Multi-Tenant Provisioning for Quantum Key Distribution Networks with Heuristics and Reinforcement Learning: A Comparative Study

Citation for the original published paper (version of record):
cao, y., Zhao, Y., Li, J. et al (2020). Multi-Tenant Provisioning for Quantum Key Distribution Networks with Heuristics and
Reinforcement Learning: A Comparative Study. IEEE Transactions on Network and Service Management, 17(2): 946-957. http://dx.doi.org/10.1109/TNSM.2020.2964003

Multi-Tenant Provisioning for Quantum Key Distribution Networks with Heuristics and Reinforcement Learning: A Comparative Study

Yuan Cao, Yongli Zhao, *Senior Member, IEEE*, Jun Li, Rui Lin, Jie Zhang, and Jiajia Chen, *Senior Member, IEEE*

***Abstract*—Quantum key distribution (QKD) networks are potential to be widely deployed in the immediate future to provide long-term security for data communications. Given the high price and complexity, multi-tenancy has become a cost-effective pattern for QKD network operations. In this work, we concentrate on addressing the online multi-tenant provisioning (On-MTP) problem for QKD networks, where multiple tenant requests (TRs) arrive dynamically. On-MTP involves scheduling multiple TRs and assigning non-reusable secret keys derived from a QKD network to multiple TRs, where each TR can be regarded as a high-security-demand organization with the dedicated secret-key demand. The quantum key pools (QKPs) are constructed over QKD network infrastructure to improve management efficiency for secret keys. We model the secret-key resources for QKPs and the secret-key demands of TRs using distinct images. To realize efficient On-MTP, we perform a comparative study of heuristics and reinforcement learning (RL) based On-MTP solutions, where three heuristics (i.e., random, fit, and best-fit based On-MTP algorithms) are presented and a RL framework is introduced to realize automatic training of an On-MTP algorithm. The comparative results indicate that with sufficient training iterations the RL-based On-MTP algorithm significantly outperforms the presented heuristics in terms of tenant-request blocking probability and secret-key resource utilization.**

***Index Terms*—Quantum key distribution networks, online multi-tenant provisioning, heuristics, reinforcement learning.**

I. Introduction

NUMEROUS high-security-demand organizations in the fields of finance, transport, energy, health, etc., are in urgent need of long-term secure communications across the Internet. Data encryption has become of critical importance to ensure the confidentiality and integrity of data transmission [1], [2]. In conventional cryptographic systems such as Rivest-Shamir-Adleman [3], the security of key distribution relies on assumptions about the limitation of existing computational power for any illegitimate party to crack the secret keys [4]. However, the rise of powerful quantum computers and quantum algorithms renders the conventional key distribution approaches insecure [5]–[7].

Quantum key distribution (QKD) [8]–[10] is a state-of-the-art method for distributing unconditionally secure symmetric secret keys based on the fundamental laws of quantum physics such as quantum no-cloning theorem [11], [12]. The symmetric cryptographic systems can then utilize the secret keys for data encryption. In recent years, significant progress has been achieved in point-to-point QKD over both optical fibers and free space [13], [14]. Moreover, different QKD implementation options such as discrete-variable QKD (DV-QKD) [15]–[18] and continuous-variable QKD (CV-QKD) [19]–[23] have been adopted, and numerous QKD protocols such as Bennett-Brassard-1984 (BB84) protocol [24] and Grosshans-Grangier-2002 (GG02) protocol [25] have been invented. The fiber-based QKD networks have been successfully deployed [14], [26]–[29], and their integration with classical networks has been widely demonstrated and validated [30]–[36]. Hence, QKD networks have a great potential to be deployed over the ubiquitous existing fiber infrastructure for telecommunication networks in the immediate future to provide long-term security for data communications.

On the other hand, the price and complexity of deploying a QKD network are still high by now, especially if the QKD network is owned by a single organization that demands high security. To solve this problem, cost-effective patterns of applying secret keys are necessary before QKD networks can be widely used. In this regard, multi-tenancy is a promising approach to improve cost efficiency for QKD network operations [37], in which the secret keys can be assigned to multiple tenants that share the QKD network. Each tenant request (TR) can be regarded as one organization with a dedicated demand for secret keys. Considering the non-reusable nature, the secret key becomes a precious and unique resource that makes the corresponding assignment different from many other types of network resources for multi-tenant provisioning (MTP). Thus, how to efficiently assign the non-reusable secret keys derived from a QKD network to multiple tenants needs to

Manuscript received XXXX. This work was supported in part by National Natural Science Foundation of China (Grant Nos. 61601052 and 61822105), Fundamental Research Funds for the Central Universities (Grant No. 2019XD-A05), State Key Laboratory of Information Photonics and Optical Communications of China (Grant No. IPOC2019ZR01), China Association for Science and Technology, Swedish Research Council, Swedish Foundation for Strategic Research, Swedish Foundation for International Collaboration in Research and Higher Education, and Göran Gustafsson Foundation. *(Corresponding author: Yongli Zhao.)*

Y. Cao, Y. Zhao, and J. Zhang are with the State Key Laboratory of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: yuancao@bupt.edu.cn; yonglizhao@bupt.edu.cn; lgr24@bupt.edu.cn).

J. Li is with the KTH Royal Institute of Technology, Electrum 229, Kista, Sweden (e-mail: jun5@kth.se).

R. Lin and J. Chen are with the Chalmers University of Technology, SE-412 96, Gothenburg, Sweden (e-mail: ruilin@chalmers.se; jiajiac@chalmers.se).

be explored.

In general, MTP can be classified into offline (i.e., static) and online (i.e., dynamic) problems. In the offline MTP (Off-MTP) problem, TRs are known in advance, whereas in the online MTP (On-MTP), TRs arrive dynamically without pre-knowledge. Recently, MTP for QKD networks has been demonstrated in the lab [37] and its offline version has been addressed [38]. Compared with Off-MTP, On-MTP involves not only the assignment of non-reusable secret keys to multiple TRs but also the scheduling of TRs. To solve the On-MTP problem in QKD networks, conventional heuristics can be adopted but their efficiency may be challenged. Inspired by artificial intelligence techniques, reinforcement learning (RL) might be promising to realize efficient On-MTP. In this regard, a comparative study of heuristics and RL can be carried out to investigate the efficiency of On-MTP and provide insights for QKD network operations.

Our previous work presented a RL based On-MTP solution for QKD networks [39], which outperforms conventional heuristics in terms of tenant-request blocking probability and secret-key resource utilization. This paper further extends our preliminary work [39] by providing a more comprehensive formulation of the On-MTP problem and performing a comparative study of heuristics and RL based On-MTP solutions. Several quantum key pools (QKPs) [40], [41] are constructed over the QKD network infrastructure to improve the management efficiency for secret keys. The major contributions of this work are four-fold: 1) we model the secret-key resources for QKPs and the secret-key demands of TRs using distinct images; 2) we present three heuristics, i.e., random, fit, and best-fit (BF) based On-MTP algorithms; 3) we introduce a RL framework to address the On-MTP problem for QKD networks, where an On-MTP algorithm is automatically trained with RL; 4) we perform a comparative evaluation of heuristics and RL based On-MTP solutions considering two QKD network types in terms of tenant-request blocking probability and secret-key resource utilization.

The remainder of this paper is structured as follows. Section II reviews the related work. Section III describes the problem formulation, where the network architecture, network model, and objective are given. Three heuristics are presented to realize On-MTP for QKD networks in Section IV. A RL framework is introduced in Section V to solve the On-MTP problem. A comprehensive result analysis of heuristics and RL for On-MTP is carried out and discussed in Section VI. Finally, we conclude this work in Section VII.

II. Related Work

This section briefly reviews the QKD networks from different perspectives, including the deployment, management, and operation of QKD networks. Typically, a QKD network first needs to be deployed in the field, then managed/controlled by the owner/operator, and operated during its lifetime to provide services for users. The point-to-point QKD mechanism is not a focus in this paper and its details can refer to [41]. Moreover, our study in this paper is not bound to any specific QKD implementation options or QKD protocols.

A. QKD Network Deployment

From the perspective of QKD network deployment, several fiber-based QKD networks (e.g., SECOQC [26], Tokyo [27], SwissQuantum [28], and Beijing-Shanghai [14] QKD networks) have been successfully deployed, and a satellite-based QKD network has been experimentally validated [13]. Each of the newly deployed QKD networks is a trusted repeater QKD network that uses the trusted repeater to extend QKD distance. On the other hand, a quantum repeater [42] that can forward the quantum signals without measuring or cloning them is still not mature, and hence has not yet deployed in the field. Accordingly, the trusted repeater QKD network is adopted in this paper. In deployed QKD networks, dedicated channels are typically needed to transmit the quantum signals, which can protect quantum signals by minimizing impacts of interference caused by intensive signals for classical data transmission. However, dedicated fibers are expensive and scarce, which may limit the practical deployment of QKD networks. In recent years, the desire to reduce the capital expenditures of QKD network deployment has motivated the research of QKD integration with classical networks, where both of physical-layer performance and network-layer performance are taken into account. In order to improve the physical-layer performance such as secret-key rate and achievable distance, a number of analytical studies [30], [43], system experiments [31], [34], [36], and field trials [33], [44] have been carried out. On the other hand, several resource assignment strategies have been proposed to optimize the network-layer performance such as blocking probability and resource utilization when QKD coexists with the classical networks [32], [35]. In [45], [46], the protection and recovery schemes were described for resilient QKD-integrated classical networks. In [47], cost-efficient QKD networking approaches were presented to minimize the deployment cost.

B. QKD Network Management

From the perspective of QKD network management, software defined networking (SDN) and QKP techniques are promising to enhance the management efficiency for QKD networks. SDN holds a programmable and flexible centralized control manner to provide efficient and easy management for QKD networks. In [48], it was demonstrated that SDN can reduce costs in a QKD network for time-sharing of the available resources. In [49], SDN was introduced in a QKD network to provide real-time monitoring of quantum parameters, e.g., secret-key rate and quantum bit error rate. In [37], SDN is adopted to achieve effective and flexible MTP over a QKD metropolitan network. In [50], the end-to-end key on demand service provisioning over a SDN-controlled QKD network was demonstrated. In [44], SDN was combined with machine learning to achieve dynamic and optimal wavelength allocation for both quantum and classical channels. Meanwhile, with regards to QKP technique, several QKPs were constructed to enable on-demand secret-key volume allocation for control channels and data channels in a

software defined optical network [40]. In [41], a time-scheduled scheme was proposed for QKP construction to efficiently schedule QKD over classical networks. Also, the QKP technique is adopted in this paper.

C. QKD Network Operation

From the perspective of QKD network operation, several cost-effective patterns and use cases have been presented. On one hand, the concepts of multi-tenancy [37]–[39] and QKD as a service [51] were proposed to improve the cost efficiency for QKD network operations. In [38], the Off-MTP problem was addressed for QKD networks by combing a secret-key rate sharing scheme with an Off-MTP algorithm, but the performance metrics were not optimized. In [37], the establishment, adjustment, and deletion of multiple TRs over a SDN-enabled QKD metropolitan network were experimentally demonstrated. In [51], a new concept of QKD as a service was proposed where multiple users can obtain their required secret-key rates from the same QKD network infrastructure by applying for their dedicated QKD services. On the other hand, QKD can be adopted in several use cases, e.g., data service security enhancement [35], [40], physical-layer attack mitigation [49], security enhancement of SDN control [52], energy efficiency improvement for Internet of things [53], and virtual network security enhancement [54], [55].

This study targets multi-tenancy for QKD network operation and concentrates on addressing one of important problems in future QKD networks, i.e., On-MTP. All the requests in the currently deployed QKD networks are statically planned, whereas the On-MTP for QKD networks is highly demanded to improve network agility and its associated problem needs to be solved in the future. Moreover, this paper extends our preliminary work presented in [39] by elaborating a formulation of On-MTP problem as well as a comprehensive comparison of presented heuristics and RL based On-MTP solutions.

III. Problem Formulation

In this section, we introduce the network architecture, describe the network model, and define the objective to formulate the On-MTP problem for QKD networks. The QKD network type considered in this work is a backbone network or a metropolitan network. Several examples of QKD backbone and metropolitan networks can be found in [14]. As noted before, each QKD backbone or metropolitan network is a trusted repeater QKD network in this paper.

A. Network Architecture

Figure 1(a) illustrates a network architecture for multiple TRs over the QKD network, comprising three layers, namely, deployment layer (DL) representing the QKD network infrastructure, management layer (ML) where QKPs are located, and operation layer (OL) that handles multiple TRs. This network architecture is shown in Fig. 1(a) at a logical level and its three layers are detailed as follows.

In the DL, QKD network infrastructure that consists of several QKD nodes (QNs) interconnected by QKD links is

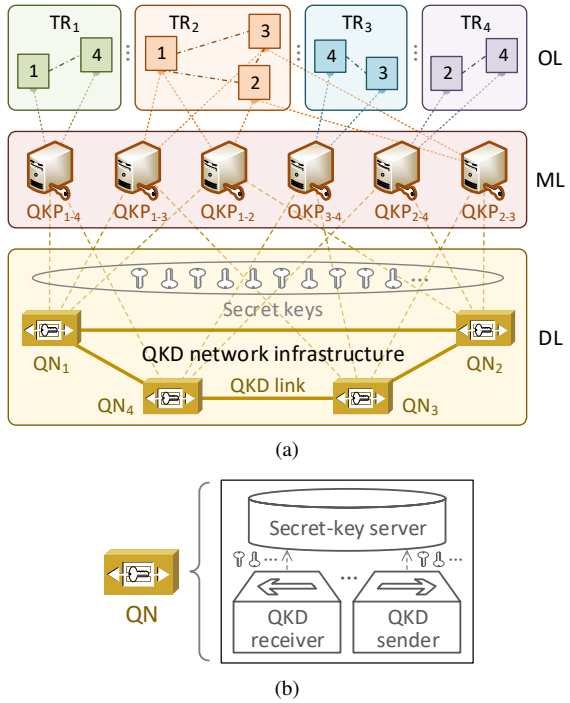

Fig. 1. (a) Network architecture: multiple TRs over the QKD network; (b) QN structure.

deployed. Also, several intermediate nodes equipped with trusted repeaters are placed along the QKD links to extend the QKD distance. Each QN or intermediate node is trustable in a trusted repeater QKD network. A QN acts as the endpoint to multiple TRs, whereas there is no TR generated or terminated at the intermediate node. The detailed structure of a QN or an intermediate node can refer to [38], which mainly consists of one or more QKD senders and receivers as well as a secret-key server. An example of the structure of a QN is shown in Fig. 1(b), illustrating the location of the QKD sender, QKD receiver, and secret-key server. In practice, a point-to-point QKD link can be established by connecting a QKD sender to a QKD receiver, thereby the point-to-point secret keys can be derived from this QKD link and stored in the corresponding secret-key servers. Additionally, an end-to-end QKD connection can be established by relaying the point-to-point secret keys from the source QN to the destination QN in a hop-by-hop manner. For example, the point-to-point secret keys are derived from the first hop (i.e., the first QKD link between the source QN and the first intermediate node), and then are encrypted and decrypted with the secret keys in the secret-key server of each intermediate node between source and destination QNs. The secret keys derived from the end-to-end QKD connection are referred as end-to-end secret keys. In principle, these end-to-end secret keys can achieve information-theoretic security with the aid of one-time pad cryptographic algorithm for encryption and decryption [56].

In the ML, several QKPs are located and constructed to improve the management efficiency for secret keys derived from the QKD network infrastructure. The four stages (i.e., secret-key exchange, storage, assignment, and destruction)

during the overall lifetime of secret keys are described in [38], which are all handled in the QNs across the QKD network in a distributed manner. The security of secret keys is assured since they are not delivered across different physical locations. That is, the secret keys derived from a QN can be delivered to a TR only when the QN is co-located with this TR. A QKP is constructed between two QNs to manage the secret keys in a pair-wise manner, which can acquire the secret-key rate information (i.e., the generation of secret keys in bits per second) from the corresponding two QNs, but it cannot access the real secret keys. In practice, all the QKPs in the ML can be implemented using a SDN controller, that is, secret-key management can be implemented in a centralized manner.

In the OL, multiple TRs arrive dynamically. As an example, the Beijing-Shanghai QKD network holds numerous real-world applications in banks and insurances [14], where each TR can be triggered by a bank or an insurance company for high security. A TR needs to obtain secret keys from several specific QNs in the QKD network within its duration. The number of specific QNs required corresponding to the secret-key demands of each TR can be different. For instance, TR_1 has one secret-key demand between QN_1 and QN_4 corresponding to $QKP_{1\text{-}4}$, whereas TR_2 has three secret-key demands among QN_1, QN_2 and QN_3 corresponding to $QKP_{1\text{-}2}$, $QKP_{1\text{-}3}$ and $QKP_{2\text{-}3}$.

TABLE I
NOTATIONS AND DEFINITIONS

Notations	Definitions
$G(N, L)$	QKD network topology
N	Set of QNs in the QKD network
L	Set of QKD links in the QKD network
Q	Set of QKPs over the QKD network
$Q_{i\text{-}j}$	QKP for a pair of QNs i and j ($i < j$, $i \neq j$)
t	One time step
T	Time-step length in each secret-key resource/demand image
K	Secret-key capacity for a QKP at each time step
R	Set of total incoming TRs over the QKD network
$r(n_r, d_r, t_r)$	A TR
n_r	Set of QNs corresponding to the secret-key demands of r
d_r	Set of secret-key demands of r
t_r	Duration of r
q_r	Set of QKPs corresponding to the secret-key demands of r
d^r_{i-j}	Secret-key demand of r between QNs i and j at each time step
M	Scheduling capacity for TRs at each time step
R_m	Set of the first M TRs in the buffer at each time step
R_s	Set of total accepted TRs over the QKD network
K_r	Total secret-key demands of r at each time step
t_c	Current time step
R_v	Set of TRs that can fit the free secret-key resources for QKPs
$K^{t_c}_{i-j}$	Free secret-key resources between QNs i and j at t_c
D_r	Matching degree between r and free secret-key resources
$R^{t_c}_u$	Set of rejected TRs at t_c
U	Total operation time of the QKD network
BP_{TR}	Tenant-request blocking probability
RU_{SK}	Secret-key resource utilization
E	Cumulative discounted reward over time
γ	Discount factor
ε_{t_c}	Reward at the current time step t_c
CR	Cumulative reward over time

B. Network Model

The network model is described according to the network architecture shown in Fig. 1(a). Table I lists the notations and their definitions used in this paper. We model a QKD network topology as $G(N, L)$, where N and L denote the sets of QNs and QKD links in the QKD network, respectively. We assume that several intermediate nodes have been deployed along the QKD links for secret-key relay during the network deployment phase, while the number of intermediate nodes does not affect MTP and hence is not considered in this work. A QN corresponds to an endpoint of multiple TRs within a single node physical location. We model a QKP as $Q_{i\text{-}j}$ between a pair of QNs i and j ($i < j$, $i \neq j$), while the set of QKPs over the QKD network is denoted by Q. Based on the definition of the QKP, its total number over a QKD network can be expressed as:

$$|Q| = \frac{|N| \cdot (|N| - 1)}{2} \tag{1}$$

where $|N|$ represents the total number of QNs.

In particular, we use an image to model the secret-key resources for each QKP between a pair of QNs, as shown in Fig. 2(a). The total number of images used to model secret-key resources for QKPs over the QKD network is equal to $|Q|$. The row and column of each image represent time and secret-key resource, respectively. There are many normalized rectangles in an image, where the vertical length and horizontal length for a rectangle are defined as one time step (denoted by t) and secret-key resource unit (denoted by 1 unit) that can accommodate a TR, respectively. The white rectangle denotes free (i.e., the secret-key resource unit is unoccupied at this time step), while a rectangle filled with other colors (besides white) represents occupied. The distinct colors of occupied rectangles represent different TRs accommodated by the QKD network, while the same color represents a TR with secret-key demands corresponding to several specific QKPs. Moreover, we assume that the time-step length in each image is the same (denoted by T), and the secret-key resources derived from a pair of QNs (i.e.,

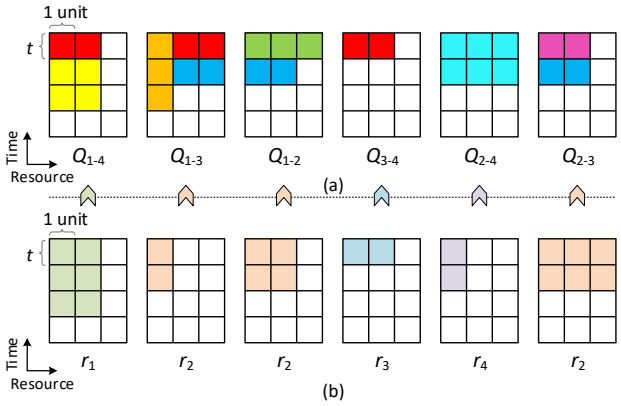

Fig. 2. Network model: (a) secret-key resource images for QKPs; (b) secret-key demand images for TRs.

secret-key capacity for a QKP) at each time step is the same (denoted by K units).

Meanwhile, we model a TR as $r(n_r, d_r, t_r)$, where n_r denotes the set of specific QNs corresponding to the secret-key demands of r, d_r represents the set of secret-key demands of r, and t_r is the duration of r. The set of total incoming TRs over the QKD network is denoted by R. Let q_r represent the set of specific QKPs corresponding to the secret-key demands of r. Then, the number of specific QKPs required corresponding to the secret-key demands of r can be expressed as:

$$|q_r| = \frac{|n_r| \cdot (|n_r| - 1)}{2} \tag{2}$$

where $|n_r|$ represents the number of QNs required corresponding to the secret-key demands of r.

On the other hand, we also use images to model the secret-key demands of TRs, as depicted in Fig. 2(b). The number of images used to model the secret-key demands of r is equal to $|q_r|$. The size of each secret-key demand image for a TR is the same as the size of secret-key resource image for a QKP, which can facilitate the scheduling of TRs and the assignment of secret keys to TRs. The images with distinct colors (besides white) represent the secret-key demands of different TRs, whereas the same color depicts the secret-key demands of a TR corresponding to several specific QKPs. For example, r_1 ($|n_{r1}| = 2$, $|q_{r1}| = 1$) has a duration of three time steps ($t_{r1} = 3t$), and needs two units of secret-key resources ($d_{r1} = \{2\}$ units) corresponding to $Q_{1\text{-}4}$ at each time step, whereas r_2 ($|n_{r2}| = 3$, $|q_{r1}| = 3$) has a duration of two time steps ($t_{r2} = 2t$), and needs two, one and three units of secret-key resources ($d_{r2} = \{2, 1, 3\}$ units) corresponding to $Q_{1\text{-}2}$, $Q_{1\text{-}3}$ and $Q_{2\text{-}3}$ at each time step, respectively.

It should be noted that the image used to model secrete-key resources and demands is equivalent to a 2-dimentional matrix. As the image is illustrative to represent resources and demands, this data structure is selected in this study as many recent works addressing other networking issues [57]–[59]. In this work, we assume that multiple TRs arrive dynamically at discrete time steps and can tolerate queuing delay. Hence, the incoming TRs first wait in a buffer, and then are scheduled according to the result of secret-key assignment as time proceeds. If no resource, the incoming TR is rejected. Given more and more secret-key demand images for TRs are generated in the buffer as time proceeds, we fix the scheduling capacity for TRs at each time step as M so that the TRs can be scheduled in a scalable way. That is, only the images for the first M TRs in the buffer are admitted at each time step, where the M TRs have not yet been scheduled and are inserted into a set R_m.

C. Objective

This work has an assumption that QKD network deployment and management have been implemented. Namely, the QKD network infrastructure has been deployed, over which QKPs have been constructed. Since QKD network operation is independent of the classical networks, only an independent QKD network is considered in this paper. We assume that both point-to-point and end-to-end secret keys can be derived from the QKD network infrastructure, thereby routing schemes are not considered for TRs. In the On-MTP problem, the scheduling of multiple TRs and the assignment of secret keys to TRs are considered.

Specifically, two objectives are considered in the On-MTP problem for QKD networks: 1) to minimize the tenant-request blocking probability (denoted by BP_{TR}) and 2) to maximize the secret-key resource utilization (denoted by RU_{SK}). These two objectives are elaborated as follows.

1) BP_{TR} is defined as the ratio of the total rejected TRs to the total incoming TRs over the QKD network, which can be expressed as:

$$BP_{\text{TR}} = 1 - \frac{|R_s|}{|R|} \tag{3}$$

where R_s is the set of total accepted TRs over the QKD network. When the QKD network can accommodate as many TRs as possible, the value of BP_{TR} ($0 \le BP_{\text{TR}} \le 1$) is minimized.

2) RU_{SK} is defined as the ratio of the total assigned secret-key resources to the total secret-key resources derived from a QKD network, which can be expressed as:

Algorithm 1: Random-Based On-MTP Algorithm.

Input: $G(N, L)$, Q, t, T, K, R, M, U.
Output: R_s, K_r for each TR in R_s, result of On-MTP for the QKD network, updated secret-key resources.

```
initialize R_s ← ∅;
for t_c = t to U do
    identify the TRs in R_m and the secret-key resource images for QKPs at t_c;
    while secret-key resource images for QKPs have white rectangles at t_c do
        if R_m ≠ ∅ then
            select a TR r(n_r, d_r, t_r) from R_m randomly;
            determine the set of specific QKPs q_r;
            search free secret-key resources for QKPs in q_r;
            if free secret-key resources for QKPs in q_r can satisfy the secret-key demands of r then
                insert r into R_s;
                assign the required secret-key resources to r using first-fit algorithm;
                compute K_r using Eq. (5);
                update the TRs in R_m and the secret-key resource images for QKPs at t_c;
            else
                remove r from R_m;
            end
        else
            break;
        end
    end
end
return R_s, K_r for each TR in R_s, result of On-MTP for the QKD network, updated secret-key resources.
```

$$RU_{\mathrm{SK}} = \frac{\sum_{r \in R_s} K_r \cdot t_r}{K \cdot U \cdot |Q|} \tag{4}$$

where K_r is the total secret-key demands of r at each time step, and U is the total operation time of the QKD network. When the secret-key resources in a QKD network can be utilized as efficiently as possible, the value of RU_{SK} ($0 \le RU_{\mathrm{SK}} \le 1$) is maximized. Additionally, K_r can be calculated as follows:

$$K_r = \sum_{i,j \in n_r} d_{i-j}^{r} \tag{5}$$

where d_{i-j}^{r} is the secret-key demand of r between a pair of QNs i and j at each time step.

IV. Heuristics Based On-MTP

In this section, according to the formulated On-MTP problem for QKD networks and the notations listed in Table I, we present three On-MTP heuristics to optimize the BP_{TR} and RU_{SK}. The presented three On-MTP heuristics are named as random, fit, and BF based On-MTP algorithms, which are shown in Algorithms 1, 2, and 3, respectively.

For each time step in the three On-MTP heuristics, the TRs in R_m and the secret-key resource images for QKPs are first identified at the current time step. If the secret-key resource images for QKPs have white rectangles and R_m contains several TRs at the current time step, the TRs from R_m are selected to realize the scheduling of TRs. Otherwise time proceeds to the next time step. The main difference of the three On-MTP heuristics is a TR selection strategy. In the random-based On-MTP algorithm, TRs from R_m are selected randomly. In the fit-based On-MTP algorithm, TRs from R_m that can fit the free secret-key resources for QKPs are selected and inserted into R_v. Moreover, in the BF-based On-MTP algorithm, the matching degree between each TR in R_v and free secret-key resources for QKPs are evaluated. Here, the matching degree (denoted by D_r) between r and free secret-key resources is defined as the ratio of the secret-key demands of r to the free secret-key resources for specific QKPs in q_r at the current time step, which can be expressed as:

$$D_r = \begin{cases} \dfrac{\sum_{i,j \in n_r} d_{i-j}^{r} \big/ K_{i-j}^{t_c}}{|q_r|} & d_{i-j}^{r} \le K_{i-j}^{t_c} \\ 0 & \text{else} \end{cases} \tag{6}$$

where $K_{i-j}^{t_c}$ is the free secret-key resources between a pair of QNs i and j at the current time step t_c. The value of D_r ($0 \le D_r \le$

Algorithm 2: Fit-Based On-MTP Algorithm.

Input: $G(N, L)$, Q, t, T, K, R, M, U.
Output: R_s, K_r for each TR in R_s, result of On-MTP for the QKD network, updated secret-key resources.

initialize $R_s \leftarrow \emptyset$;
for $t_c = t$ to U **do**
  identify the TRs in R_m and the secret-key resource images for QKPs at t_c;
  while secret-key resource images for QKPs have white rectangles at t_c **do**
    if $R_m \neq \emptyset$ **then**
      search free secret-key resources for QKPs;
      select the TRs from R_m that can fit the free secret-key resources and insert them into R_v;
      if $R_v \neq \emptyset$ **then**
        select a TR $r(n_r, d_r, t_r)$ from R_v randomly;
        call steps 10 to 13 in **Algorithm 1**;
      else
        break;
      end
    else
      break;
    end
  end
end
return R_s, K_r for each TR in R_s, result of On-MTP for the QKD network, updated secret-key resources.

Algorithm 3: BF-Based On-MTP Algorithm.

Input: $G(N, L)$, Q, t, T, K, R, M, U.
Output: R_s, K_r for each TR in R_s, result of On-MTP for the QKD network, updated secret-key resources.

initialize $R_s \leftarrow \emptyset$;
for $t_c = t$ to U **do**
  identify the TRs in R_m and the secret-key resource images for QKPs at t_c;
  while secret-key resource images for QKPs have white rectangles at t_c **do**
    if $R_m \neq \emptyset$ **then**
      search free secret-key resources for QKPs;
      select the TRs from R_m that can fit the free secret-key resources and insert them into R_v;
      if $R_v \neq \emptyset$ **then**
        determine the set of specific QKPs q_r for each TR in R_v;
        compute the matching degree D_r between each TR in R_v and free secret-key resources using Eq. (6);
        select a TR $r(n_r, d_r, t_r)$ with highest matching degree from R_v;
        call steps 10 to 13 in **Algorithm 1**;
      else
        break;
      end
    else
      break;
    end
  end
end
return R_s, K_r for each TR in R_s, result of On-MTP for the QKD network, updated secret-key resources.

1) is equal to 1 when the TR can fit best the free secret-key resources for QKPs, but it is equal to 0 when the secret-key demands are larger than the free secret-key resources for QKPs. In the BF-based On-MTP algorithm, a TR with a higher D_r is given a higher priority to be selected.

After the TR selection, the free secret-key resources for QKPs are checked whether the secret-key demands of the selected TRs can be satisfied in Algorithm 1, whereas this step is not performed in Algorithms 2 and 3 since the TRs that can fit the free secret-key resources for QKPs have been selected during the TR selection phase. When the secret-key demands of a TR can be satisfied, the required secret-key resources are assigned to this TR according to the first-fit algorithm. Currently, the feasibility and practicality of implementing the first-fit algorithm for secret-key assignment has been verified [38], [40]. Finally, when $t_c = U$, the values of BP_{TR} and RU_{SK} can be calculated based on the returned results of the three On-MTP heuristics.

The complexities of the presented three On-MTP heuristics are discussed as follows. In the worst condition, the complexities to handle one TR in Algorithms 1, 2, and 3 are $O(K^2 \cdot |Q|^2)$, $O(K \cdot |Q| \cdot (K \cdot |Q| + |R_m|))$, and $O(K \cdot |Q| \cdot (K \cdot |Q| + 3 \cdot |R_m|))$, respectively. It is obvious that the time complexity of the three On-MTP heuristics in a descending order is Algorithm 3 > Algorithm 2 > Algorithm 1.

V. Reinforcement Learning Based On-MTP

Over the past few years, the RL has attracted increasingly attentions in machine learning research area owing to its success in solving complex decision-making problems [60]. In [60], a detailed survey and rigorous derivations related to the RL are provided. The On-MTP problem involves deciding whether or not a new TR should be accepted, which can be regarded as a decision-making problem. Accordingly, the RL might provide a viable alternative to heuristics for On-MTP, which performs the task of learning how an agent should take a series of actions in an environment in order to maximize the expected cumulative discounted reward. In this study, we introduce a RL framework to solve the On-MTP problem for QKD networks, as depicted in Fig. 3. In this RL framework, three steps are implemented as follows: 1) the RL agent observes the current state from the environment (i.e., QKD network) at each time step; 2) the RL agent picks an action at each time step; and 3) the state of the environment occurs transitions following the action and the environment returns the RL agent a reward that indicates how good the action is. The state transitions and rewards are assumed to possess the Markov property [61], which means that the future of the process only relies on the current observation. That is, the state-transition probabilities and rewards rely only on the state of the environment and the action picked by the agent [58]. The goal of this RL framework is to maximize the cumulative discounted reward (denoted by E) over time that can be expressed as:

$$E = \sum_{t_c=t}^{U} \gamma^{t_c/t} \cdot \varepsilon_{t_c} \tag{7}$$

where $\gamma \in (0, 1]$ is the discount factor, and ε_{t_c} is the reward at the current time step t_c.

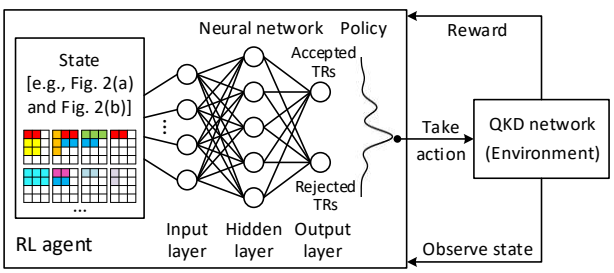

Fig. 3. RL framework for On-MTP.

In order to use this RL framework for On-MTP, the state, action, and reward should be defined. The observed current state from the QKD network contains the current status of secret-key resources for all QKPs and secret-key demands of all TRs in R_m, which can be represented by the secret-key resource images and the secret-key demand images, respectively.

The RL agent picks actions according to a policy. The policy contains a large number of possible {state, action} pairs, which is difficult to be stored in tabular form. Hence, a policy is commonly represented by a function approximator to overcome this difficulty [60]. The combination of RL and deep leaning (called deep RL [62]) has been successful in handling large-scale complicated tasks by using deep neural networks (DNNs) as function approximators [63], [64], but it is at the expense of complexity. Therefore, this study still considers a RL method and uses a simple neural network (NN) with one fully connected hidden layer to represent the policy (called policy network), as shown in Fig. 3. The three layers of this NN are detailed as follows. The first layer (called input layer) is given the input values, where all the images in the state space are collected as input to this NN. The values of the middle layer (called hidden layer) are a transformation of the input values by a non-linear parametric function [62]. The last layer (called output layer) provides the output values transformed from the hidden layer, which can output an action deciding to accept or reject a new TR. This NN can be trained using the gradient-descent method [57]–[60]. The gradient descent is used to move the policy parameters in a direction that can increase the reward. We can take the gradient of the cumulative discounted reward and update the policy parameters in the direction of the gradient. In this work, the NN is trained by using a variant of the REINFORCE algorithm introduced in [58].

Specifically, the RL agent can schedule multiple TRs at the same time step. In such a case, the action space will select a subset of the TRs from R_m to accept or reject. With each valid action, the required secret-key resources are assigned to the corresponding accepted TRs at the first possible time step using the first-fit algorithm, and then the agent observes a state transition and the new TR is scheduled. After picking a void/invalid action, time proceeds to the next time step and any newly arrived TRs are revealed to the RL agent.

In this work, the discount factor is set to 1 [58], and then the goal of this RL framework is to maximize the cumulative reward

(denoted by *CR*) over time. *CR* is defined as the sum of rewards over time, which can be expressed as:

$$CR = \sum_{t_c=t}^{U} \varepsilon_{t_c} \tag{8}$$

We define the reward ε_{t_c} at the current time step t_c to achieve the joint optimization for BP_{TR} and RU_{SK}, which is expressed as:

$$\varepsilon_{t_c} = -\frac{\sum_{r \in R_u^{t_c}} K_r \cdot t_r}{K \cdot T \cdot |Q|} \tag{9}$$

where $R_u^{t_c}$ is the set of rejected TRs at the current time step t_c. The minimum value of BP_{TR} and maximum value of RU_{SK} can be obtained via maximizing the value of *CR*. This is because when the value of *CR* is maximal, the amount of rejected TRs is minimal, and the rejected TRs demand a minimum number of secret-key resources. Based on the introduced RL framework, an On-MTP algorithm can be automatically trained for QKD networks.

VI. Evaluation and Discussion

In this section, we perform the simulation to comparatively evaluate the heuristics and RL based On-MTP solutions for QKD networks. Two QKD network types are considered in the simulation, i.e., QKD backbone and metropolitan networks. Figure 4 illustrates two realistic network topologies used in the simulation, namely 4-QN QKD backbone network topology (i.e., Beijing-Shanghai QKD network [14]) and 6-QN QKD metropolitan network topology (i.e., SECOQC QKD network [26]). As described in Section III, several intermediate nodes equipped with trusted repeaters have been placed along the QKD links during the network deployment phase, while the amount of intermediate nodes is not considered in this work. For example, the Beijing-Shanghai QKD network topology with 4 QNs and 28 intermediate nodes is used in the simulation, such intermediate nodes are only used for secret-key relay in this study. The simulation is performed with a custom-built Python-based event-driven simulator. This simulator adopts NetworkX [65] to implement the graph representation of network model, and Keras [66] as the machine learning library to implement the policy network in the introduced RL framework. The Python code is written according to the network topology and the formulated network model to construct a QKD network in the simulator.

The simulation settings for the two types of QKD networks are described as follows. The numbers of QKPs over the QKD backbone and metropolitan networks are 6 ($|N| = 4$) and 15 ($|N| = 6$), respectively. In practice, the requests originated from real applications in the currently deployed QKD networks are not public, since the currently deployed QKD networks are typically for very specific purposes (e.g., military and banking) and the requests are highly confidential. The Bernoulli process is a commonly used request distribution for results verification in multiple network scenarios such as the 5G network [57], the computer network [58], and the multi-tenant network [67]. The QKD network is still immature so that it is difficult to obtain a specific request distribution. Hence, we select the Bernoulli process as the request distribution in the simulation for QKD networks, that is, multiple TRs arrive online following a Bernoulli process. The set of secret-key demands of each TR is set to [1, 10] units. The sets of specific QNs corresponding to the secret-key demands of each TR over the QKD backbone and metropolitan networks are set to [2, 4] and [2, 6], respectively. The duration of each TR is uniformly distributed within [5*t*, 10*t*]. The time-step length in each secret-key resource image or secret-key demand image is 20*t*. The secret-key capacity for a QKP at each time step is 20 units. The scheduling capacity for TRs at each time step is 10. The total operation time of each QKD network (i.e., simulation length) is 100*t*. In the introduced RL framework, the NN has a fully connected hidden layer with 20 neurons, while the policy parameters are updated with a learning rate of 0.001 [58]. In the following we evaluate and discuss the comparative results of heuristics and RL based On-MTP solutions for QKD networks (including training and test results), where the training results are obtained using 20 training sets of different TRs and the test results are averaged with 200 times repetition. The simulation runs on a computer with 3.7 GHz Inter Core i7-8700K CPU, 16 GB RAM, and 6 GB NVIDIA GTX 1060 GPU.

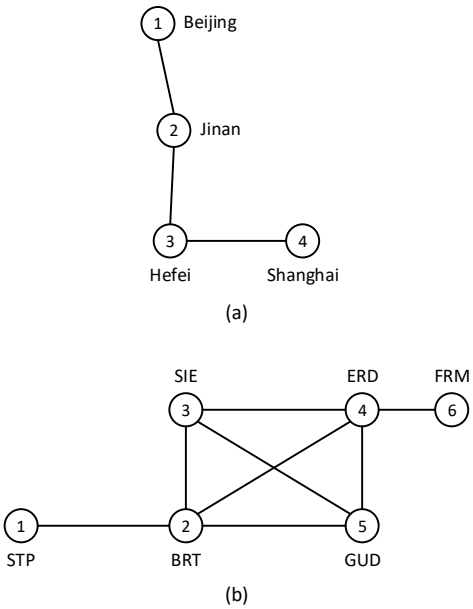

Fig. 4. Network topologies used in simulation: (a) 4-QN QKD backbone network topology (i.e., Beijing-Shanghai QKD network [14]) and (b) 6-QN QKD metropolitan network topology (i.e., SECOQC QKD network [26]).

A. Training Results

The training results of BP_{TR}, RU_{SK}, and *CR* as a function of training iterations for the RL based On-MTP solutions are depicted in Figs. 5–7, respectively. For comparison purpose, we also include the results for three On-MTP heuristics, although training iterations are not relevant.

We consider two QKD network topologies, while the average TR arrival rate (i.e., average number of new TRs arrival at each time step) is set to 1.0. The training results for the RL-based On-MTP solution have been smoothed by deburring. It can be

observed that each of the three performance metrics (i.e., BP_{TR}, RU_{SK}, and CR) as a function of training iterations shows the same variation tendency on the different QKD network topologies. Hence, the scalability of our presented heuristics and RL based On-MTP solutions is demonstrated. In addition, these training results for the three On-MTP heuristics (i.e., random, fit, and BF based On-MTP algorithms) remain constant with the growing training iterations.

As shown in Figs. 5–7, the BP_{TR}, RU_{SK}, and CR for the RL-based On-MTP solution at training iteration 1 demonstrate similarly to that for the random-based On-MTP algorithm, but they are improved (i.e., BP_{TR} decreases, RU_{SK} increases, and CR increases) with the increase of training iterations. From Figs. 5–7 we can see the three performance metrics (i.e., BP_{TR}, RU_{SK}, and CR) for the RL-based On-MTP solution are better than that for the fit-based On-MTP algorithm after 1250 training iterations and better than that for the BF-based On-MTP algorithm after 5000 training iterations for both of the two QKD network topologies. This phenomenon reflects that the RL framework learns to accept more TRs and utilize the secret-key resources more efficiently with the growing training iterations, and consequently the trained RL-based On-MTP algorithm is improved. In particular, the three performance metrics for the RL-based On-MTP solution become stable after 30000 training iterations for both of the two QKD network topologies, indicating that the introduced RL framework converges after 30000th training iteration. When the RL framework becomes converged, the RL-based On-MTP algorithm for obtaining the minimum value of BP_{TR} and the maximum value of RU_{SK} is successfully trained.

Moreover, the average training time for RL-based On-MTP solution is 5.932 seconds per training iteration on the 4-QN QKD backbone network topology, whereas it is 14.654 seconds per training iteration on the 6-QN QKD metropolitan network

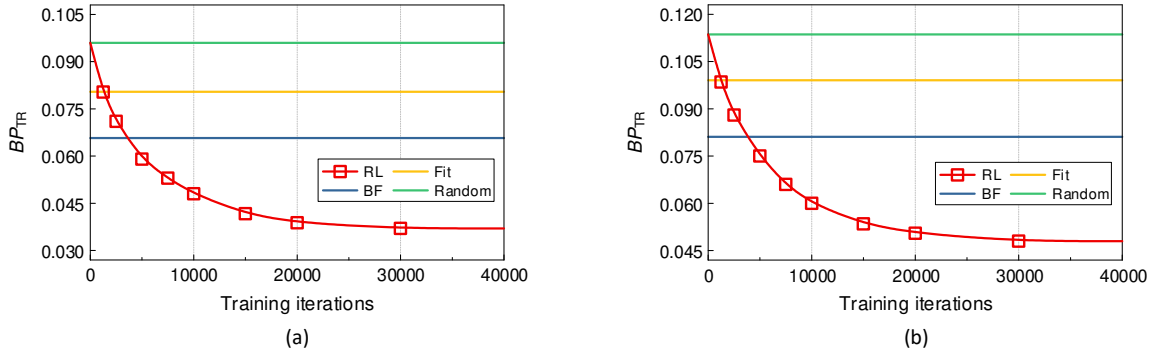

Fig. 5. Training results of BP_{TR} versus training iterations: (a) 4-QN QKD backbone network topology; (b) 6-QN QKD metropolitan network topology.

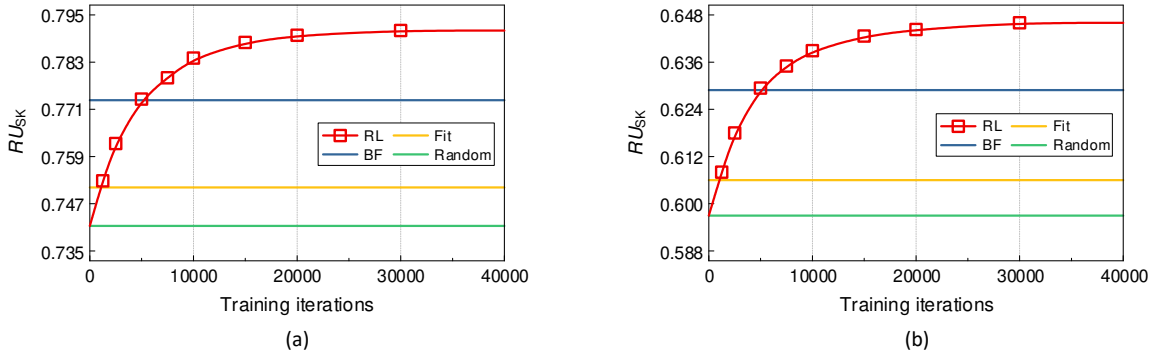

Fig. 6. Training results of RU_{SK} versus training iterations: (a) 4-QN QKD backbone network topology; (b) 6-QN QKD metropolitan network topology.

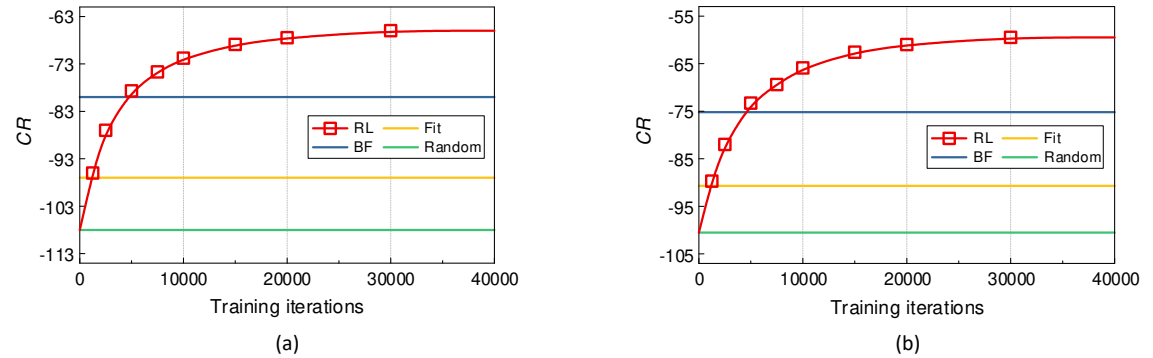

Fig. 7. Training results of CR versus training iterations: (a) 4-QN QKD backbone network topology; (b) 6-QN QKD metropolitan network topology.

topology. The reason is that the numbers of secret-key resource images for QKPs and secret-key demand images for TRs on the 6-QN QKD metropolitan network topology are larger than those on the 4-QN QKD backbone network topology. According to the simulation settings, the number of QKPs and the set of specific QNs corresponding to the secret-key demands of each TR over the 6-QN QKD metropolitan network are larger than those over the 4-QN QKD backbone network. Hence, the training time for the RL-based On-MTP solution increases when the size of a QKD network topology rises and the size of subsets of QNs corresponding to the secret-key demands of each TR becomes larger. In addition, the RL-based On-MTP algorithm needs only be trained once under a given average TR arrival rate for a QKD network, and retraining is required only when the QKD network topology or the average TR arrival rate changes notably.

B. Test Results

The test results of BP_{TR} and RU_{SK} versus average TR arrival rate for the three On-MTP heuristics and the RL-based On-MTP algorithm are compared in Figs. 8 and 9, respectively, where the two QKD network topologies are considered. The values of BP_{TR} for the three On-MTP heuristics and the RL-based On-MTP algorithm are almost equal to 0 when the average TR arrival rate is lower than 0.7, during which the difference of the three On-MTP heuristics and the RL-based On-MTP algorithm is minor. Hence, the average TR arrival rate starts with the value of 0.7 in Figs. 8 and 9 for comparison purpose. It can be seen that BP_{TR} and RU_{SK} for the random, fit, BF, and RL based On-MTP algorithms increase with the rise of average TR arrival rate on both the two QKD network topologies, which results from the increase of the total number of incoming TRs and the total secret-key demands during the operation time of the QKD network.

In terms of BP_{TR} and RU_{SK} on the two QKD network topologies, Figs. 8 and 9 illustrate that the BF-based On-MTP algorithm outperforms the fit-based On-MTP algorithm, and the fit-based On-MTP algorithm outperforms the random-based On-MTP algorithm. The reason is that the random-based On-MTP algorithm randomly schedules a TR, whereas the fit-based On-MTP algorithm only schedules a TR that can fit the free secret-key resources for QKPs, and the BF-based On-MTP algorithm always first schedule a TR that fits best the free secret-key resources for QKPs. In particular, it can be observed that the BP_{TR} and RU_{SK} for the RL-based On-MTP algorithm can outperform the three On-MTP heuristics on the two QKD network topologies. Thus, the scalability and effectiveness of our introduced RL framework for training an On-MTP algorithm are verified.

Tables II and III list the BP_{TR} and RU_{SK} improvements of using the RL-based On-MTP algorithm relative to the three On-MTP heuristics on both QKD network topologies, respectively. The BP_{TR} and RU_{SK} improvements of the RL-based On-MTP algorithm versus the three On-MTP heuristics vary as average TR arrival rate changes. As illustrated

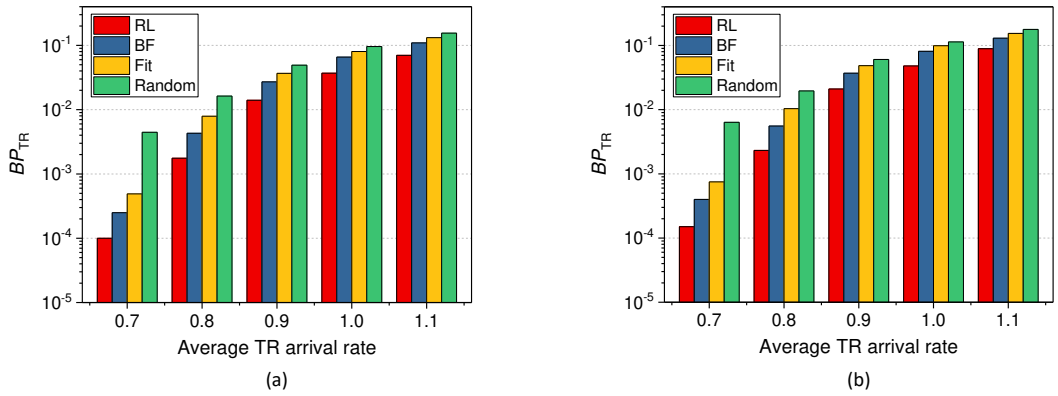

Fig. 8. Test results of BP_{TR} versus average TR arrival rate: (a) 4-QN QKD backbone network topology; (b) 6-QN QKD metropolitan network topology.

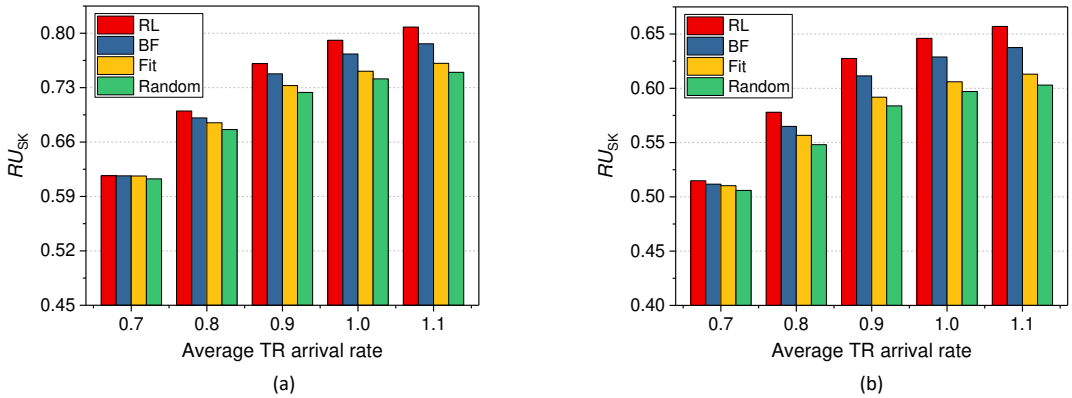

Fig. 9. Test results of RU_{SK} versus average TR arrival rate: (a) 4-QN QKD backbone network topology; (b) 6-QN QKD metropolitan network topology.

TABLE II
BP_{TR} IMPROVEMENTS OF RL-BASED ON-MTP ALGORITHM VERSUS THREE ON-MTP HEURISTICS

	4-QN QKD backbone network topology			6-QN QKD metropolitan network topology		
Average TR arrival rate	RL vs. BF (%)	RL vs. Fit (%)	RL vs. Random (%)	RL vs. BF (%)	RL vs. Fit (%)	RL vs. Random (%)
0.7	60.0	79.6	97.8	62.5	80.0	97.6
0.8	59.0	77.7	89.2	58.2	77.6	88.2
0.9	48.3	61.9	71.5	43.2	56.6	65.3
1.0	43.7	54.0	61.4	40.8	51.5	57.7
1.1	35.9	47.0	54.9	31.5	42.2	50.0

TABLE III
RU_{SK} IMPROVEMENTS OF RL-BASED ON-MTP ALGORITHM VERSUS THREE ON-MTP HEURISTICS

	4-QN QKD backbone network topology			6-QN QKD metropolitan network topology		
Average TR arrival rate	RL vs. BF (%)	RL vs. Fit (%)	RL vs. Random (%)	RL vs. BF (%)	RL vs. Fit (%)	RL vs. Random (%)
0.7	0.02	0.06	0.67	0.61	0.89	1.76
0.8	1.28	2.22	3.54	2.31	3.82	5.46
0.9	1.77	3.87	5.13	2.62	6.03	7.49
1.0	2.29	5.31	6.70	2.72	6.60	8.21
1.1	2.73	6.13	7.77	3.05	7.18	8.96

in Table II, the BP_{TR} improvements of using the RL-based On-MTP algorithm relative to the three On-MTP heuristics decrease with the average TR arrival rate rising on the two QKD network topologies. For example, when the average TR arrival rate is 0.7, the BP_{TR} improvements of using RL-based On-MTP algorithm are up to 60.0%, 79.6%, and 97.8% compared to the BF, fit, and random based On-MTP algorithms respectively on the 4-QN QKD backbone network topology, and 62.5%, 80.0%, and 97.6% respectively on the 6-QN QKD metropolitan network topology.

In contrast, according to Table III, the RU_{SK} improvements of the RL-based On-MTP algorithm versus the three On-MTP heuristics increase with the rise of average TR arrival rate on the two QKD network topologies. For instance, when the average TR arrival rate is 1.1, the RU_{SK} improvements of the RL-based On-MTP algorithm are up to 2.73%, 6.13%, and 7.77% versus the BF, fit, and random based On-MTP algorithms respectively on the 4-QN QKD backbone network topology, and they are up to 3.05%, 7.18%, and 8.96% respectively on the 6-QN QKD metropolitan network topology.

Therefore, the RL-based On-MTP algorithm can effectively obtain better BP_{TR} and RU_{SK} than the three On-MTP heuristics, since the introduced RL framework gradually learns and trains a more efficient On-MTP algorithm to achieve the better performance.

VII. CONCLUSION

In this paper, we address the On-MTP problem for QKD networks, where the scheduling of multiple TRs and the assignment of non-reusable secret keys to TRs are considered. The secret-key resources for QKPs and the secret-key demands of TRs are modeled with distinct images. A comparative study of heuristics and RL based On-MTP solutions for two types of QKD networks is performed, where the three On-MTP heuristics are presented and an On-MTP algorithm is trained using the introduced RL framework. Simulation results demonstrate the scalability and effectiveness of our presented heuristics and RL based On-MTP solutions. Based on the comparative evaluation of BP_{TR} and RU_{SK}, the BF-based On-MTP algorithm performs the best for On-MTP among all three considered heuristics. Moreover, the RL-based On-MTP algorithm can significantly outperform any tested heuristics, indicating that the introduced RL framework gradually learns and trains a more efficient On-MTP algorithm to optimize performance. We realize different QKD network topologies or different request distributions can lead to different results. Therefore, in the future given the requests originated from real applications in the practical QKD networks are available to be accessed we will also perform the real testing to verify the formulated network model, and carry out the comparative evaluation of the presented heuristics and the introduced RL framework in this work.

REFERENCES

[1] Google Cloud Security Whitepapers [Online]. Available: https://services.google.com/fh/files/misc/security_whitepapers_march2018.pdf.

[2] L. Gong, L. Zhang, W. Zhang, X. Li, X. Wang, and W. Pan, "The application of data encryption technology in computer network communication security," *AIP Conf. Proc.*, vol. 1834, no. 1, Apr. 2017, Art. no. 040027.

[3] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[4] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, CRC Press, 2014.

[5] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annual Symposium on Foundations of Computer Science*, Santa Fe, NM, USA, Nov. 1994, pp. 124–134.

[6] L. R. Schreiber and H. Bluhm, "Toward a silicon-based quantum computer," *Science*, vol. 359, no. 6374, pp. 393–394, Jan. 2018.

[7] L. Gyongyosi and S. Imre, "A survey on quantum computing technology," *Comput. Sci. Rev.*, vol. 31, pp. 51–71, Feb. 2019.

[8] H.-K. Lo, M. Curty, and K. Tamaki, "Secure quantum key distribution," *Nature Photon.*, vol. 8, no. 8, pp. 595–604, Aug. 2014.

[9] E. Diamanti, H.-K. Lo, B. Qi, and Z. Yuan, "Practical challenges in quantum key distribution," *npj Quantum Inform.*, vol. 2, no. 1, Nov. 2016, Art. no. 16025.

[10] L. Gyongyosi, L. Bacsardi, and S. Imre, "A survey on quantum key distribution," *Infocommun. J.*, vol. XI, no. 2, pp. 14–21, June 2019.

[11] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dusek, N. Lutkenhaus, and M. Peev, "The security of practical quantum key distribution," *Rev. Mod. Phys.*, vol. 81, no. 3, pp. 1301–1350, Sept. 2009.

[12] L. Gyongyosi, S. Imre, and H. V. Nguyen, "A survey on quantum channel capacities," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 1149–1205, Jan. 2018.

[13] S.-K. Liao, W.-Q. Cai, J. Handsteiner, B. Liu, J. Yin, L. Zhang, D. Rauch, M. Fink, J.-G. Ren, W.-Y. Liu, Y. Li, Q. Shen, Y. Cao, F.-Z. Li, J.-F. Wang, Y.-M. Huang, L. Deng, T. Xi, L. Ma, T. Hu, L. Li, N.-L. Liu, F. Koidl, P. Wang, Y.-A. Chen, X.-B. Wang, M. Steindorfer, G. Kirchner, C.-Y. Lu, R. Shu, R. Ursin, T. Scheidl, C.-Z. Peng, J.-Y. Wang, A. Zeilinger, and J.-W. Pan, "Satellite-relayed intercontinental quantum network," *Phys. Rev. Lett.*, vol. 120, no. 3, Jan. 2018, Art. no. 030501.

[14] Q. Zhang, F. Xu, Y.-A. Chen, C.-Z. Peng, and J.-W. Pan, "Large scale quantum key distribution: Challenges and solutions [Invited]," *Opt. Express*, vol. 26, no. 18, pp. 24260–24273, Sept. 2018.

[15] Y. Zhao, B. Qi, X. Ma, H.-K. Lo, and L. Qian, "Experimental quantum key distribution with decoy states," *Phys. Rev. Lett.*, vol. 96, no. 7, Feb. 2006, Art. no. 070502.

[16] B. Korzh, C. C. W. Lim, R. Houlmann, N. Gisin, M. J. Li, D. Nolan, B. Sanguinetti, R. Thew, and H. Zbinden, "Provably secure and practical quantum key distribution over 307 km of optical fibre," *Nature Photon.*, vol. 9, no. 3, pp. 163–168, Feb. 2015.

[17] H.-L. Yin, T.-Y. Chen, Z.-W. Yu, H. Liu, L.-X. You, Y.-H. Zhou, S.-J. Chen, Y. Mao, M.-Q. Huang, W.-J. Zhang, H. Chen, M. J. Li, D. Nolan, F. Zhou, X. Jiang, Z. Wang, Q. Zhang, X.-B. Wang, and J.-W. Pan, "Measurement-device-independent quantum key distribution over a 404 km optical fiber," *Phys. Rev. Lett.*, vol. 117, no. 19, Nov. 2016, Art. no. 190501.

[18] B. Fröhlich, M. Lucamarini, J. F. Dynes, L. C. Comandar, W. W.-S. Tam, A. Plews, A. W. Sharpe, Z. Yuan, and A. J. Shields, "Long-distance quantum key distribution secure against coherent attacks," *Optica*, vol. 4, no. 1, pp. 163–167, Jan. 2017.

[19] L. Gyongyosi and S. Imre, "Multiple access multicarrier continuous-variable quantum key distribution," *Chaos, Solitons & Fractals*, vol. 114, pp. 491–505, Sept. 2018.

[20] L. Gyongyosi and S. Imre, "Low-dimensional reconciliation for continuous-variable quantum key distribution," *Appl. Sci.*, vol. 8, no. 1, Jan. 2018, Art. no. 87.

[21] L. Gyongyosi and S. Imre, "Secret key rate proof of multicarrier continuous-variable quantum key distribution," *Int. J. Commun. Syst.*, vol. 32, no. 4, Mar. 2019, Art. no. e3865.

[22] L. Gyongyosi, "Singular value decomposition assisted multicarrier continuous-variable quantum key distribution," *Theor. Comput. Sci.*, vol. 801, pp. 35–63, Jan. 2020.

[23] N. Hosseinidehaj, Z. Babar, R. Malaney, S. X. Ng, and L. Hanzo, "Satellite-based continuous-variable quantum communications: State-of-the-art and a predictive outlook," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 881–919, Feb. 2019.

[24] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," in *Proc. IEEE Int. Conf. Comput. Syst. Signal Process.*, Bangalore, India, 1984, pp. 175–179.

[25] F. Grosshans and P. Grangier, "Continuous variable quantum cryptography using coherent states," *Phys. Rev. Lett.*, vol. 88, no. 5, Jan. 2002, Art. no. 057902.

[26] M. Peev, C. Pacher, R. Alléaume, C. Barreiro, J. Bouda, W. Boxleitner, T. Debuisschert, E. Diamanti, M. Dianati, J. F. Dynes, S. Fasel, S. Fossier, M. Fürst, J.-D. Gautier, O. Gay, N. Gisin, P. Grangier, A. Happe, Y. Hasani, M. Hentschel, H. Hübel, G. Humer, T. Länger, M. Legré, R. Lieger, J. Lodewyck, T. Lorünser, N. Lütkenhaus, A. Marhold, T. Matyus, O. Maurhart, L. Monat, S. Nauerth, J.-B. Page, A. Poppe, E. Querasser, G. Ribordy, S. Robyr, L. Salvail, A. W. Sharpe, A. J. Shields, D. Stucki, M. Suda, C. Tamas, T. Themel, R. T. Thew, Y. Thoma, A. Treiber, P. Trinkler, R. Tualle-Brouri, F. Vannel, N. Walenta, H. Weier, H. Weinfurter, I. Wimberger, Z. L. Yuan, H. Zbinden, and A. Zeilinger, "The SECOQC quantum key distribution network in Vienna," *New J. Phys.*, vol. 11, no. 7, July 2009, Art. no. 075001.

[27] M. Sasaki, M. Fujiwara, H. Ishizuka, W. Klaus, K. Wakui, M. Takeoka, A. Tanaka, K. Yoshino, Y. Nambu, S. Takahashi, A. Tajima, A. Tomita, T. Domeki, T. Hasegawa, Y. Sakai, H. Kobayashi, T. Asai, K. Shimizu, T. Tokura, T. Tsurumaru, M. Matsui, T. Honjo, K. Tamaki, H. Takesue, Y. Tokura, J. F. Dynes, A. R. Dixon, A. W. Sharpe, Z. L. Yuan, A. J. Shields, S. Uchikoga, M. Legré, S. Robyr, P. Trinkler, L. Monat, J.-B. Page, G. Ribordy, A. Poppe, A. Allacher, O. Maurhart, T. Länger, M. Peev, and A. Zeilinger, "Field test of quantum key distribution in the Tokyo QKD Network," *Opt. Express*, vol. 19, no. 11, pp. 10387–10409, May 2011.

[28] D. Stucki, M. Legre, F. Buntschu, B. Clausen, N. Felber, N. Gisin, L. Henzen, P. Junod, G. Litzistorf, P. Monbaron, L. Monat, J.-B. Page, D. Perroud, G. Ribordy, A. Rochas, S. Robyr, J. Tavares, R. Thew, P. Trinkler, S. Ventura, R. Voirol, N. Walenta, and H. Zbinden, "Long-term performance of the SwissQuantum quantum key distribution network in a field environment," *New J. Phys.*, vol. 13, no. 12, Dec. 2011, Art. no. 123001.

[29] S. Wang, W. Chen, Z.-Q. Yin, H.-W. Li, D.-Y. He, Y.-H. Li, Z. Zhou, X.-T. Song, F.-Y. Li, D. Wang, H. Chen, Y.-G. Han, J.-Z. Huang, J.-F. Guo, P.-L. Hao, M. Li, C.-M. Zhang, D. Liu, W.-Y. Liang, C.-H. Miao, P. Wu, G.-C. Guo, and Z.-F. Han, "Field and long-term demonstration of a wide area quantum key distribution network," *Opt. Express*, vol. 22, no. 18, pp. 21739–21756, Sept. 2014.

[30] B. Qi, W. Zhu, L. Qian, and H.-K. Lo, "Feasibility of quantum key distribution through a dense wavelength division multiplexing network," *New J. Phys.*, vol. 12, no. 10, Oct. 2010, Art. no. 103042.

[31] S. Aleksic, F. Hipp, D. Winkler, A. Poppe, B. Schrenk, and G. Franzl, "Perspectives and limitations of QKD integration in metropolitan area networks," *Opt. Express*, vol. 23, no. 8, pp. 10359–10373, Apr. 2015.

[32] Y. Cao, Y. Zhao, X. Yu, and Y. Wu, "Resource assignment strategy in optical networks integrated with quantum key distribution," *J. Opt. Commun. Netw.*, vol. 9, no. 11, pp. 995–1004, Nov. 2017.

[33] Y. Mao, B.-X. Wang, C. Zhao, G. Wang, R. Wang, H. Wang, F. Zhou, J. Nie, Q. Chen, Y. Zhao, Q. Zhang, J. Zhang, T.-Y. Chen, and J.-W. Pan, "Integrating quantum key distribution with classical communications in backbone fiber network," *Opt. Express*, vol. 26, no. 5, pp. 6010–6020, Mar. 2018.

[34] F. Karinou, H. H. Brunner, C.-H. F. Fung, L. C. Comandar, S. Bettelli, D. Hillerkuss, M. Kuschnerov, S. Mikroulis, D. Wang, C. Xie, M. Peev, and A. Poppe, "Toward the integration of CV quantum key distribution in deployed optical networks," *IEEE Photon. Technol. Lett.*, vol. 30, no. 7, pp. 650–653, Apr. 2018.

[35] Y. Zhao, Y. Cao, W. Wang, H. Wang, X. Yu, J. Zhang, M. Tornatore, Y. Wu, and B. Mukherjee, "Resource allocation in optical networks secured by quantum key distribution," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 130–137, Aug. 2018.

[36] T. A. Eriksson, T. Hirano, B. J. Puttnam, G. Rademacher, R. S. Luís, M. Fujiwara, R. Namiki, Y. Awaji, M. Takeoka, N. Wada, and M. Sasaki, "Wavelength division multiplexing of continuous variable quantum key distribution and 18.3 Tbit/s data channels," *Commun. Phys.*, vol. 2, no. 1, Jan. 2019, Art. no. 9.

[37] Y. Cao, Y. Zhao, X. Yu, and J. Zhang, "Multi-tenant provisioning over software defined networking enabled metropolitan area quantum key distribution networks," *J. Opt. Soc. Am. B*, vol. 36, no. 3, pp. B31–B40, Mar. 2019.

[38] Y. Cao, Y. Zhao, R. Lin, X. Yu, J. Zhang, and J. Chen, "Multi-tenant secret-key assignment over quantum key distribution networks," *Opt. Express*, vol. 27, no. 3, pp. 2544–2561, Feb. 2019.

[39] Y. Cao, Y. Zhao, J. Li, R. Lin, J. Zhang, and J. Chen, "Reinforcement learning based multi-tenant secret-key assignment for quantum key distribution networks," in *Proc. Opt. Fiber Commun. Conf.*, San Diego, CA, USA, Mar. 2019, Paper M2A.7.

[40] Y. Cao, Y. Zhao, C. Colman-Meixner, X. Yu, and J. Zhang, "Key on demand (KoD) for software-defined optical networks secured by quantum key distribution (QKD)," *Opt. Express*, vol. 25, no. 22, pp. 26453–26467, Oct. 2017.

[41] Y. Cao, Y. Zhao, Y. Wu, X. Yu, and J. Zhang, "Time-scheduled quantum key distribution (QKD) over WDM networks," *J. Lightwave Technol.*, vol. 36, no. 16, pp. 3382–3395, Aug. 2018.

[42] N. Sangouard, C. Simon, H. de Riedmatten, and N. Gisin, "Quantum repeaters based on atomic ensembles and linear optics," *Rev. Mod. Phys.*, vol. 83, no. 1, pp. 33–80, Mar. 2011.

[43] S. Bahrani, M. Razavi, and J. A. Salehi, "Wavelength assignment in hybrid quantum-classical networks," *Sci. Rep.*, vol. 8, no. 1, Feb. 2018, Art. no. 3456.

[44] Y. Ou, E. Hugues-Salas, F. Ntavou, R. Wang, Y. Bi, S. Yan, G. Kanellos, R. Nejabati, and D. Simeonidou, "Field-trial of machine learning-assisted quantum key distribution (QKD) networking with SDN," in *Proc. Europ. Conf. Opt. Commun.*, Rome, Italy, Sept. 2018.

[45] H. Wang, Y. Zhao, X. Yu, Z. Ma, J. Wang, A. Nag, L. Yi, and J. Zhang, "Protection schemes for key service in optical networks secured by quantum key distribution (QKD)," *J. Opt. Commun. Netw.*, vol. 11, no. 3, pp. 67–78, Mar. 2019.

[46] H. Wang, Y. Zhao, X. Yu, B. Chen, and J. Zhang, "Resilient fiber-based quantum key distribution (QKD) networks with secret-key re-allocation strategy," in *Proc. Opt. Fiber Commun. Conf.*, San Diego, CA, USA, Mar. 2019, Paper W2A.25.

[47] Y. Cao, Y. Zhao, J. Wang, X. Yu, Z. Ma, and J. Zhang, "Cost-efficient quantum key distribution (QKD) over WDM networks," *J. Opt. Commun. Netw.*, vol. 11, no. 6, pp. 285–298, June 2019.

[48] A. Aguado, E. Hugues-Salas, P. A. Haigh, J. Marhuenda, A. B. Price, P. Sibson, J. E. Kennard, C. Erven, J. G. Rarity, M. G. Thompson, A. Lord, R. Nejabati, and D. Simeonidou, "Secure NFV orchestration over an SDN-controlled optical network with time-shared quantum key distribution resources," *J. Lightwave Technol.*, vol. 35, no. 8, pp. 1357–1362, Apr. 2017.

[49] E. Hugues-Salas, F. Ntavou, D. Gkounis, G. T. Kanellos, R. Nejabati, and D. Simeonidou, "Monitoring and physical-layer attack mitigation in SDN-controlled quantum key distribution networks," *J. Opt. Commun. Netw.*, vol. 11, no. 2, pp. A209–A218, Feb. 2019.

[50] Y. Cao, Y. Zhao, X. Yu, L. Cheng, Z. Li, G. Liu, and J. Zhang, "Experimental demonstration of end-to-end key on demand service provisioning over quantum key distribution networks with software defined networking," in *Proc. Opt. Fiber Commun. Conf.*, San Diego, CA, USA, Mar. 2019, Paper Th1G.4.

[51] Y. Cao, Y. Zhao, J. Wang, X. Yu, Z. Ma, and J. Zhang, "SDQaaS: software defined networking for quantum key distribution as a service," *Opt. Express*, vol. 27, no. 5, pp. 6892–6909, Mar. 2019.

[52] A. Aguado, V. Lopez, J. Martinez-Mateo, T. Szyrkowiec, A. Autenrieth, M. Peev, D. Lopez, and V. Martin, "Hybrid conventional and quantum security for software defined and virtualized networks," *J. Opt. Commun. Netw.*, vol. 9, no. 10, pp. 819–825, Oct. 2017.

[53] A. Mavromatis, F. Ntavou, E. Hugues-Salas, G. T. Kanellos, R. Nejabati, and D. Simeonidou, "Experimental demonstration of quantum key distribution (QKD) for energy-efficient software-defined Internet of Things," in *Proc. Europ. Conf. Opt. Commun.*, Rome, Italy, Sept. 2018.

[54] R. Nejabati, R. Wang, A. Bravalheri, A. Muqaddas, N. Uniyal, T. Diallo, R. Tessinari, R. S. Guimaraes, S. Moazzeni, E. Hugues-Salas, G. T. Kanellos, and D. Simeonidou, "First demonstration of quantum-secured, inter-domain 5G service orchestration and on-demand NFV chaining over flexi-WDM optical networks," in *Proc. Opt. Fiber Commun. Conf.*, San Diego, CA, USA, Mar. 2019, Paper Th4C.6.

[55] Y. Cao, Y. Zhao, J. Wang, X. Yu, Z. Ma, and J. Zhang, "KaaS: Key as a service over quantum key distribution integrated optical networks," *IEEE Commun. Mag.*, vol. 57, no. 5, pp. 152–159, May 2019.

[56] C. E. Shannon, "Communication theory of secrecy systems," *Bell Labs Tech. J.*, vol. 28, no. 4, pp. 656–715, Oct. 1949.

[57] M. R. Raza, C. Natalino, P. Öhlen, L. Wosinska, and P. Monti, "A slice admission policy based on reinforcement learning for a 5G flexible RAN," in *Proc. Europ. Conf. Opt. Commun.*, Rome, Italy, Sept. 2018.

[58] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in *Proc. 15th ACM Workshop on Hot Topics in Networks*, Atlanta, GA, USA, Nov. 2016.

[59] M. R. Raza, C. Natalino, P. Öhlen, L. Wosinska, and P. Monti, "Reinforcement learning for slicing in a 5G flexible RAN," *J. Lightwave Technol.*, vol. 37, no. 20, pp. 5161–5169, Oct. 2019.

[60] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, Cambridge, MA: MIT Press, 2018.

[61] J. R. Norris, *Markov Chains*, Cambridge University Press, 1998.

[62] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, *An Introduction to Deep Reinforcement Learning*, Now Foundations and Trends, 2018.

[63] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[64] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

[65] NetworkX [Online]. Available: https://networkx.github.io/.

[66] Keras [Online]. Available: https://keras.io/.

[67] C. Natalino, M. R. Raza, A. Rostami, P. Öhlen, L. Wosinska, and P. Monti, "Machine learning aided orchestration in multi-tenant networks," in *Proc. IEEE Photon. Soc. Summer Top. Meeting Ser.*, Waikoloa Village, HI, USA, July 2018, pp. 125–126.