Dynamic Routing and Post-processing Strategies for Hybrid Quantum Key Distribution Networks

Omar Amer*, Walter O. Krawec†, Md Zakir Hossain†, Victoria U. Manfredi‡, Bing Wang†

*Global Technology Applied Research, JPMorgan Chase Bank, New York City, NY, United States
†School of Computing, University of Connecticut, Storrs, CT, United States
‡Department of Mathematics and Computer Science, Wesleyan University, Middletown, CT, United States

***Abstract*—In this paper, we consider hybrid quantum key distribution (QKD) networks with primarily quantum repeaters and a small number of trusted nodes. While the trusted nodes need to be trusted, the use of such nodes, together with efficient routing algorithms, can significantly improve the key generation rate. We show that when trusted nodes are placed at asymmetric locations relative to Alice and Bob, however, existing routing algorithms can lead to low key rate. To address this issue, we develop dynamic routing strategies that adjust routing decisions based on the current key pool conditions in the network. In addition, we investigate new and existing classical post-processing techniques that complement the dynamic routing strategies. Using extensive simulations, we show that our dynamic routing strategies can significantly outperform static strategies, and the post-processing techniques are beneficial in high noise scenarios. In addition, combining the dynamic routing strategies and post-processing techniques can further improve the overall key rate.**

I. Introduction

Quantum key distribution (QKD) allows two parties, Alice and Bob, to establish a shared secret key, secure against computationally unbounded adversaries [1]. Despite the great potential of QKD, however, numerous challenges remain. Of particular importance is improving the overall key generation rate while also supporting large distances between two parties. Quantum networks provide a promising direction in improving key generation rate over long distance.

To date, the majority of research in quantum networks spans two directions: the study of networks consisting of only trusted nodes (TNs) [2]–[13], and the study of networks consisting of only quantum repeaters, i.e., a true quantum internet [14]–[16]. These two directions represent the two extreme end-points of the timeline in quantum network development. Quantum repeaters allow for two end-points to perform entanglement swapping, allowing them to establish Bell pairs even if the repeater is controlled by an adversary. This provides a tremendously strong security guarantee. However, the technological requirements of a quantum repeater still surpass today's engineering capabilities, especially in terms of their need for short-term quantum memories. TNs, on the other hand, are available today and are used in most of the current large-scale metro-area QKD networks (e.g., Vienna [17], Hefei [18], Beijing-Shanghai [19], and Tokyo [20]). This is due to the fact that TNs simply behave like Alice or Bob, and perform actual key distribution. That is, they do not require any quantum storage but only classical storage. The downside of TNs, as their name implies, is that they must be trusted as they will have full information on the final secret key.

In this paper, we consider *hybrid* near-future quantum networks that have a majority of quantum repeaters and only a small number of TNs that are physically secure (or secured through other means). The reason for focusing on such networks is because they achieve a better tradeoff in terms of security and performance, compared to other alternatives such as quantum-repeater-only or TN-only networks, or a network with majority as TNs and a small number of repeaters. First, the security of such QKD networks follows from the strong security guarantee of the quantum repeaters, the physical security of the TNs, and the perfect security of using one-time pad for transferring secret keys among TNs (see Section II). Second, such networks are efficient and practical. The study in [21] shows that even a small number of TNs can significantly improve the key generation rate compared to a quantum-repeater-only network, and securing a small number of TNs incurs much less cost than securing a large number of TNs in a TN-only network or a network with majority nodes as TNs. Overall, the above hybrid QKD networks, though not as powerful as the true future quantum internet, provide a realistic middle-ground between today's TN-based QKD networks and the true future quantum internet.

In such hybrid quantum networks, the placement of the TNs can significantly affect the key generation rate. TNs may not always be placed at symmetric locations between Alice and Bob, as assumed in [21]. This is because a quantum network may reuse an existing optical fiber infrastructure, and the locations of the TNs may be constrained by the need to provide physical security. In addition, the locations of Alice and Bob may change over time. We show that the routing algorithm proposed in [21] does not perform well under asymmetric TN placement. To address this issue, we develop *dynamic routing strategies* that adjust routing decisions based on the current key pool conditions in the network. The dynamic routing strategies, however, can lead to long and more diverse network paths. We therefore further investigate new and existing classical post-processing techniques to complement the above dynamic routing strategies. Our work makes the following main contributions:

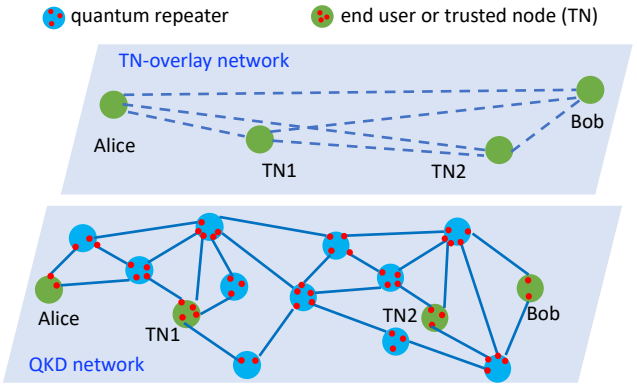


Fig. 1. Hybrid network architecture with quantum repeaters and a small number of TNs. The lower part of the figure shows the QKD network, and the upper part shows the TN-overlay network.

• We develop dynamic routing strategies for hybrid quantum networks that contains a majority of quantum repeaters and a small number of TNs (§III). This strategy runs on top of an existing routing algorithm and makes routing decisions in each network round based on the current key pool conditions, specifically, the current amount of secret key bits among the users and TNs, to maximize key generation rate for end users.

• We propose a new classical post-processing technique involving dynamic pooling of raw key bits into distinct blocks based on network characteristics, e.g., path lengths (§IV). In addition, we employ existing classical advantage distillation (CAD) [22] techniques, adapting the CAD parameters appropriately for each distinct key pool. These post-processing techniques are broadly relevant to other QKD systems, not only hybrid QKD networks.

• We perform extensive evaluation of the dynamic routing strategies and post-processing techniques in a wide range of network scenarios (§V and §VI). Our results show that the dynamic routing strategies can significantly outperform the standard static strategy, and the post-processing techniques are beneficial in high noise scenarios. In addition, combining the dynamic routing strategies and the post-processing techniques can further improve the overall key rate. As an example, we show that, with the post-processing techniques, the dynamic routing strategies can improve the key rate up to 23% over the static strategy in random graph networks.

II. Network Model

We consider a hybrid quantum network as illustrated in the lower part of Fig. 1. This network is modeled as a graph with each node representing either an end user (Alice or Bob who wish to establish a shared secret key), a TN, or a quantum repeater. End users and TNs are denoted $T_0, T_1, \cdots, T_n, T_{n+1}$, with T_0 being Alice, T_{n+1} being Bob, and all other T_i being TNs. Each pair T_i and T_j have a *raw key pool*, denoted $\mathcal{RK}_{i,j}$, and a *secret key pool*, denoted $\mathcal{SK}_{i,j}$, storing shared key material between those nodes. The raw key consists of shared key bits *before* classical post-processing steps (error correction and privacy amplification) are run, whereas the secret key consists of secret key bits following the completion of these classical post-processing steps.

Each edge in this graph connecting two nodes represents a fiber link allowing for the transmission of qubits between nodes; this link may be noisy and lossy. We assume that all nodes have a quantum memory capable of storing a single qubit for every adjacent node. But this qubit can be stored only for a short amount of time due to short quantum memory coherence time [23], after which it must be discarded. In particular, the qubit can only be stored for a single network round; see later. The memory may also be noisy, causing the state to decohere with a certain probability.

As in [21], [24], [25], we assume the nodes in the network have synchronized clocks and the network operates in *rounds*. Each round is divided into three *stages*, for completing the quantum portion of QKD over the quantum network. After a sufficient number of rounds, classical post-processing is conducted to obtain secret key between Alice and Bob. We next briefly describe these steps.

Stage 1 (link-level entanglement establishment). In this stage, each pair of connected nodes, u and v, in the quantum network attempts to share half of an entangled pair with each other over the fiber link. This process succeeds with probability $P = 10^{-\alpha L/10}$, where L is the length of the fiber link connecting u and v, and α is a constant [26]–[28]. Otherwise, the qubit is lost and no shared entanglement is established in this round. In addition, even when entanglement is established successfully, we assume with probability D, the entanglement may depolarize and become a completely mixed state, due to noise in the link, memory system, or both. Thus, at the end of this stage, with probability P, each pair of neighbors u, v share the state

$$\rho_{u,v} = (1-D)\left|\Phi^+\right\rangle\left\langle\Phi^+\right| + D\frac{I}{4}, \tag{1}$$

where $\left|\Phi^+\right\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right)$ and I is the identity matrix. Otherwise, with probability $1-P$, those neighbors share the vacuum state (i.e., they share no state). We assume nodes are able to determine whether they have a vacuum or not, while they cannot determine if the state is the desired Bell state $\left|\Phi^+\right\rangle$ or a completely mixed state.

Stage 2 (entanglement routing). In this stage, a routing algorithm is executed (see Section III) to create end-to-end entanglement between nodes T_i and T_j. Specifically, the routing algorithm will determine a set of paths, each ending with T_i and T_j, and has only quantum repeaters as interior nodes. These paths will be used by the repeaters to attempt to create an entangled pair between T_i and T_j by performing Bell statement measurement (BSM), also called entanglement swapping, at each repeater in the path. If successful, this will create a long-distance end-to-end shared entangled state between T_i and T_j. Since two different paths cannot rely on a single qubit, these paths must be disjoint in their use of the available stored qubits. As in the first stage, we assume nodes T_i and T_j will know whether this path entanglement attempt

was successful or not; however, they do not know if the state is the desired Bell state $|\Phi^+\rangle$ or not.

In more detail, given a path $T_i \to u_1 \to \cdots \to u_\ell \to T_j$ consisting of ℓ repeaters, each repeater, u_i, will perform a BSM on its respective half of the shared state $\rho_{u_i,u_{i+1}}$, see Eq. (1). This succeeds, independently for each repeater, with probability R. Therefore, with probability R^ℓ, nodes T_i and T_j share a state of the form:

$$\rho_{T_i,T_j} = (1-D)^{\ell+1} |\Phi^+\rangle \langle\Phi^+| + \left[1-(1-D)^{\ell+1}\right] \frac{I}{4} \quad (2)$$

where D is decoherence rate as described in Stage 1. Otherwise, with probability $1-R^\ell$, nodes T_i and T_j do not share a quantum state from this path.

Note that, unlike routing in entanglement networks, here we have several interesting optimization decisions unique to our scenario. Primarily, our goal is to maximize the secret key generation rate between T_0 and T_{n+1}. Maximizing the number of shared states or raw key bits between any T_i and T_j does not necessarily lead to maximal secret key generation rate between T_0 and T_{n+1}, as we will show later. Thus, new methods are required to fully harness the capabilities of this network. In Section III, we show that static routing strategies such as in existing work [21] are not sufficient when the TNs are placed at asymmetric locations; we then develop dynamic routing strategies that outperform static strategies.

Stage 3 (raw key generation). In this stage, with entanglement shared, each pair T_i and T_j can conduct the quantum portion of their chosen entanglement based QKD protocol, in our case the E91 protocol [29]. If the two parties select the same randomly chosen basis, which occurs with probability $1/2$, the two parties add the resultant raw key bit to their raw key pool $\mathcal{RK}_{i,j}$. Otherwise, no addition to their raw key pools is made. At the end of this stage, all remaining qubits in the system are discarded, as we assume quantum memories can only store qubits for a single network round. The network then repeats the above three stages for the next round.

Post-processing. When the network has been running for a sufficient number of rounds, post-processing is executed. Here, error correction and privacy amplification are run between nodes T_i and T_j, adding secret key material to $\mathcal{SK}_{i,j}$. As we are interested only in the asymptotic performance in this paper, we assume perfect error correction and use asymptotic key rates for the E91 protocol. In particular, this means if the error rate in $\mathcal{RK}_{i,j}$ is $Q_{i,j}$, then $|\mathcal{SK}_{i,j}| = |\mathcal{RK}_{i,j}|(1-2h(Q_{i,j}))$ [30], [31], where $h(x) = -x\log x - (1-x)\log(1-x)$ is the binary entropy function.

Finally, since the ultimate goal of the network is for Alice and Bob to share as much secret key material as possible, this last step is to push secret key material from each $\mathcal{SK}_{i,j}$ to $\mathcal{SK}_{0,n+1}$ (i.e., the secret key pools of Alice and Bob). For this purpose, we consider the *TN-overlay network* as illustrated in the upper part of Fig. 1. We refer to it as TN-overlay network since it only consists of the TNs and end users (Alice and Bob), and is an overlay on top of the quantum network. In the TN-overlay network, the capacity between parties i, j is the number of secret key bits, $|\mathcal{SK}_{i,j}|$, between them. As in [21], we can then use standard max-flow techniques [32] to find a maximal flow of secret key material from Alice to Bob over the TN-overlay network. Here, sending x bits from party i through party j to party k signifies that TN j publishes the XOR of the first x bits of $\mathcal{SK}_{i,j}$ and $\mathcal{SK}_{j,k}$, allowing party k to recover the value $\mathcal{SK}_{i,j}$. At the end of this, we will have succeeded in pushing the maximal amount of key material to Alice and Bob, maximizing $|\mathcal{SK}_{0,n+1}|$. The above process of transmitting the XOR of two secret keys in the TN-overlay network is equivalent to encrypting one of the keys with one-time pad and thus is perfectly secure [33], [34].

Our main performance metric is the final *secret key generation rate (key rate)* between Alice and Bob, namely the ratio $|\mathcal{SK}_{0,n+1}|/N$, where N is the total number of network rounds simulated. We note that, in the above, it is possible that some TNs have leftover secret key material that cannot be pushed to Alice and Bob, causing waste. Our main contributions in this work are developing *dynamic* routing strategies to balance network operation to identify and avoid such inefficiencies, and developing post-processing techniques complementary to such dynamic strategies to further improve key rate.

III. Dynamic Routing Strategies

In this section, we first motivate the need for dynamic routing strategies, and then describe our proposed strategies.

A. The Need for Dynamic Routing Strategies

We use an algorithm in [21] as an example to illustrate the need for dynamic routing strategies. This algorithm assumes that link-level entanglement status (i.e., success or not for each link) at the end of Stage 1 is communicated using classical channels, is known to all $\{T_0, \ldots, T_{n+1}\}$ in the network, and is then used for entanglement routing in Stage 2. Specifically, the algorithm selects the shortest path (in number of hops) between any pair of nodes in $\{T_0, \ldots, T_{n+1}\}$; recall that T_0 and T_{n+1} represent Alice and Bob, and $T_1, \ldots, T_n$ are TNs. When two paths of equal length are found, one of them is selected randomly. Once a path between T_i and T_j is found, then all the links along the path are removed (since a qubit can only be used once), and the procedure repeats for the remaining links until no path can be found.

The above routing algorithm favors shorter paths that tend to have lower noise and higher success probability, and hence are more likely to succeed in establishing keys. While it has been shown to be effective when TNs are placed equally close to Alice and Bob [21], it can lead to waste in key material due to its static strategy of always favoring shorter paths.

One example is shown in Fig. 2, where Alice and Bob are along the diagonal of a grid network. It shows two settings: (i) 1TN-IDEAL, where a single TN, T_1, is placed in the center of the grid, equal distance to Alice and Bob, and (ii) OFF-CENTER, where a single TN, T_1', is placed along the diagonal, but closer to Alice than Bob. Assuming each quantum repeater has BSM success probability of 0.85, each quantum link has decoherence rate of 0.02 and success probability of 0.96, the key rate achieved using the above routing algorithm in the

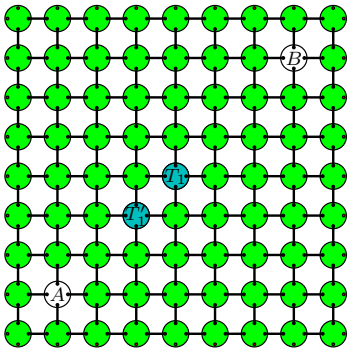


Fig. 2. An example that illustrate the benefits of dynamic routing strategies. It shows two single-TN settings: 1TN-IDEAL with the TN at T_1, and OFF-CENTER with the TN at T_1'.

OFF-CENTER setting is 0.037, only 24% of that in the 1TN-IDEAL setting (i.e., 0.153). This is because in the OFF-CENTER setting, the shorter distance between Alice and T_1' causes key establishment between them to be placed at a higher priority, leading to significantly more secret key bits between Alice and T_1' than those between T_1' and Bob. The extra key bits, however, end up being wasted since the final key rate between Alice and Bob requires keys between Alice and T_1', as well as between T_1' and Bob.

To resolve the above wastage, a dynamic routing strategy should identify T_i and T_j pairs that need to be prioritized in routing to properly balance the key pools in the network to maximize the key rate between Alice and Bob. Specifically, in Fig. 2, since the amount of secret keys between T_1' and Bob is significantly less than that between Alice and T_1', a dynamic routing strategy will prioritize QKD between T_1' and Bob. As we shall see in Section V-A, using the dynamic routing strategies that we develop, the key rate in the OFF-CENTER setting is increased to 0.066, 78% higher than that under the static strategy.

B. Dynamic Routing Strategies

Our dynamic routing strategies consider the TN-overlay network of a quantum network (see Fig. 1). Let $\mathcal{G}$ denote the TN-overlay network. That is, $\mathcal{G}$ has a set of nodes $\{T_0, \ldots, T_{n+1}\}$ and a set of edges $\{(T_i, T_j)\}$, where T_0 is Alice (A) and T_{n+1} is Bob (B). Let $c(T_i, T_j)$ be the current *capacity* of edge (T_i, T_j), which represents the amount of secret key bits that have been created between nodes T_i and T_j. Note that this quantity can be easily estimated by tracking which paths were used to generate the raw key bits between T_i and T_j and calculating the key rates based on the noise rates of these paths (see Section II), even if the raw key bits shared between T_i and T_j have not been actually converted to secret key bits.

We develop two dynamic strategies: one is a basic algorithm that applies to homogeneous topologies, and the other is an extended algorithm that applies to general topologies. Both algorithms run in each network round, using $c(T_i, T_j)$, the current amount of secret key bits between each T_i and T_j, in the TN-overlay network $\mathcal{G}$. They have low overhead since $\mathcal{G}$ is small (we consider a small number of TNs in this paper) and only requires classical communication to obtain $c(T_i, T_j)$. Specifically, each algorithm contains two steps: (i) locate an edge in $\mathcal{G}$ that has *surplus* key material, i.e., the maximum flow between Alice and Bob in the graph is lower than the capacity of this edge (and hence some key material along this edge will not be used for establishing keys between Alice and Bob), and then (ii) determine how best to spread network resources away from this edge in future rounds. These two algorithms only take effect when the secret key bits in $\mathcal{G}$ are unbalanced, as described below.

1) Basic Algorithm: In each network round, for step (i), we identify the edge that has the highest capacity in $\mathcal{G}$. Let $e_{\max} = (T_i, T_j)$ denote this edge and $c_{\max}$ denote its capacity. For step (ii), we identify edges in $\mathcal{G}$ that are *under-full* and prioritize them. Specifically, we associate with each edge $(u, v) \in \mathcal{G}$ a weight $d_{u,v}$, defined as $dist(u, v) - \epsilon$, where $dist(u, v)$ is the distance (number of hops in the shortest path) between u and v in the underlying quantum network and $\epsilon > 0$ is a small constant. We define $d_{u,v}$ to be related to distance since shorter paths are more likely to lead to successful entanglement distribution as well as less noisy keys. The subtraction of ϵ in the weight is to favor paths with more TNs; see below.

With these weights in $\mathcal{G}$, we then identify the shortest $\mathcal{P}_{\min} = A \to T_i - T_j \to B$ path from Alice to Bob that includes edge $e_{\max} = (T_i, T_j)$[1]. The above shortest path $\mathcal{P}_{\min}$ is composed of two shortest paths, $A \to T_i$ and $T_j \to B$. For each of them, if multiple paths have the same distance, a path with more TNs is more preferred due to subtraction of ϵ in the weight definition. After determining $\mathcal{P}_{\min}$ in the overlay graph $\mathcal{G}$, we identify the edges that are under-full in $\mathcal{P}_{\min}$ with respect to $e_{\max}$, i.e., any e_i on the path such that $(1+\sigma)c_i \le c_{\max}$, where c_i is the capacity of edge e_i and $\sigma \ge 0$ is a pre-determined constant. The choice of σ impacts how often the balancing action (i.e., spreading network resource away from $e_{\max}$) will be triggered. Setting σ too small may lead to frequent triggering and fluctuations, while setting σ too large may lead to delayed triggering and too much surplus key material. We experimented with a few choices of σ (from 0.01 to 0.20), and set $\sigma = 0.15$ in the rest of the paper.

Let $\mathcal{E}_{\text{uf}}$ denote the set of under-full edges identified in the above. Let $c_{\min}$ denote the capacity of the edge in $\mathcal{E}_{\text{uf}}$ that has the lowest capacity. We then prioritize any edges $e_i \in \mathcal{E}_{\text{uf}}$ so that $c_i \le (1 + \delta)c_{\min}$ for some tolerance value $\delta \ge 0$. We use δ to limit the number of edges to prioritize, and set it to be a small constant, 0.05, in the rest of the paper. Let $\mathcal{E}_p$ denote this set of edges to be prioritized. Since it is not beneficial to establish keys between two TNs that are far away from each other, we remove from $\mathcal{E}_p$ all the edges (T_k, T_k') with $dist(T_k, T_k') \ge \theta \cdot dist(A, B)$, and $\theta \in (0, 1)$ is a parameter. In the rest of the paper, we set $\theta = 0.75$ empirically, since it effectively excludes TN pairs that have no or little key material from being prioritized.

[1]For ease of exposition, we assume that there is a single shortest path of this form; when there are multiple shortest paths, one can be selected at random.

Once the set of edges to be prioritized, $\mathcal{E}_p$, is identified, in step (ii), we simply prioritize the edges (i.e., TN pairs) in $\mathcal{E}_p$ first. When there are no more paths for the edges in $\mathcal{E}_p$, we fall back to the standard operation, attempting all remaining node pairs ordered by distance. If there are multiple edges in $\mathcal{E}_p$, they will be prioritized in a random order.

In the special case where there is a single TN T_1 between Alice and Bob, the above algorithm basically reduces to the following. It first identifies one edge, (A, T_1) or (T_1, B), as $e_{\max}$, and the other edge as $e_{\min}$. If the capacities of these two edges differ significantly, specifically, $(1+\sigma)c_{\min} \leq c_{\max}$, it then prioritizes $e_{\min}$ until in one round the capacity of $e_{\min}$ is close to that of $e_{\max}$, specifically, when $(1+\sigma)c_{\min} > c_{\max}$.

Illustration. We next illustrate the above algorithm using an example with two TNs. Fig. 3a shows the topology. The two TNs, T_1 and T_2, are along the diagonal of the grid; T_1 is closer to A and T_2 is closer to B. Suppose that in one round, entanglements are established along all the links except for the two links incident to T_1 marked in red; see Fig. 3a. Fig. 3b illustrates the routing decisions with the static routing strategy. We see that the link-level entanglements are used for A and T_1, and after that, no paths can be used for T_1 and T_2. In contrast, with the dynamic routing strategy in Fig. 3c, edge (T_1, T_2) is identified as under-full and is prioritized over edge (A, T_1). Two paths, each of length 8, are used for (T_1, T_2).

In this example, the dynamic routing strategy leads to significantly higher key rate than the static strategy. Specifically, with the static strategy, the two edges (A, T_1) and (T_2, B) have secret key rates of 0.651 and 0.332, respectively, while the edge (T_1, T_2) is a bottleneck and limits the overall key rate to 0.081. When using our dynamic routing strategy, we sacrifice some key rate on both (A, T_1) and (T_2, B) (dropping them to 0.408 and 0.219, respectively), while increasing the secret key rate of the bottleneck edge (T_1, T_2) to 0.157. As a result, the overall key rate becomes 0.157, approximately twice as that with the static routing strategy.

2) Extended algorithm: In the basic algorithm described above, the weight $d(u, v)$ for edge (u, v) in the TN-overlay network $\mathcal{G}$ depends on the path length (i.e., the number of hops between u and v in the underlying quantum network). This is sufficient for a homogeneous topology, e.g., a grid where the number of edge-disjoint shortest paths between each pair of nodes is the same (i.e., 2). In less regular graphs, however, only considering path length may not be sufficient, since some node pairs may have many more paths than other pairs. We next extend the basic algorithm to general topologies. The main changes are in how we define $d(u, v)$ and identify the edges to prioritize in $A \to T_i$ and $T_j \to B$.

In the extended algorithm, we define the weight $d(u, v)$ using a heuristic on the estimated secret key rate between two TNs, u and v. Consider the underlying quantum network, where we ignore all the TNs (i.e., these nodes and the edges incident to them), except for u and v. Let $\ell_{\min}$ be the length of the shortest path between u and v. We then consider paths length from $\ell_{\min}$ to $\ell_{\min} + k$ to estimate the secret key rate between u and v, where $k \geq 0$ is a predetermined constant. The choice of k represents a trade-off between complexity and accuracy: a larger k leads to more paths that are considered and higher accuracy, at the cost of higher complexity. We use $k = 3$ in our simulations in Section VI.

Let $n_{\min}$ be the number of edge-disjoint paths of length $\ell_{\min}$ between u and v. Then following the asymptotic key rate formula for each path (see Section II), these $n_{\min}$ paths can lead to a total secret key rate of

$$d_{u,v}(\ell_{\min}) = n_{\min}\left(1 - 2h\left(\frac{1-(1-D)^{\ell_{\min}}}{2}\right)\right), \qquad (3)$$

where D is the decoherence rate, $h(\cdot)$ is the binary entropy function, and $(1-(1-D)^{\ell_{\min}})/2$ is the noise rate on a path of length $\ell_{\min}$, following the depolarizing channel in Eq. (2).

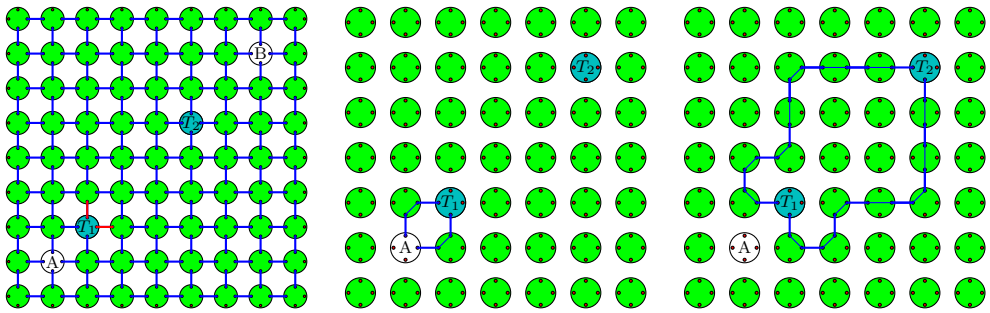
We then remove all the edges in all the paths identified as above, and repeat the above process to identify edge-disjoint paths of length $(\ell_{\min}+1)$, and estimate the key rate from these paths similarly as in Eq. (3). The above process continues until we identify and estimate a bound on the key rate from edge-disjoint paths of length $(\ell_{\min}+k)$. Then the weight of $d_{u,v}$ is set to $\sum_{\ell=\ell_{\min}}^{\ell_{\min}+k} d_{u,v}(\ell)$, i.e., the sum of the estimated key rates corresponding to paths of lengths $\ell_{\min}$ to $\ell_{\min}+k$ obtained in the above.

With the above weight definition, we now describe how to identify the edges to prioritize in $A \to T_i$ and $T_j \to B$. For ease of exposition, we only describe this for $A \to T_i$; the same approach applies to $T_j \to B$. Consider all the paths in $\mathcal{G}$, i.e., the TN-overlay network. We select the best path to be the *bottleneck shortest path* [35] with respect to this new weight definition. That is, we identify a path between A and T_i such that the minimum edge weight along the path is maximal with respect to the set of possible paths. This path may contain one or multiple edges in $\mathcal{G}$. We then use the same procedure described in the basic algorithm to identify $\mathcal{E}_p$, i.e., the set of edges to prioritize. In the above, finding the bottleneck shortest path can be accomplished in linear time [35].

The above extended algorithm reduces to the basic algorithm for homogeneous topologies and $k = 0$ (i.e., based on the number of hops of the shortest path). For general topologies, as we shall see in Section VI, it provides better solutions since the weight for each edge defined above represents the key generation rate more accurately than path length.

IV. Classical Post-Processing Methods

The dynamic routing strategies described earlier, while reducing the wastage in the network, can also lead to long paths and paths with more diverse lengths, which can be detrimental to the final secret key distillation. We next consider two classical post-processing methods to address the above issues. The first method, *key-pool segmenting*, is a new method to address the heterogeneous noise rates that can be caused by diverse path lengths. The second method, classical advantage distillation (CAD) [36], is a well-established concept often applied to QKD protocols, though we are not aware of any research evaluating its performance in multi-path quantum



(a) Entanglement status. (b) Static routing strategy. (c) Dynamic routing strategy.

Fig. 3. An example illustrating the different routing decisions by the static and dynamic routing strategies. In (b) and (c), only partial topologies including A, T_1 and T_2 are shown.

networks. The above two post-processing techniques can be beneficial alone, and, as we shall show, combining them can lead to even larger benefits.

A. Key-pool Segmenting

In standard QKD protocols, one usually assumes that the quantum channel used to transmit quantum states between two parties has a certain fidelity, and each raw key bit shared between these two parties has the same error probability. In our network, multiple paths can be used to share entanglement between T_i and T_j, which may not be of the same length. Therefore, even if each individual link in the network has the same fidelity, these paths may result in entanglements with different fidelities. Specifically, in one round, the shortest path may be used between T_i and T_j, resulting in a key bit with the lowest possible noise; in another round, maybe only a longer, noisier path can be used, resulting in a noisier raw key bit. The key-pool between T_i and T_j therefore is a mixture of key bits with different noise rates, with the expected noise of the entire key-pool being a weighted average thereof.

This mixture of key-pools can degrade the overall quality of the key-pool to the point that the total key rate is reduced (see example below). To mitigate this impact, we propose a segmenting approach to post-processing: rather than process the entire raw key together, we allow the network to segment the raw key-pool based on the expected noise rate of each bit, partitioning it into sub-pools with similar expected noise rates. When two parties wish to transform their raw keys into secret-keys, they process each sub-pool separately, before ultimately combining it into a single key-pool. More formally, consider the scenario where the raw key-pool is divided into b segments, each segment $i = 1, 2, \cdots, b$ having a noise level of Q_i and accounts for a fraction $p_i \in (0, 1]$ of the original key pool. Then, asymptotically, we can treat each segment as an individual key-pool and process it separately, leading to an ultimate final key rate of $1 - 2\sum_{i=1}^{b} p_i h(Q_i)$.

The above method can result in substantial improvement in key rate in certain networks. Furthermore, it is practical and easy to implement, as the repeaters can periodically disclose (through classical channels) their previous routing decisions to allow parties to discover which paths were used. The lengths of these paths, assuming relatively static noise profiles on the links, can then be used to find the expected noise rate of each bit (if noise levels change over time, one can use channel tomography to estimate noise levels periodically).

We next present an example, and then show that, in the asymptotic regime, this segmenting approach results in strictly better or equal key rates than no segmenting. Consider a raw key composed of a mixture of two sub-pools, K_1 and K_2, where they have an expected noise rate of $Q_1 = 0.04$ and $Q_2 = 0.10$, and account for $p_1 = 75\%$ and $p_2 = 25\%$ of the total key-pool, respectively. The total average noise rate is $Q_{avg} = 0.055$, which leads to a key rate of $1 - 2h(Q_{avg}) = 0.385$. The key rate with segmenting is simply the weighted average of the key rate of each sub-pool, resulting in a key rate of $p_1[1-2h(Q_1)]+p_2[1-2h(Q_2)] = 0.402$, 4.4% higher than the key rate without segmenting.

In general, the key-pool segmenting approach will never lead to a lower key rate, and can only improve the asymptotic key rate when there are at least two key pools with different noise levels. This is easily seen due to the concavity of Shannon entropy on the interval $[0, 1/2]$. In the finite-key regime, there may be sampling considerations that do not allow segmenting to guarantee equal or better performance. We leave the finite-key regime as future work.

B. Classical Advantage Distillation (CAD)

After key-pool segmenting as described above, some key pools may have high noise. For such pools, CAD [22] can be used to extract additional secret key material from noisy raw key-pools. While CAD is a well-established tool, we discuss it here for two reasons: (i) apply it to our networks and quantify its resulting benefits, and (ii) demonstrate that there is an interesting interplay between CAD and the key-pool segmenting technique that we propose: in some network configurations, combining them can lead to positive key rates, which cannot be achieved using one technique alone (see Section V).

CAD is a post-processing method undertaken by two parties once their raw key pool has been generated. For ease of exposition, we describe CAD assuming the two parties are Alice and Bob; in our setting, they can be any two parties, T_i and T_j. We use specifically the CAD process in [36],

[37]. Prior to error correction, the two parties engage in the following procedure for a pre-determined CAD level $C \geq 1$:

1) Alice selects C raw key bits that all have the same value, sending their corresponding indices to Bob.
2) Bob notifies Alice whether or not, in his raw key, those C bits take the same value.
3) If these C bits take the same value in Bob's raw key, then Alice and Bob shrink the block of C bits into a single bit, and add the bit to a new raw key string; otherwise no new bit is added to their new raw key. Either way, they discard these C bits.
4) Alice and Bob repeat this process until the entire original key has been depleted.

When running CAD with blocks of size C, it is clear that the raw key will shrink to, at a minimum, $1/C$ of the original length. If the key is noisy, it may shrink even more. At low noise levels, CAD does not improve key rate (or equivalently, $C = 1$ is optimal), but as the noise level increases, the optimal C also increases. As an example, for $C = 2$, the noise tolerance of the E91 protocol increases from the usual 11% to 18.2% [36], [37]. For a given noise rate and CAD level C, one can compute the final key rate in a straightforward manner using results in [36]. CAD can be used with and without the key-pool segmenting technique. When the key-pool segmenting technique is used, we use the noise estimates for each individual sub-pool to select the optimal CAD level.

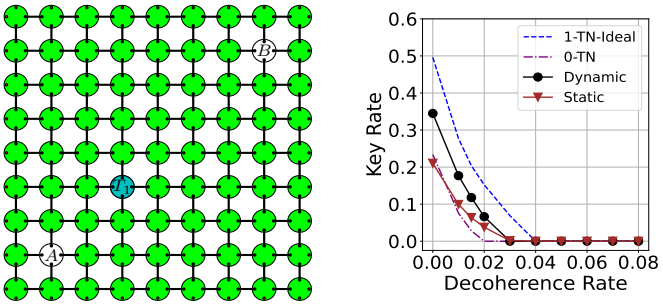
V. Evaluation Results for Grid Topologies

In this section, we use simulations to evaluate the performance of the dynamic routing strategies, as well as the impact of key-pool segmenting and CAD on the resultant key rate. We focus on a simple grid topology; evaluation using random graphs is deferred to Section VI. Our goal is illustrating the benefits of the dynamic routing strategies and the post-processing techniques, and also when they may be less helpful. Specifically, we consider a square grid of size 7×7, with Alice and Bob located at the corners, embedded in an another square 9×9 grid. The link success probability P and BSM success probability R are set to 0.96 and 0.85, respectively; they will be varied in Section VI for random graphs. The decoherence rate varies from 0 to 0.08, modeling a range of technologies [23].

The results for dynamic routing strategies below are obtained using the basic algorithm; using the extended algorithm leads to the same results. While the dynamic routing strategies can be used with any underlying routing algorithms, for simplicity, we use it with the distance-based routing algorithm in Section III-A; using it with other routing algorithms is left as future work. Each result was obtained over 10^7 network rounds, using a custom-built simulator written in Python.

A. Results of Dynamic Routing Strategy

We consider several scenarios with asymmetric TN placement. For each scenario, we present the performance of our dynamic routing strategy as decoherence rate increases, and compare the key rates with those obtained using the static strategy. For reference, we also show the key rate when using one or two TNs under ideal placement, referred to as 1TN-IDEAL and 2TN-IDEAL, where one or two TNs are evenly placed along the diagonal of the grid. In addition, we also show the results with no TN. In all the results below, the key rates with no TN are significantly lower than those with TNs, demonstrating the benefits of using TNs, even at asymmetric locations.



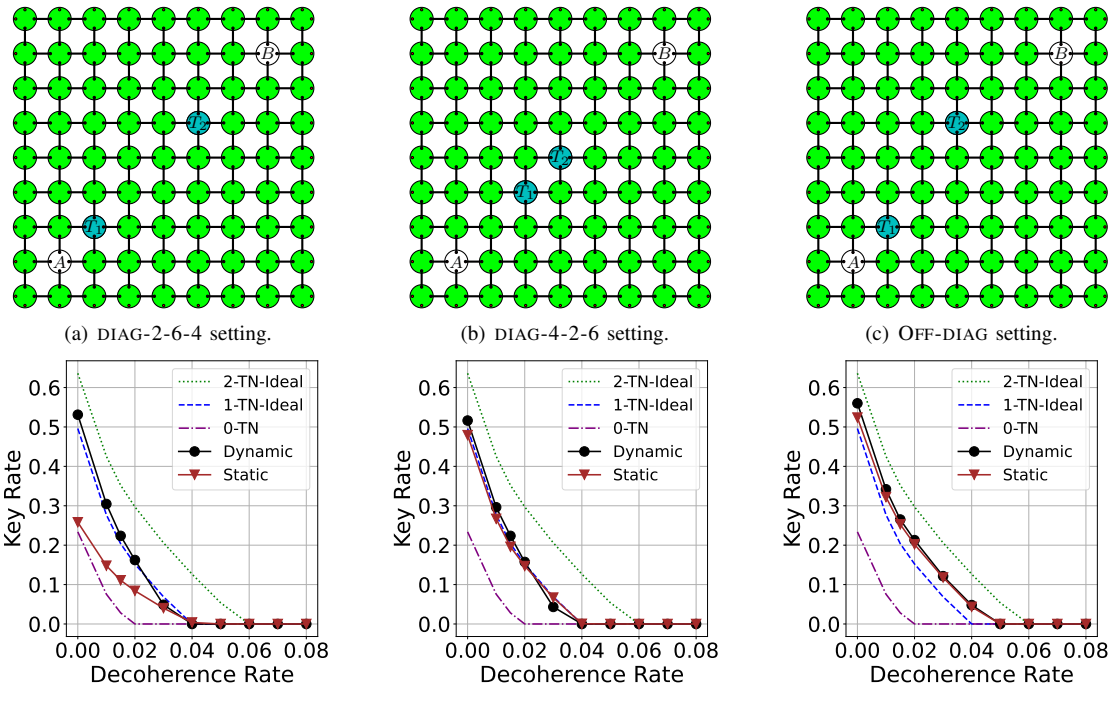
(a) OFF-CENTER setting. (b) Key rate.

Fig. 4. Grid topology with one TN placed off the center: comparison of dynamic and static strategies.

Single TN. Fig. 4a shows an OFF-CENTER setting where a single TN is placed along the diagonal, closer to Alice than Bob. Fig. 4b shows the key rate under static and dynamic strategies for this setting. For reference, it also plots the results under 1TN-IDEAL setting. We see that the dynamic strategy leads to significantly higher key rate than the static strategy. For instance, when there is no decoherence, the dynamic strategy leads to a key rate of 0.345, 64% higher than the value (0.210) under the static strategy. When the decoherence rate is 0.02, the improvement is 78% (0.066 vs. 0.037). As expected, even under the dynamic strategy, the key rate in OFF-CENTER setting is lower than that under 1TN-IDEAL, due to the longer distance between the TN and Bob.

Two TNs. We consider three settings: Fig. 5a and b show two settings where the two TNs are placed along the diagonal; Fig. 5c shows one setting where one TN is on the diagonal, while the other is off the diagonal. We refer to the first setting as DIAG-2-6-4, where the three numbers represent the Manhattan distances between Alice and T_1, between T_1 and T_2, and between T_2 and Bob, respectively. Similarly, we refer to the second setting as DIAG-4-2-6. In the third setting, T_1 is on the diagonal, with Manhattan distance 2 from Alice, and T_2 is off the diagonal, equidistant from both Bob and T_1 (both with Manhattan distance 5), and we refer to it as OFF-DIAG setting. Fig. 5d-f show the results of the above three settings. The results under 1TN-IDEAL and 2TN-IDEAL are also shown in these figures.

The DIAG-2-6-4 setting differs from 2TN-IDEAL in that TN T_1 is closer to Alice, causing the distance between the two TNs, T_1 and T_2, to be larger. In this setting, we observe significant benefits of using dynamic over static routing strategy. With no decoherence, the key rate of the dynamic routing strategy is 0.531, more than $2\times$ of that under the static



(a) DIAG-2-6-4 setting. (b) DIAG-4-2-6 setting. (c) OFF-DIAG setting.

(d) Results for DIAG-2-6-4 setting. (e) Results for DIAG-4-2-6 setting. (f) Results for OFF-DIAG setting.

Fig. 5. Grid topology with two TNs at asymmetric locations: comparison of dynamic and static strategies.

routing strategy (0.258), and also slightly higher than 1TN-IDEAL (which has key rate 0.496). In general, the dynamic routing strategy achieves slightly higher or similar key rate as the 1TN-IDEAL setting for all the decoherence rates.

The DIAG-4-2-6 setting differs from 2TN-IDEAL in that T_2 is closer to T_1, causing the distance between T_2 and Bob to be larger. In this case, since the two TNs are so close to each other, they work similarly as a single TN. As a result, we observe much less benefit under the dynamic strategy over the static strategy (e.g., leading to 7.7% improvement in key rate when the decoherence rate is 0.0). The results are also very close to those under the 1TN-IDEAL setting.

Last, in the OFF-DIAG setting, we see that the dynamic strategy outperforms the static strategy when the decoherence rate is low. As an example, when the decoherence rate is 0.0, the key rate under dynamic routing is 0.56, 7.7% higher than that under static routing (0.52), and only 17% lower than the 2TN-IDEAL setting. Interestingly, in this setting, with the two TNs, both dynamic and static routing strategies achieve visibly higher key rate than the 1TN-IDEAL setting.

B. Results of Classical Post-Processing

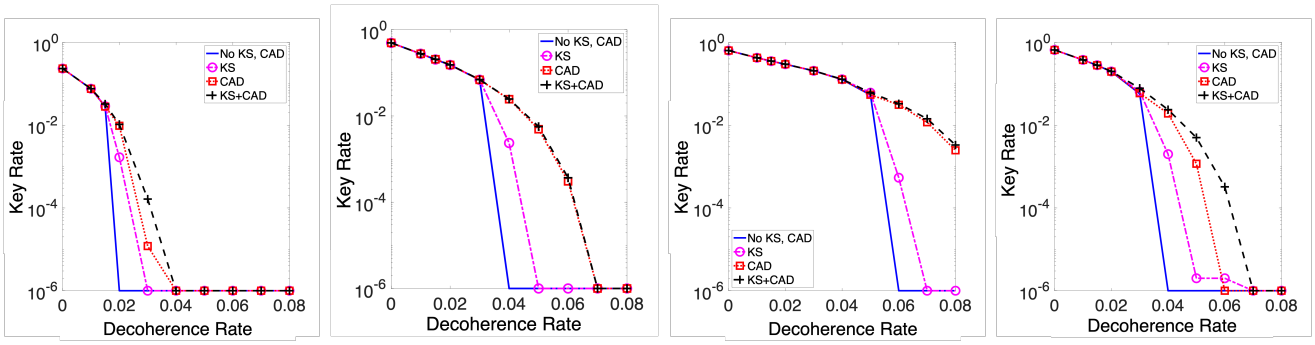
We now evaluate the impact of the two classical post-processing techniques, CAD and our key-pool segmenting, on key rate. For ease of exposition, we consider four settings, NO-TN, 1TN-IDEAL and 2TN-IDEAL as described earlier, and 2TN-CORNER (i.e., two TNs at the two corners of the grid). For all these settings, it is sufficient to use the static routing strategy, since they either have no TN or the TN(s) are placed at symmetric location(s). In Section V-C, we use the dynamic routing strategy with these two post-processing techniques for scenarios with asymmetric TN placement. Since both key-pool segmenting and CAD tend to bring benefits at high noise rates (and therefore at low key rates), we plot the graphs below on a log-linear scale to highlight the benefits of these two techniques.

The NO-TN setting (Fig. 6a) serves as the baseline. In this setting, we see that CAD and key-pool segmenting each improves key rate when the decoherence rate is larger than 0.02. In addition, combining these two techniques can improve the key rate significantly than using them in isolation, e.g., leading to an order of magnitude improvement in key rate when the decoherence rate is 0.03.

In 1TN-IDEAL and 2TN-IDEAL settings (Fig. 6b and c), while key-pool segmenting does improve key rate significantly when the decoherence rate $D \geq 0.03$, using key-pool egmenting and CAD together only brings slightly higher key rate beyond what is already provided by CAD. The observation is different in the 2TN-CORNER setting (Fig. 6d), where we observe significantly higher benefits from combining key-pool segmenting and CAD over using each technique separately. The significantly higher benefits from key-pool segmenting beyond CAD in this setting might be because the TNs are at the two corners, and hence can lead to more diverse path lengths, resulting in more visible benefits from key-pool segmenting, which differentiates paths of diverse qualities.

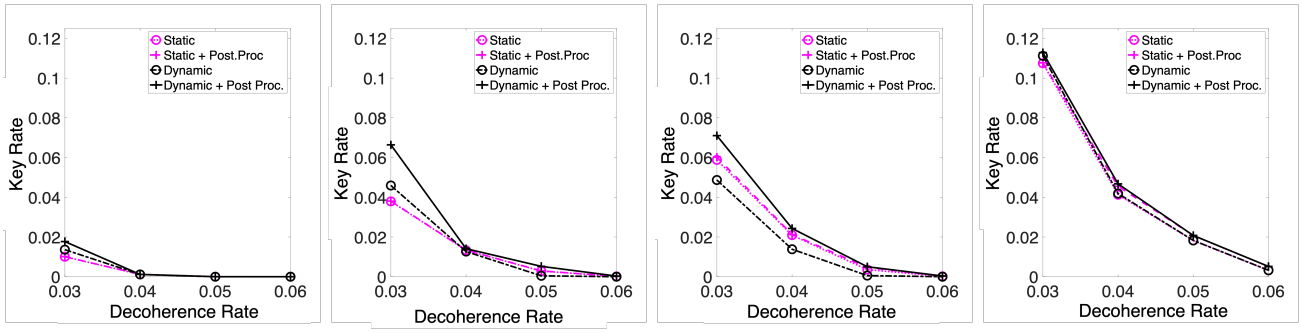
C. Results of Dynamic Routing Strategy with Post-Processing

We now evaluate the benefits of combining the dynamic routing strategy and the two post-processing methods for the settings with asymmetric TN placement. Specifically, we



(a) NO-TN. (b) 1TN-IDEAL. (c) 2TN-IDEAL. (d) 2TN-CORNER.

Fig. 6. Grid topology: performance of CAD and key-pool segmenting (KS) techniques with static routing strategy (which suffices for these settings).



(a) OFF-CENTER. (b) DIAG-2-6-4. (c) DIAG-4-2-6. (d) OFF-DIAG.

Fig. 7. Grid topology: key rate when combining dynamic routing strategy and post-processing techniques.

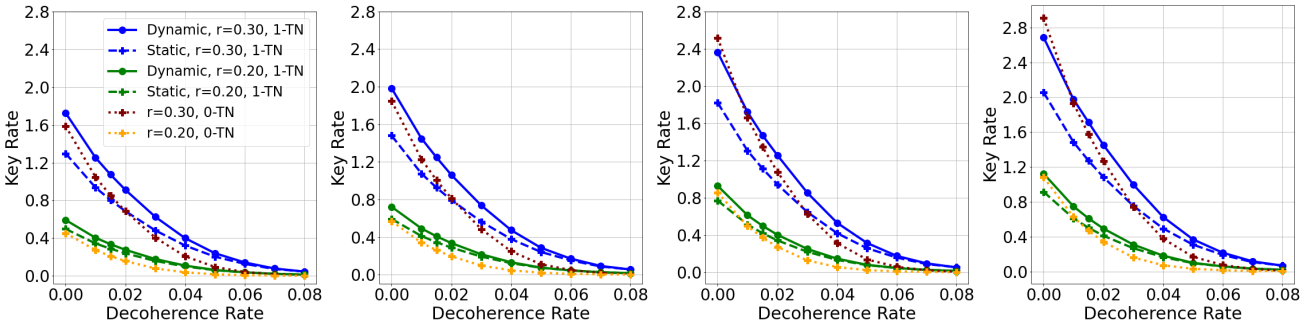
consider four asymmetric settings that we considered earlier: one with a single TN, i.e., OFF-CENTER, the other three with two TNs, i.e., DIAG-2-6-4, DIAG-4-2-6, and OFF-DIAG. Since post-processing techniques only bring benefits under high noise rates, Fig. 7 only plots the key rate when the decoherence rate $D \geq 0.03$. We see that for all the settings, adding post-processing to the raw key pools obtained from the dynamic routing strategy improves the key rate (comparing the solid and dashed black lines), and the benefits are particularly clear under DIAG-2-6-4 and DIAG-4-2-6. The above benefits are expected since the dynamic routing strategy can lead to much longer paths, and higher noise rate, which can benefit from post-processing techniques. For comparison, Fig. 7 also plots the key rate under the static strategy, with and without post-processing. As expected, the benefits of using post-processing techniques become much less noticeable, since shorter paths are chosen with higher priority at all times under the static strategy.

VI. EVALUATION RESULTS FOR RANDOM GRAPHS

We now consider random geometric graphs where nodes are randomly placed in an area and two nodes are connected by a link if their distance is less than a radius r. Specifically, we consider a unit square area with 50 nodes. We vary r to 0.2, 0.25 and 0.3 (we do not set r less than 0.2 since such values often lead the graph disconnected). For each r, we create 10 random graphs. With $r = 0.2$, 0.25 and 0.3, the average node degree is 5.0, 7.7, and 10.2, and the standard deviation of node degree is 2.4, 3.2, and 3.7, respectively. In each graph, Alice and Bob are placed close to the two opposite corners of the graph. We set BSM success probability R to be 0.85 or 0.95, and link success probability P to 0.85 and 0.96, leading to four combinations of R and P values. As earlier, we assume that one or two TNs are used for QKD between Alice and Bob. The TNs are placed randomly along a stripe between Alice and Bob, which naturally lead to asymmetric TN placement between Alice and Bob due to the random topology.

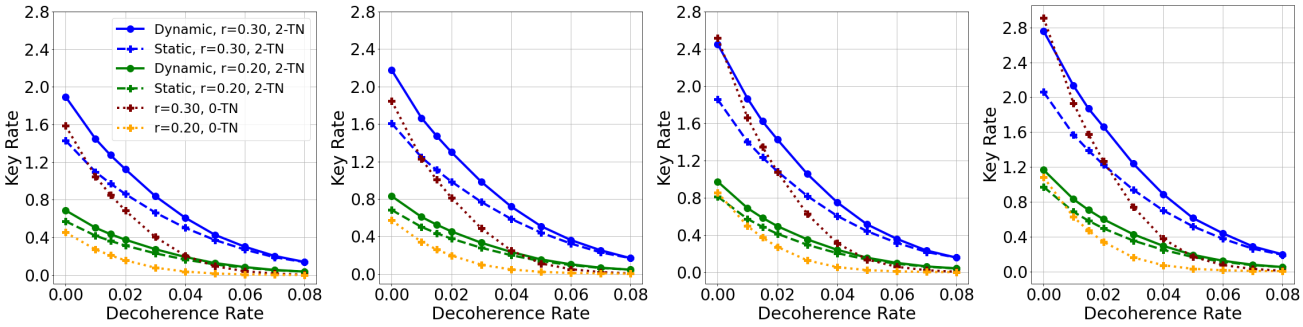
In the following, we compare the results when using static versus dynamic routing strategy with various settings of BSM success probability R, link success probability P, graph radius r, and the number of TNs. For each setting, the decoherence rate is again varied from 0 to 0.08. The two classical post-processing techniques (key-pool segmenting and CAD) are used in all the settings. The dynamic routing strategy uses the extended algorithm in Section III-B2.

Single TN. Fig. 8a-d plot the results when using a single TN under the four combinations of R and P when decoherence rate varies from 0 to 0.08. In each figure, we compare the results of dynamic and static routing strategies when the radius $r = 0.2$ or 0.3; the results when $r = 0.25$ are in between and omitted for clarity. For comparison, the results with no TN are also plotted in the figure. Each point in the graph is the average value of 100 settings, i.e., 10 random graphs and 10 random placements of a single TN in each graph. As expected, for a given strategy and R and P combination, the key rate is



(a) $R = 0.85$, $P = 0.85$. (b) $R = 0.85$, $P = 0.96$. (c) $R = 0.95$, $P = 0.85$. (d) $R = 0.95$, $P = 0.96$.

Fig. 8. Random graphs with a single TN: comparison of dynamic and static routing strategies. The results with no TN are also plotted for comparison.



(a) $R = 0.85$, $P = 0.85$. (b) $R = 0.85$, $P = 0.96$. (c) $R = 0.95$, $P = 0.85$. (d) $R = 0.95$, $P = 0.96$.

Fig. 9. Random graphs with two TNs: comparison of dynamic and static routing strategies. The results with no TN are also plotted for comparison.

higher for higher r since the graph is denser, and hence more paths between T_i and T_j.

We see that the dynamic strategy outperforms the static strategy in all the settings. The amount of improvement is more visible when the decoherence rate is low. For $r = 0.2$, 0.25 and 0.3, the improvement from the dynamic strategy over the static strategy for the various R and P combinations is up to 23%, 18%, and 23%, respectively. We also see that the key rates with TN and dynamic strategy are significantly higher than those with no TN in all but two settings (where decoherence rate $D = 0$ and BSM success probability $R = 0.95$), where there is no need to use TN since the quantum network itself is of high quality.

Two TNs. Fig. 9a-d plot the results when two TNs are used under the four combinations of R and P values. Again, in each figure, we compare the results of dynamic and static routing strategies when the radius $r = 0.2$ or 0.3, and the results when $r = 0.25$ are in between (omitted). Compared to the results when using one TN (Fig. 8), we see that using two TNs generally leads to higher key rate. Each point in the graph is the average value of 150 settings, with 10 random graphs and 15 random placements of two TNs in each graph. We again see that the dynamic strategy outperforms the static strategy in all the settings, and the amount of improvement is more significant when the decoherence rate is low. Specifically, for $r = 0.2$, 0.25 and 0.3, the improvement from the dynamic strategy over the static strategy is up to 22%, 23%, and 23%, respectively. Again, the key rates with TN and dynamic strategy are significantly higher than those with no TN in all but two settings.

VII. Related Work

Early quantum networks contain end users and optical switches that allow users to route qubits to each other. Such networks have been used for QKD applications [2], [38]. In later stage TN-based quantum networks, much work [2]–[13] has been devoted to studying routing strategies and general design strategies for QKD applications. For a general survey, the reader is referred to [39], [40]. Much current research is devoted to the true quantum internet with only quantum repeaters and no TNs; in particular to its behavior and performance in terms of maximizing entanglement distribution rates, with [24], [25], [41]–[44] particularly focusing on routing within a quantum internet. Our study differs from all the above studies in that we consider hybrid quantum networks with both quantum repeaters and TNs. Our round-based network model follows existing studies for repeater-based quantum networks (e.g., [24], [25]), while we consider both loss and noise, instead of only loss.

Our work is inspired by [21], which proposed two routing algorithms for QKD in hybrid quantum networks. Their routing algorithms are suitable for scenarios with symmetric placement of TNs, but not for those with asymmetric placements of TNs. Our proposed dynamic routing strategies can run on top of their routing algorithms (as well as other routing algorithms) to balance the key pools in the network to improve

the overall key generation rate. In addition, we investigate new and existing post-processing techniques that can be combined with the dynamic routing strategies to further improve the key rate. Last, our key-pool segmenting approach is more general than those in existing studies [45]–[47], which simply discard measurements with high quantum error rates.

VIII. Conclusion and Future Work

In this paper, we have considered QKD in hybrid quantum networks, consisting of a majority of quantum repeaters and a small number of physically secure TNs. We developed dynamic routing strategies that make routing decisions based on the current key-pool conditions in the network and a key-pool segmenting post-processing technique. We then evaluated the impacts of these two techniques and CAD using simulations. Our results show that the dynamic routing strategies significantly improve the key rate between two users when the TNs are placed at asymmetric locations, while the post-processing techniques lead to substantial improvement in high noise scenarios. Furthermore, combining them can further improve the key rate.

The random graphs that we explored contain 50 nodes with one or two TNs. As future work, we will explore larger network topologies with more TNs. Our key rate analysis assumes asymptotic scenarios. We will explore finite-key scenarios as future work. In this work, we have used the dynamic routing strategy and post-processing techniques with one routing algorithm. As future work, we will explore using them with other routing algorithms.

Disclaimer

This paper was prepared for information purposes by the teams of researchers from the various institutions identified above, including the Global Technology Applied Research group of JPMorgan Chase Bank, N.A.. This paper is not a product of the Research Department of JPMorgan Chase & Co. or its affiliates. Neither JPMorgan Chase & Co. nor any of its affiliates make any explicit or implied representation or warranty and none of them accept any liability in connection with this paper, including, but limited to, the completeness, accuracy, reliability of information contained herein and the potential legal, compliance, tax or accounting effects thereof. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction.

References

[1] S. Pirandola, U. L. Andersen, L. Banchi, M. Berta, D. Bunandar, R. Colbeck, D. Englund, T. Gehring, C. Lupo, C. Ottaviani, *et al.*, "Advances in quantum cryptography," *Advances in optics and photonics*, vol. 12, no. 4, pp. 1012–1236, 2020.

[2] P. Toliver, R. J. Runser, T. E. Chapuran, J. L. Jackel, T. C. Banwell, M. S. Goodman, R. J. Hughes, C. G. Peterson, D. Derkacs, J. E. Nordholt, *et al.*, "Experimental investigation of quantum key distribution through transparent optical switch elements," *IEEE Photonics Technology Letters*, vol. 15, no. 11, pp. 1669–1671, 2003.

[3] C. Le Quoc, P. Bellot, and A. Demaille, "Stochastic routing in large grid-shaped quantum networks," in *2007 IEEE International Conference on Research, Innovation and Vision for the Future*, pp. 166–174, IEEE, 2007.

[4] H. Wen, Z. Han, Y. Zhao, G. Guo, and P. Hong, "Multiple stochastic paths scheme on partially-trusted relay quantum key distribution network," *Science in China Series F: Information Sciences*, vol. 52, no. 1, pp. 18–22, 2009.

[5] Y. Tanizawa, R. Takahashi, and A. R. Dixon, "A routing method designed for a quantum key distribution network," in *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 208–214, IEEE, 2016.

[6] C. Yang, H. Zhang, and J. Su, "The qkd network: model and routing scheme," *Journal of Modern Optics*, vol. 64, no. 21, pp. 2350–2362, 2017.

[7] Q. Han, L. Yu, W. Zheng, N. Cheng, and X. Niu, "A novel qkd network routing algorithm based on optical-path-switching.," *J. Inf. Hiding Multim. Signal Process.*, vol. 5, no. 1, pp. 13–19, 2014.

[8] M. Mehic, P. Fazio, S. Rass, O. Maurhart, M. Peev, A. Poppe, J. Rozhon, M. Niemiec, and M. Voznak, "A novel approach to quality-of-service provisioning in trusted relay quantum key distribution networks," *IEEE/ACM Transactions on Networking*, 2019.

[9] C. Yang, H. Zhang, and J. Su, "Quantum key distribution network: Optimal secret-key-aware routing method for trust relaying," *China Communications*, vol. 15, no. 2, pp. 33–45, 2018.

[10] Q. Li, Y. Wang, H. Mao, J. Yao, and Q. Han, "Mathematical model and topology evaluation of quantum key distribution network," *Optics Express*, vol. 28, no. 7, pp. 9419–9434, 2020.

[11] J. Yao, Y. Wang, Q. Li, H. Mao, A. A. A. El-Latif, and N. Chen, "An efficient routing protocol for quantum key distribution networks," *Entropy*, vol. 24, no. 7, p. 911, 2022.

[12] P. K. Tysowski, X. Ling, N. Lütkenhaus, and M. Mosca, "The engineering of a scalable multi-site communications system utilizing quantum key distribution (qkd)," *Quantum Science and Technology*, vol. 3, no. 2, p. 024001, 2018.

[13] L.-Q. Chen, M.-N. Zhao, K.-L. Yu, T.-Y. Tu, Y.-L. Zhao, and Y.-C. Wang, "Ada-qkdn: a new quantum key distribution network routing scheme based on application demand adaptation," *Quantum Information Processing*, vol. 20, no. 9, pp. 1–22, 2021.

[14] H. J. Kimble, "The quantum internet," *Nature*, vol. 453, no. 7198, pp. 1023–1030, 2008.

[15] M. Caleffi, A. S. Cacciapuoti, and G. Bianchi, "Quantum internet: From communication to distributed computing!," in *Proceedings of the 5th ACM International Conference on Nanoscale Computing and Communication*, pp. 1–4, 2018.

[16] S. Wehner, D. Elkouss, and R. Hanson, "Quantum internet: A vision for the road ahead," *Science*, vol. 362, no. 6412, p. eaam9288, 2018.

[17] M. Peev, C. Pacher, R. Alléaume, C. Barreiro, J. Bouda, W. Boxleitner, T. Debuisschert, E. Diamanti, M. Dianati, J. Dynes, *et al.*, "The secoqc quantum key distribution network in vienna," *New Journal of Physics*, vol. 11, no. 7, p. 075001, 2009.

[18] T.-Y. Chen, J. Wang, H. Liang, W.-Y. Liu, Y. Liu, X. Jiang, Y. Wang, X. Wan, W.-Q. Cai, L. Ju, *et al.*, "Metropolitan all-pass and inter-city quantum communication network," *Optics express*, vol. 18, no. 26, pp. 27217–27225, 2010.

[19] Q. Zhang, F. Xu, Y.-A. Chen, C.-Z. Peng, and J.-W. Pan, "Large scale quantum key distribution: challenges and solutions," *Optics express*, vol. 26, no. 18, pp. 24260–24273, 2018.

[20] M. Sasaki, M. Fujiwara, H. Ishizuka, W. Klaus, K. Wakui, M. Takeoka, S. Miki, T. Yamashita, Z. Wang, A. Tanaka, *et al.*, "Field test of quantum key distribution in the tokyo qkd network," *Optics express*, vol. 19, no. 11, pp. 10387–10409, 2011.

[21] O. Amer, W. O. Krawec, and B. Wang, "Efficient routing for quantum key distribution networks," in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE); arXiv:2005.12404*, pp. 137–147, IEEE, 2020.

[22] U. M. Maurer, "Secret key agreement by public discussion from common information," *IEEE transactions on information theory*, vol. 39, no. 3, pp. 733–742, 1993.

[23] A. Dahlberg, M. Skrzypczyk, T. Coopmans, L. Wubben, F. Rozpedek, M. Pompili, A. Stolk, P. Pawełczak, R. Knegjens, J. de Oliveira Filho, R. Hanson, and S. Wehner, "A link layer protocol for quantum networks," in *Proc. of ACM SIGCOMM*, 2019.

[24] M. Pant, H. Krovi, D. Towsley, L. Tassiulas, L. Jiang, P. Basu, D. Englund, and S. Guha, "Routing entanglement in the quantum internet," *npj Quantum Information*, vol. 5, no. 1, pp. 1–9, 2019.

[25] S. Shi and C. Qian, "Concurrent entanglement routing for quantum networks: Model and designs," in *ACM Sigcomm*, August 2020.
[26] O. Svelto, *Principles of Lasers*. Springer US, 2010.
[27] H. Kaushal, V. K. Jain, and S. Kar, *Free Space Optical Communication*. Springer, 2017.
[28] S. Pirandola, R. Laurenza, C. Ottaviani, and L. Banchi, "Fundamental limits of repeaterless quantum communications," *Nature communications*, vol. 8, no. 1, pp. 1–15, 2017.
[29] A. K. Ekert, "Quantum cryptography based on bell's theorem," *Physical review letters*, vol. 67, no. 6, p. 661, 1991.
[30] P. W. Shor and J. Preskill, "Simple proof of security of the bb84 quantum key distribution protocol," *Phys. Rev. Lett.*, vol. 85, pp. 441–444, Jul 2000.
[31] R. Renner, N. Gisin, and B. Kraus, "Information-theoretic security proof for quantum-key-distribution protocols," *Phys. Rev. A*, vol. 72, p. 012332, Jul 2005.
[32] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. MIT Press and McGraw-Hill, 4 ed., 2022.
[33] S. M. Bellovin, "Frank miller: Inventor of the one-time pad," *Cryptologia*, vol. 35, no. 3, pp. 203–222, 2011.
[34] C. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, 1949.
[35] V. Kaibel and M. Peinhardt, "On the bottleneck shortest path problem," 2006. ZIB-Report 06-22 (May 2006).
[36] J. Bae and A. Acín, "Key distillation from quantum channels using two-way communication protocols," *Physical Review A*, vol. 75, no. 1, p. 012334, 2007.
[37] H. F. Chau, "Practical scheme to share a secret key through a quantum channel with a 27.6% bit error rate," *Physical Review A*, vol. 66, no. 6, p. 060302, 2002.
[38] C. Elliott, A. Colvin, D. Pearson, O. Pikalo, J. Schlafer, and H. Yeh, "Current status of the darpa quantum network," in *Quantum Information and computation III*, vol. 5815, pp. 138–149, International Society for Optics and Photonics, 2005.
[39] Y. Cao, Y. Zhao, Q. Wang, J. Zhang, S. X. Ng, and L. Hanzo, "The evolution of quantum key distribution networks: On the road to the qinternet," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 839–894, 2022.
[40] M. Mehic, M. Niemiec, S. Rass, J. Ma, M. Peev, A. Aguado, V. Martin, S. Schauer, A. Poppe, C. Pacher, *et al.*, "Quantum key distribution: a networking perspective," *ACM Computing Surveys (CSUR)*, vol. 53, no. 5, pp. 1–41, 2020.
[41] M. Caleffi, "Optimal routing for quantum networks," *IEEE Access*, vol. 5, pp. 22299–22312, 2017.
[42] L. Gyongyosi and S. Imre, "Entanglement-gradient routing for quantum networks," *Scientific reports*, vol. 7, no. 1, pp. 1–14, 2017.
[43] K. Chakraborty, F. Rozpedek, A. Dahlberg, and S. Wehner, "Distributed routing in a quantum internet," *arXiv preprint arXiv:1907.11630*, 2019, unpublished.
[44] Y. Zhao and C. Qiao, "Redundant entanglement provisioning and selection for throughput maximization in quantum networks," in *IEEE INFOCOM*, 2021.
[45] C. Erven, B. Heim, E. Meyer-Scott, J. P. Bourgoin, R. Laflamme, G. Weihs, and T. Jennewein, "Studying free-space transmission statistics and improving free-space quantum key distribution in the turbulent atmosphere," *New Journal of Physics*, vol. 14, 2012.
[46] G. Vallone, D. Marangon, M. Canale, I. Savorgnan, D. Bacco, M. Barbieri, S. Calimani, C. Barbieri, N. Laurenti, and P. Villoresi, "Adaptive real time selection for quantum key distribution in lossy and turbulent free-space channels," *Phys. Rev. A*, vol. 91, 2015.
[47] W. Wang, F. Xu, and H.-K. Lo, "Prefixed-threshold real-time selection method in free-space quantum key distribution," *Phys. Rev. A*, vol. 97, 2018.