, formal analysis, visualization, supervision, funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: The Natural Science Foundation of Jilin Province: research on converged network architecture based on a virtualized QKD model. No.: 20210101417JC.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data used in this paper can be obtained by contacting the authors of this study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tsai, C.-W.; Yang, C.-W.; Lin, J.; Chang, Y.-C.; Chang, R.-S. Quantum Key Distribution Networks: Challenges and Future Research Issues in Security. *Appl. Sci.* **2021**, *11*, 3767. [CrossRef]
2. Diamanti, E.; Lo, H.K.; Qi, B.; Yuan, Z. Practical challenges in quantum key distribution. *Npj Quantum Inf.* **2016**, *2*, 16025. [CrossRef]
3. Bencheikh, K.; Symul, T.; Jankovic, A.; Levenson, J.A. Quantum key distribution with continuous variables. *J. Mod. Opt.* **2001**, *48*, 1903–1920. [CrossRef]
4. Dong, H.; Song, Y.; Yang, L. Wide Area Key Distribution Network Based on a Quantum Key Distribution System. *Appl. Sci.* **2019**, *9*, 1073. [CrossRef]
5. Chatterjee, S.; Goswami, K.; Chatterjee, R.; Sinha, U. Polarization bases compensation towards advantages in satellite-based QKD without active feedback. *Commun. Phys.* **2023**, *6*, 116. [CrossRef]
6. Song, T.T.; Qin, S.J.; Wen, Q.Y.; Wang, Y.K.; Jia, H.Y. Finite-key security analyses on passive decoy-state QKD protocols with different unstable sources. *Sci. Rep.* **2015**, *5*, 15276. [CrossRef]
7. Beutel, F.; Gehring, H.; Wolff, M.A.; Schuck, C.; Pernice, W. Detector-integrated on-chip QKD receiver for GHz clock rates. *Npj Quantum Inf.* **2021**, *7*, 40. [CrossRef]
8. Chen, L.Q.; Chen, J.Q.; Chen, Q.Y.; Zhao, Y.L. A quantum key distribution routing scheme for hybrid-trusted QKD network system. *Quantum Inf. Process* **2023**, *22*, 75. [CrossRef]
9. Xuan, W. *Research on Dynamic Routing and Application Access of Quantum Secure Communication Network*; Xidian University: Xi'an, China, 2014.
10. Han, Q.; Yu, L.; Zheng, W.; Cheng, N.; Niu, X. A novel QKD network routing algorithm based on optical-path-switching. *J. Inf. Hiding Multimed. Signal Process.* **2014**, *5*, 13–19.
11. Ma, C.; Guo, Y.; Su, J.; Yang, C. Hierarchical routing scheme on wide-area quantum key distribution network. In Proceedings of the 2016 2nd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 14–17 October 2016; 2009–2014.
12. Tanizawa, Y.; Takahashi, R.; Dixon, A.R. A routing method designed for a Quantum Key Distribution network. In Proceedings of the 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), Vienna, Austria, 5–8 July 2016; 208–214.

13. Yan, J.; Zhang, Y.; Li, H.; Yang, Y.; Zheng, D. An Improved Quantum Key Distribution Routing Algorithm Based on Reinforcement Learning. *IEEE Access* **2020**, *8*, 87277–87285.
14. Zhu, Q.; Yu, X.; Zhao, Y.; Nag, A.; Zhang, J. Resource Allocation in Quantum-Key-Distribution- Secured Datacenter Networks With Cloud–Edge Collaboration. *IEEE Internet Things J.* **2023**, *10*, 10916–10932. [CrossRef]
15. Scarani, V.; Bechmann-Pasquinucci, H.; Cerf, N.J.; Dušek, M.; Lütkenhaus, N.; Peev, M. The security of practical quantum key distribution. *Rev. Mod. Phys.* **2009**, *81*, 1301. [CrossRef]
16. Wang, H.; Zhao, Y.; Nag, A. Quantum-Key-Distribution (QKD) Networks Enabled by Software-Defined Networks (SDN). *Appl. Sci.* **2019**, *9*, 2081. [CrossRef]
17. Aguado, A.; López, V.; Brito, J.P.; Pastor, A.; López, D.R.; Martin, V. Enabling Quantum Key Distribution Networks via Software-Defined Networking. In Proceedings of the 2020 International Conference on Optical Network Design and Modeling (ONDM), Barcelona, Spain, 18–21 May 2020.
18. Cao, Y.; Zhao, Y.; Yu, X.; Zhang, J. Multi-tenant provisioning over software defined networking enabled metropolitan area quantum key distribution networks. *JOSA B* **2019**, *36*, B31–B40. [CrossRef]
19. Lopez, V.; Pastor, A.; Lopez, D.; Aguado, A.; Martin, V. Applying QKD to improve next-generation network infrastructures. In Proceedings of the 2019 European Conference on Networks and Communications (EuCNC), Valencia, Spain, 18–21 June 2019; 283–288.
20. Tang, Z.; Zhang, P.; Krawec, W.O. Enabling Resilient Quantum-Secured Microgrids Through Software-Defined Networking. *IEEE Trans. Quantum Eng.* **2022**, *3*, 4100811. [CrossRef]
21. Cao, Y.; Zhao, Y.; Wang, J.; Yu, X.; Ma, Z.; Zhang, J. KaaS: Key as a Service over Quantum Key Distribution Integrated Optical Networks. *IEEE Commun. Mag.* **2019**, *57*, 152–159. [CrossRef]
22. Boyer, M.; Liss, R.; Mor, T. Composable Security of Generalized BB84 Protocols Against General Attacks. *arXiv* **2022**, arXiv:2208.12154.
23. Winiarczyk, P.; Zabierowski, W. BB84 analysis of operation and practical considerations and implementations of quantum key distribution systems. In Proceedings of the 2011 11th International Conference the Experience of Designing and Application of CAD Systems in Microelectronics (CADSM), Polyana, Ukraine, 23–25 February 2011; pp. 23–26.
24. Hong, C.H.; Heo, J.; Jang, J.G.; Kwon, D. Quantum identity authentication with single photon. *Quantum Inf. Process.* **2017**, *16*, 236. [CrossRef]
25. Oliveira, R.D.; Arabul, E.; Wang, R.; Vrontos, C.; Nejabati, R.; Simeonidou, D. Programmable, Latency-Aware and Dynamic Quantum-Secured Optical Network with Key Refresh Rate Negotiation and QKD Sharing. In Proceedings of the 2023 Optical Fiber Communications Conference and Exhibition (OFC), San Diego, CA, USA, 5–9 March 2023; pp. 1–3.
26. Sun, Y. A Differentialized Service Providing Scheme on Trusted Relay Quantum Key Distribution Networks. *Acta Photonica Sin.* **2014**, *17*, 74–78.
27. Yang, C.; Zhang, H.; Su, J. Quantum key distribution network: Optimal secret-key-aware routing method for trust relaying. *China Commun.* **2018**, *15*, 33–45. [CrossRef]
28. Salvail, L.; Peev, M.; Diamanti, E.; Alléaume, R.; Lütkenhaus, N.; Länger, T. Security of Trusted Repeater Quantum Key Distribution Networks. *J. Comput. Secur.* **2010**, *18*, 61–87. [CrossRef]
29. Lin, X.; Hou, G.; Lin, W.; Chen, K. Quantum key distribution in partially-trusted QKD ring networks. In Proceedings of the 2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education (ICISCAE), Dalian, China, 27–29 September 2020; pp. 33–36.
30. Mehic, M.; Fazio, P.; Rass, S.; Maurhart, O.; Peev, M.; Poppe, A.; Rozhon, J.; Niemiec, M.; Voznak, M. A Novel Approach to Quality-of-Service Provisioning in Trusted Relay Quantum Key Distribution Networks. *IEEE/ACM Trans. Netw.* **2020**, *28*, 168–181. [CrossRef]
31. Huttner, B.; Alléaume, R.; Diamanti, E.; Fröwis, F.; Grangier, P.; Hübel, H.; Martin, V.; Poppe, A.; Slater, J.A.; Spiller, T.; et al. Long-range QKD without trusted nodes is not possible with current technology. *Npj Quantum Inf.* **2022**, *8*, 108. [CrossRef]
32. Cao, Y.; Zhao, Y.; Colman-Meixner, C.; Yu, X.; Zhang, J. Key on demand (KoD) for software-defined optical networks secured by quantum key distribution (QKD). *Opt. Express* **2017**, *25*, 26453–26467. [CrossRef] [PubMed]
33. Lo, H.K.; Curty, M.; Tamaki, K. Secure quantum key distribution. *Nat. Photon* **2014**, *8*, 595–604. [CrossRef]
34. Wang, Q.; Yu, X.; Zhu, Q.; Zhao, Y.; Zhang, J. Quantum key pool construction and key distribution scheme in multi-domain QKD optical networks (QKD-ON). In Proceedings of the 4th Optics Young Scientist Summit (OYSS 2020), Ningbo, China, 28 February 2021.
35. Jia, J.; Dong, B.; Kang, L.; Xie, H.; Guo, B. Cost-Optimization-Based Quantum Key Distribution over Quantum Key Pool Optical Networks. *Entropy* **2023**, *25*, 661. [CrossRef]
36. Zhang, Q.; Ayoub, O.; Gatto, A.; Wu, J.; Musumeci, F.; Tornatore, M. Routing, Channel, Key-Rate and Time-Slot Assignment for QKD in Optical Networks. *IEEE Trans. Netw. Serv. Manag.* 2023; *early access*. [CrossRef]
37. Yu, X.; Ning, X.; Zhu, Q.; Lv, J.; Zhao, Y.; Zhang, H.; Zhang, J. Multi-Dimensional Routing, Wavelength, and Timeslot Allocation (RWTA) in Quantum Key Distribution Optical Networks (QKD-ON). *Appl. Sci.* **2020**, *11*, 348. [CrossRef]
38. Akhtar, M.S.; Krishnakumar, G.; Vishnu, B.; Sinha, A. Fast and Secure Routing Algorithms for Quantum Key Distribution Networks. *IEEE/ACM Trans. Netw.* 2023; *early access*. [CrossRef]

39. Meng, X.; Yu, X.; Chen, W.; Zhao, Y.; Zhang, J. Residual-adaptive Key Provisioning in Quantum-Key-Distribution Enhanced Internet of Things (Q-IoT). In Proceedings of the 2020 International Wireless Communications and Mobile Computing (IWCMC), Limassol, Cyprus, 15–19 June 2020; pp. 2022–2027.
40. Pan, X.; Lu, Z.; Wang, W.; Hua, Z.; Xu, Y.; Li, W.; Cai, W.; Li, X.; Wang, H.; Song, Y.P.; et al. Deep quantum neural networks on a superconducting processor. *Nat. Commun.* **2023**, *14*, 4006. [CrossRef]