Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Quantum utility routing in continuous-variable QKD networks with trusted and untrusted relays

Farsad Ahmad
Lahore University of Management Sciences
Aeysha Khalique
aeysha.khalique@lums.edu.pk
Lahore University of Management Sciences

Article

Keywords:

Posted Date: November 9th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11374976/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Quantum utility routing in continuous-variable QKD networks with trusted and untrusted relays

Farsad Ahmad[1,*] **and Aeysha Khalique**[1,+]

[1]Department of Physics, Syed Babar Ali School of Science and Engineering, Lahore University of Management Sciences, Lahore, 54792, Pakistan
[*]farsad.ahmad@lums.edu.pk
[+]aeysha.khalique@lums.edu.pk

October 15, 2026

Abstract

Continuous-variable quantum key distribution (CV-QKD) networks require routing strategies that account simultaneously for secret key generation, relay trust and finite network resources. We introduce Quantum Utility Routing (QUR), a weighted routing framework for networks combining trusted point-to-point links with untrusted CV Bell measurement relays. Transition level composable finite size secret key rates (SKRs) are incorporated directly into the route search, allowing QUR to balance bottleneck SKR against route geometry and resource use. We consider graph optimized QUR ($\mathrm{QUR}_{\mathrm{go}}$) and a fixed weight implementation ($\mathrm{QUR}_{\mathrm{fw}}$) that uses machine learning to avoid repeated graph specific weight optimization, and benchmark both against A* and Max–Min routing. At the default network configuration, $\mathrm{QUR}_{\mathrm{go}}$ and $\mathrm{QUR}_{\mathrm{fw}}$ improve the mean path SKR relative to A* while requiring substantially fewer network resources than Max–Min. The framework also maintains secure connectivity over a broad range of trusted node availability by exploiting untrusted infrastructure as measurement relays. These results demonstrate that routing in CV-QKD networks can be treated as a configurable trade off between cryptographic performance and network resource consumption.

Introduction

Quantum key distribution (QKD) enables distant users to establish secret keys whose security is derived from the laws of quantum mechanics rather than the computational difficulty of a mathematical problem [1]. Continuous-variable QKD (CV-QKD) is particularly attractive for existing communication networks because it can encode information in the quadratures of coherent or squeezed light and recover it using coherent detection [2–4]. CV-QKD utilizes standard optical fiber, well established telecommunication components, high bandwidth coherent receivers and increasingly integrated photonic hardware [5–7]. Practical implementations and switched CV-QKD network demonstrations further indicate that CV-QKD is progressing from isolated laboratory links towards a deployable communication infrastructure [8, 9]. The QKD infrastructure of a quantum network can be built from discrete variable (DV) systems, CV systems, or by combining the two. Metropolitan QKD was demonstrated for both DV [10, 11] and CV [12] technologies.

Most QKD protocols are developed as point-to-point (P2P) systems, whereas practical networks must connect distant users over distances where direct links may become inefficient due to loss, particularly under finite sized security. Hence, extending QKD beyond individual links requires switching, relaying and route selection [13–15]. This introduces a network-level optimization problem in which the network must determine an optimal way to establish a secure key route. This problem cannot be reduced to physical distance alone, since key generation, key availability, relay placement, resource utilization and trusted–untrusted node constraints can all influence the route [14, 16–18]. In CV-QKD, this dependence is further strengthened because the secret key rate (SKR) is determined by channel loss, excess noise, state preparation, detection and reconciliation efficiency, and security assumptions with finite sized effects [5, 6, 8, 19]. In a multi-hop network, the end-to-end key rate is limited by the weakest transition along the route [13, 18]. Multipath routing offers a trade off, they can increase the available end-to-end key by combining them or reduce dependence on a single path [13, 20, 21], but each additional route also consumes links and nodes that may be required elsewhere in the network. Practical routing must therefore balance key generation performance against network resource use [22].

A further complication arises from node trust. Trusted relay networks provide a practical way of extending QKD beyond a single quantum link, but trusted key forwarding requires the intermediate nodes on a route to remain within the trust boundary [17]. As a network grows and spans infrastructure controlled by different organizations, trusted node assumption can become restrictive. Measurement device independent QKD provides a complementary approach in which the intermediate measurement stations are not trusted [23]. In continuous-variable measurement-device-independent QKD (CV-MDI-QKD), remote quantum states can be correlated through a publicly announced CV Bell measurement while the measurement station is assigned to the untrusted party [24, 25]. Composable finite size security has subsequently been developed for CV-MDI protocols and for more general Gaussian quantum networks containing untrusted relays [26, 27].

These developments motivate the hybrid network considered here. Trusted nodes establish direct P2P links and untrusted nodes may participate in public CV Bell measurements to connect two trusted nodes. The quality of a transition is determined by total loss and noise, number of relays and their placement, state parameters such as squeezing, and finite sized security penalties [27]. Route selection over such a topology is inherently multi-objective: a useful route should preserve a high bottleneck SKR while limiting distance, hop count, relay use and unnecessary detour, even though these objectives can conflict. Research in CV-QKD provides realistic models incorporating practical implementation effects [5, 8, 19].

CV QKD networking studies have considered dynamic SKR based routing and shortest path selection with optical resource allocation [15, 28]. Recent studies have incorporated composable end-to-end security into routing analyses [29]. Here, we investigate a routing policy that combines transition level composable finite sized key rates with explicit penalties for distance, relay use, transition count and route geometry, and evaluate the resulting key-rate–resource trade off in mixed trusted–untrusted networks.

To investigate these questions, we introduce *Quantum Utility Routing* (QUR), a weighted routing framework for CV-QKD networks. Each transition is assigned a composable finite size SKR under collective Gaussian attacks, and complete end-to-end routes are formed through trusted node key forwarding, with their rate determined by the weakest transition. QUR combines six routing factors and the resulting utility trades cryptographic performance against the network cost of establishing the route.

As a weighted search, QUR benefits from weight optimization, but repeated optimization is impractical in dynamic networks. We therefore consider graph-optimized QUR ($\mathrm{QUR}_{\mathrm{go}}$), where weights are optimized for each graph, and fixed-weight QUR ($\mathrm{QUR}_{\mathrm{fw}}$), a learned score model selects a common weight vector for each network configuration. Similar machine learning approaches have been used in QKD to reduce repeated parameter optimization [30, 31]. $\mathrm{QUR}_{\mathrm{fw}}$ therefore measures how much of the $\mathrm{QUR}_{\mathrm{go}}$ trade off can be retained without repeating the weight optimization on every graph. We compare QUR with two complementary benchmarks. A*, which provides a geometric shortest path reference [32], and Max–Min which selects the route with the largest bottleneck SKR [33].

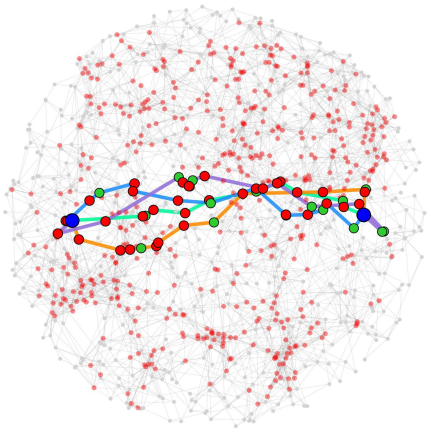
The principal contribution of this work is an integration of well established physical-layer CV-QKD security with network-layer route selection. The framework embeds composable rates directly into routing, accommodates both trusted P2P operation and untrusted Bell measurement relays, and balances SKR against network resource use under our desired criteria. Hence, it enables a direct comparison between short routes, SKR-maximizing routes and routes that seek a controlled compromise between key rate performance and network efficiency in CV-QKD networks.

Results

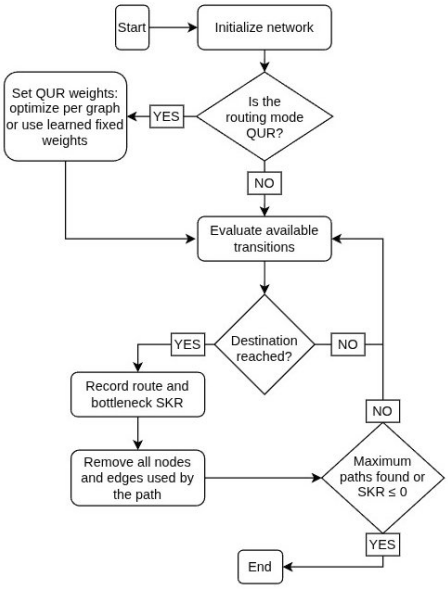
Network setup

Before stating the results, we briefly describe the setup. We simulate metropolitan scale CV quantum networks using a Waxman random graph [18]. Nodes are distributed uniformly over a disk and connected according to the Waxman connection probability. Each node represents a network user equipped with two mode squeezed vacuum (TMSV) sources and the required measurement capabilities. Edges represent optical channels through which neighboring users establish entangled links. Consecutive links can be connected by CV Bell measurements at intermediate nodes, with the outcomes publicly announced, thereby producing an effective entanglement-swapped link between the endpoint users. Each graph contains both trusted and untrusted nodes, with trust status determined by node degree: nodes whose degree exceeds a threshold $\mathcal{D}$ are classified as untrusted (see Figure 1a).

The default setting contains 1000 nodes distributed within a disk of radius 50 km, a degree threshold $\mathcal{D} = 10$,



(a) Example Waxman network. Blue nodes mark the source–destination pair, red nodes are untrusted, and green nodes are trusted. The colored curves show the routes returned after successive path extraction.



(b) Routing workflow for all algorithms. $\mathrm{QUR}_{\mathrm{go}}$ scores random weight vectors, while $\mathrm{QUR}_{\mathrm{fw}}$ uses the ensemble learned weight vector. A*, Max–Min and both QUR modes then find a route, record its bottleneck SKR, remove nodes and edges used, and continue until the path limit is reached or no positive rate route remains.

Figure 1: Simulated network and routing procedure. (a) Example Waxman network at degree threshold $\mathcal{D} = 8$, when almost 53% of the nodes are untrusted. (b) Workflow used by all routing methods.

and a maximum of five node disjoint routes per graph. The source and destination are trusted nodes selected near $(-0.7R, 0)$ and $(0.7R, 0)$, respectively. Each point in a parameter scan is averaged over 50 independently generated graphs.

When the network radius is varied, the link distance is rescaled while maintaining the characteristic distance $r_0 = 0.07R$. This preserves the graph adjacency structure and only increases the link lengths. If no positive rate route is found, we record zero paths and zero SKR, and include that result in the average. The general workflow of all routing algorithms used in this work is shown in Figure 1b.

Quantum Utility Routing

QUR operates on a network whose topology, node positions, link lengths, and trust status are known. During the route search, P2P and relay assisted transitions between trusted nodes are identified and assigned a transition SKR K_ℓ. Transitions with non-positive SKR are excluded.

For a transition ℓ, QUR assigns the cost

$$c_\ell^{\mathrm{QUR}} = q_\ell + w_d d_\ell^{\mathrm{norm}} + w_R r_\ell + w_H h_\ell + w_A a_\ell + w_D \delta_\ell, \tag{1}$$

where

$$q_\ell = \max(0, -\log_{10} K_\ell).$$

Here, K_ℓ is the SKR of transition ℓ evaluated using the security model in Supplementary Note 1 and the parameters in Methods. The six free weights $\boldsymbol{w} = (w_d, w_R, w_H, w_A, w_D, w_T)$ determine the relative importance of the cost factors. Their values control the trade off between SKR performance and network resource efficiency. The cost factors $d_\ell^{\mathrm{norm}}, r_\ell, h_\ell, a_\ell$ and δ_ℓ describe distance, relay use, hop count, movement away from the target, and detour respectively. We explain the mathematics of these coefficients in Methods.

The complete route search procedure is given in Algorithm 1.

Algorithm 1 Quantum Utility Routing (QUR)

Input: Network $\mathcal{G}$, source S, target T, node positions and trust states, maximum number of paths M, and QUR weight vector $\boldsymbol{w} = (w_d, w_R, w_H, w_A, w_D, w_T)$.

Output: Up to M routes between S and T and their corresponding bottleneck SKRs.

1: **Step 1: Initialize the search.** Begin at the source S with zero accumulated QUR cost and assign infinite cost to all other trusted nodes.

2: **Step 2: Explore transitions from the current node.** Identify all P2P and relay assisted transitions from the current trusted node to other trusted nodes. For each transition ℓ, determine its SKR K_ℓ and calculate its QUR cost c_ℓ^{QUR} using Eq. 1.

3: **Step 3: Update and rank the reachable nodes.** For each transition $\ell = (u, v)$, calculate the accumulated cost

$$C_{\mathrm{QUR}}(v) = \sum_{\ell \in \mathcal{R}_{S \to u}} c_\ell^{\mathrm{QUR}} + c_\ell^{\mathrm{QUR}}(v),$$

and rank the reachable nodes using

$$f_{\mathrm{QUR}}(v) = C_{\mathrm{QUR}}(v) + w_T \frac{\|\boldsymbol{x}_v - \boldsymbol{x}_T\|_2}{R}.$$

If more than one route reaches the same node, keep the route with the lower accumulated cost. If the costs are equal, keep the route with the larger bottleneck SKR.

4: **Step 4: Explore the next node.** Select the node with the lowest f_{QUR}, explore all of its outgoing transitions as in Step 2, and rank the reachable nodes again. Repeat Steps 2–4 until the target T is reached. The SKR of the selected route is

$$K_{\mathcal{R}} = \min_{\ell \in \mathcal{R}} K_\ell.$$

5: **Step 5: Search for additional routes.** Record the selected route, exclude its used physical edges and internal nodes, and repeat Steps 1–4 until M routes are found or no further route is available.

$\ell = (u, v)$ denotes a transition from the currently explored trusted node u to a reachable trusted node v. The vector $\boldsymbol{x}_u$ gives the spatial position of node u, while $\boldsymbol{x}_T$ gives the target position. $\|\cdot\|_2$ denotes the Euclidean norm and R denotes the network radius.

For comparison, the search priorities of QUR, A* and Max–Min for a partial route are

$$\begin{aligned} f_{\mathrm{QUR}}(v) &= C_{\mathrm{QUR}}(v) + w_T \frac{\|\boldsymbol{x}_v - \boldsymbol{x}_T\|_2}{R}, \\ f_{A^*}(v) &= \sum_{\ell \in \mathcal{R}_{S \to v}} d_\ell + \|\boldsymbol{x}_v - \boldsymbol{x}_T\|_2, \\ f_{\mathrm{MM}}(v) &= \max_{\mathcal{R} \in \mathcal{R}_{S \to v}} \min_{\ell \in \mathcal{R}} K_\ell. \end{aligned} \tag{2}$$

Thus, A* primarily favors geometric progress, Max–Min favors the strongest bottleneck SKR, while QUR combines transition quality, route geometry, and network resource use. The weight vector $\boldsymbol{w} = (w_d, w_R, w_H, w_A, w_D, w_T)$ is specific to QUR. Neither A* nor Max–Min uses an optimized weight vector.

Two QUR operating modes are considered. In graph optimized QUR ($\mathrm{QUR}_{\mathrm{go}}$), the six weights are optimized independently for each generated network using Monte Carlo sampling. In fixed weight QUR ($\mathrm{QUR}_{\mathrm{fw}}$), a histogram based gradient boosting model is trained using graph characteristics and the performance of randomly sampled weight vectors, from which a common weight vector is obtained for each network configuration. Hence, the distinction between $\mathrm{QUR}_{\mathrm{go}}$ and $\mathrm{QUR}_{\mathrm{fw}}$ lies only in how the weight vector is selected. For both modes, the weight vector optimization follows Algorithm 2.

Algorithm 2 Selection of QUR weight vectors

Input: Network ensemble and QUR weight space $\mathcal{W}$.
Output: Graph-specific weights for QUR_{go} and one common weight vector for QUR_{fw} for the network configuration.

1: **Step 1: Evaluate sampled weights.** For each graph, sample 8,000 distinct weight vectors and evaluate each vector using Algorithm 1. Assign each vector the score $S(\boldsymbol{w})$ defined in Eq. 25.
2: **Step 2: Select QUR_{go}.** For each graph independently, select the weight vector with the highest score.
3: **Step 3: Learn the fixed weight model.** Train the histogram based gradient boosting model using the graph properties, tested weight vectors, and their transformed scores.
4: **Step 4: Select QUR_{fw}.** Form a common weight pool of the highest scoring vectors of each graph with additional random vectors. Evaluate their predicted performance and select the vector with the highest mean relative performance.

The weight space $\mathcal{W}$ used in this work can be found in Methods.

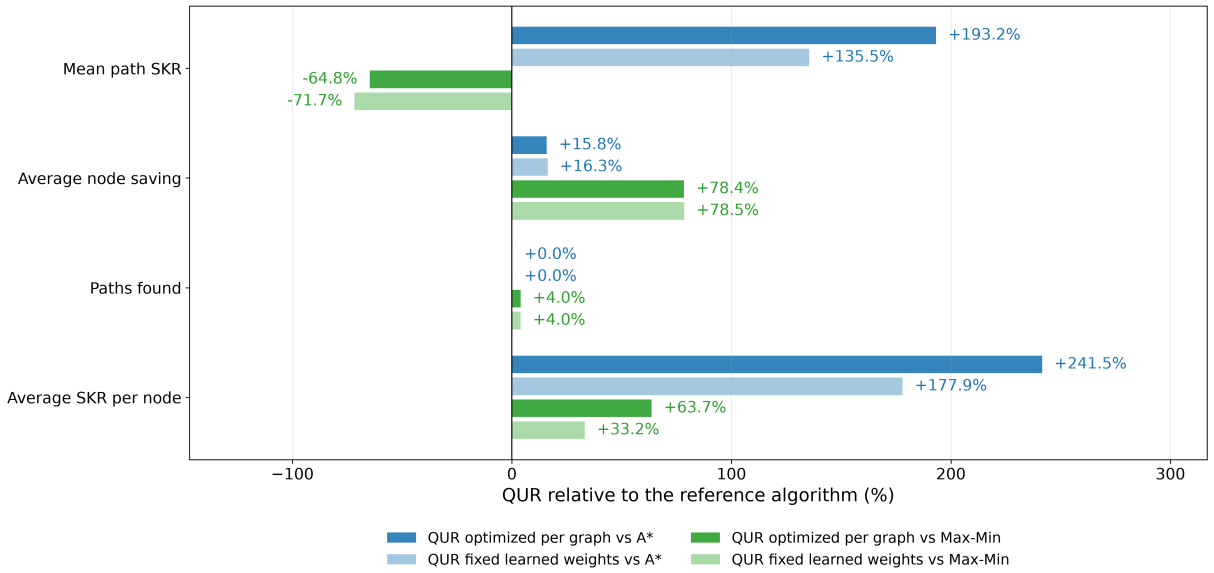


Figure 2: Overall comparison at the default network setting. Bars report QUR_{go} and QUR_{fw} relative to A* or Max–Min. Average node saving is the mean of internal and unique node saving. Average SKR per node is the mean of relative changes in the two efficiency measures defined in Eqs. 4 and 5. Positive values favor the corresponding QUR mode.

Routing performance and resource efficiency

Figure 2 reveals three distinct routing regimes. A* preferentially selects geometrically short routes, but these routes can contain weak quantum transitions and therefore provide relatively low bottleneck SKR. Max–Min takes the opposite approach, it maximizes the weakest transition along the route, but does this at the cost of substantially larger node use. QUR occupies the intermediate operating regime, improving the bottleneck SKR of the short routes favored by A* while avoiding the large resource requirement associated with Max–Min.

This distinction becomes clearer when we look at SKR efficiency from Figure 3. Both QUR modes outperform the reference algorithms in the SKR-per-node comparison, showing that their advantage is not simply associated with short routes or high rate routes individually. Rather, QUR improves the SKR obtained from the intermediate node resources available to a connection.

At the default configuration, QUR_{go} increases the mean path SKR by 193.2% relative to A*, while using 78.4% fewer nodes than Max–Min. QUR_{fw} shows the same qualitative behavior.

Importantly, this resource advantage is not obtained by reducing multipath availability. The mean number of accepted paths remains similar across the routing methods. QUR therefore achieves its improved resource

efficiency primarily by changing which routes are selected, rather than by accepting fewer routes.

The fixed weight result is important because removing graph specific optimization changes the resource use behavior of QUR marginally. Although $\mathrm{QUR}_{\mathrm{fw}}$ sacrifices some SKR relative to $\mathrm{QUR}_{\mathrm{go}}$, it retains almost the same node saving and multipath behavior. This indicates that the useful QUR operating regime is not produced only by highly specialized weights for individual graphs. Instead, a common weighting vector appears to capture much of the routing behavior across the network ensemble.

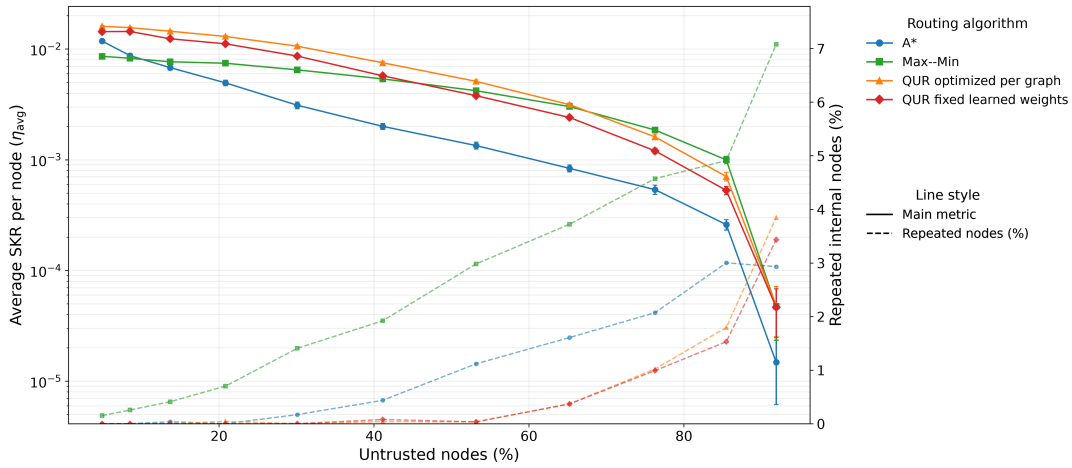


Figure 3: Average SKR efficiency and node reuse as functions of the untrusted node percentage. Solid curves show η_{avg} from Eq. 6. Dashed curves show repeated nodes as a percentage of all nodes.

Routing through highly untrusted networks

A node v is labeled untrusted when its degree satisfies $\deg(v) > \mathcal{D}$. Decreasing $\mathcal{D}$ reduces the trusted node population and provides a controlled way of testing routing when much of the network cannot participate in trusted key forwarding.

We can observe from Figure 3 and Figure 5 that increasing the untrusted node fraction does not immediately destroy secure connectivity. Because untrusted nodes can still participate as public Bell measurement relays, they remain useful for establishing transitions between trusted nodes rather than being removed entirely from the network. From our simulations, we observe that at $\mathcal{D} = 6$, where 76.3% of the nodes are untrusted, $\mathrm{QUR}_{\mathrm{go}}$ still achieves 100.0% routing success and finds approximately 3.92 routes per graph. Thus, a trusted minority can continue to support end-to-end key distribution when the surrounding untrusted infrastructure remains available as untrusted relays.

A different regime appears at the most severe trust restriction. At $\mathcal{D} = 4$, we recorded that approximately 91.9% of the network is untrusted and routing success falls sharply. At this point the limiting factor is no longer only the SKR of relay assisted transitions but also that the remaining trusted nodes become too sparse to form a positive SKR connection between the endpoints.

Hence the trust scan shows a transition from a rate-limited regime to a trusted-connectivity-limited regime in which too few trusted nodes remain to support end-to-end routing. Allowing untrusted Bell measurement relays substantially extends the former regime, but cannot compensate indefinitely.

Multipath capability

Max–Min can leave fewer residual routes because a high bottleneck route may consume many nodes before the next search begins. On a graph-by-graph basis, both modes of QUR match the A* path count in 100.0% of graphs and the Max–Min path count in 84.0%.

Multiple node disjoint routes can be used in two ways. Their keys may be used independently to increase the end-to-end SKR, or path keys may be combined at the endpoints,

$$k_{\mathrm{final}} = k_1 \oplus k_2 \oplus \cdots \oplus k_m, \tag{3}$$

where k is the final key string. If the path keys are independently generated and at least one remains secret conditioned on the other path keys and Eve's information, the XOR key remains secret. This use requires authenticated communication, equal key lengths and trusted endpoint processing [18, 20]. In the default setting, both QUR modes match the A* path count on all tested graphs and match the Max–Min path count on most graphs. So the node saving obtained by QUR is not simply a result of accepting fewer paths.

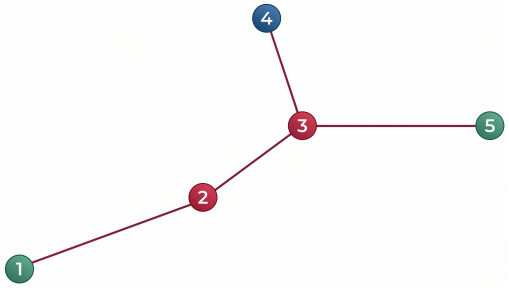


Figure 4: Example of within-route node reuse. Green nodes are trusted and red nodes are untrusted. The path constructed from the route is $1 \to 2 \to 3 \to 4 \to 3 \to 5$, in which node 3 appears in two separate transitions i.e. $1 \to 2 \to 3 \to 4$ and $4 \to 3 \to 5$.

Node reuse and SKR efficiency

The SKR of a route does not by itself describe how efficiently the network is being used so we evaluate the SKR obtained per node. Since nodes and edges may be reused within the same route, we use two measures of node usage. The first counts every node in the path, including repeated ones, labeled internal nodes while the second counts only the distinct nodes used by all paths labeled unique nodes. For each graph, the internal node efficiency is defined as

$$\eta_{\text{int}} = \begin{cases} \overline{K}/\overline{n}, & \overline{n} > 0, \\ 0, & \overline{n} = 0, \end{cases} \tag{4}$$

where $\overline{K}$ is the mean bottleneck SKR of the accepted paths and $\overline{n}$ is their mean number of nodes. For the unique nodes, we define

$$\eta_{\text{unique}} = \begin{cases} K_{\text{sum}}/N_{\text{unique}}, & N_{\text{unique}} > 0, \\ 0, & N_{\text{unique}} = 0, \end{cases} \tag{5}$$

where K_{sum} is the sum of bottleneck SKR of each path and N_{unique} is the total number of unique nodes used by all paths. Their mean,

$$\eta_{\text{avg}} = \frac{\eta_{\text{int}} + \eta_{\text{unique}}}{2}, \tag{6}$$

is used as the overall SKR-per-node measure.

We can observe from Figure 3 that generally, both modes of QUR offer a higher key rate per node used when compared to the two reference algorithms. Importantly, this improvement is not obtained simply by returning fewer paths. The mean number of accepted paths is similar across the routing algorithms, and QUR_{go} matches the A* path count in 100.0% of the tested graphs. Thus, the higher SKR-per-node efficiency reflects a different choice of routes rather than an artificial reduction in multipath availability.

Figure 3 shows how η_{avg} changes as the fraction of untrusted nodes increases, while Figure 5 compares the corresponding mean path SKR and node requirements across the $\mathcal{D}$ and R scans. Both figures also show the percentage of nodes reused isnide each accepted path. As expected, node reuse increases as $\mathcal{D}$ decreases because fewer trusted intermediate stations remain available for routing. This effect is particularly pronounced for Max–Min, whose high nodal requirement rapidly depletes the set of available trusted nodes when multiple paths are constructed.

Discussion

The main implication of these results is that secure routing in a CV-QKD network should not be treated as either a shortest path or a maximum SKR problem alone. These objectives represent different operating extremes: reducing geometric distance does not guarantee strong quantum transitions, whereas maximizing

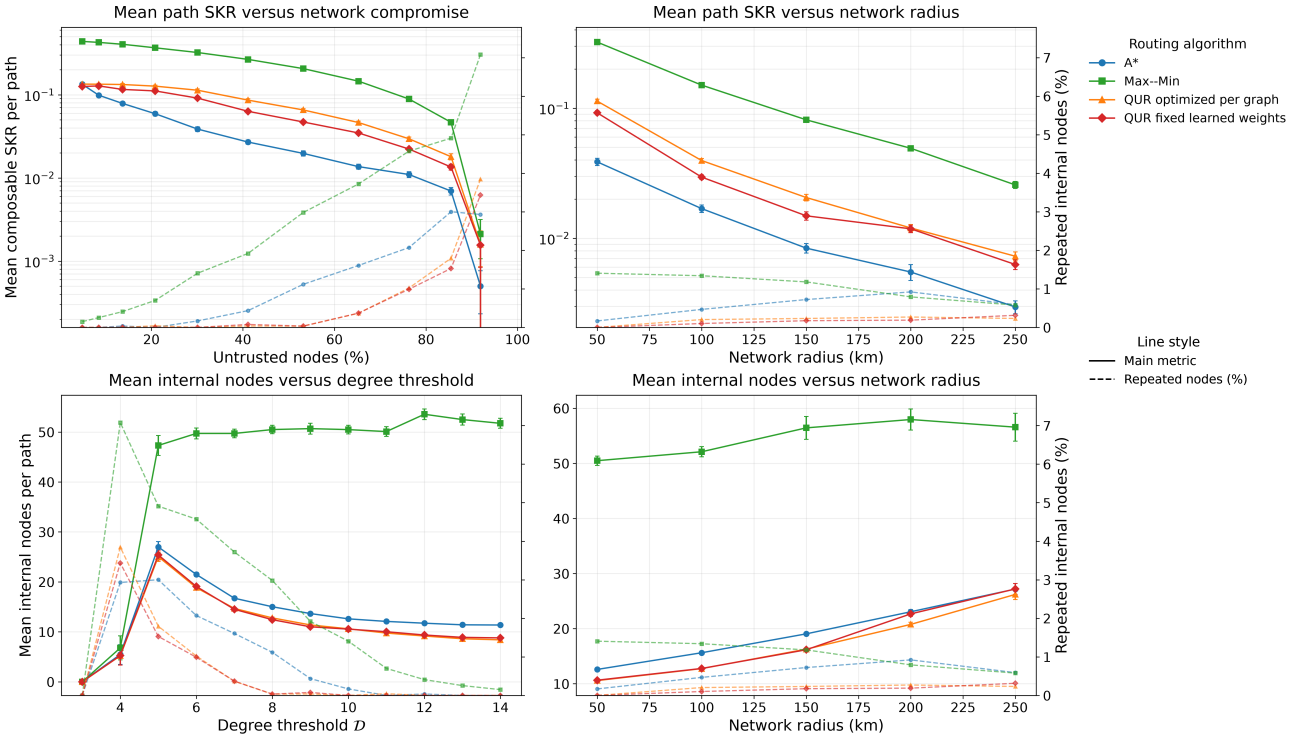


Figure 5: Routing performance and within-route node reuse. The upper panels show the mean composable SKR per path against untrusted node percentage and network radius. The lower panels show the mean number of nodes per path against degree threshold and radius. Dashed curves and the right axes show repeated internal node percentage.

the bottleneck SKR can consume a large fraction of the available network resources. QUR instead treats routing as a configurable trade off between physical-layer key generation and network-layer resource use. The significance of the framework is not a single universally optimal set of weights, but the ability to tune the routing policy according to the requirements of the network.

This flexibility is important for shared quantum networks. Intermediate stations, optical hardware and available trusted nodes are finite resources, and consuming them for one connection can restrict the routes available to subsequent users or paths. From this perspective, node use is not merely an auxiliary routing statistic but an operational cost. The SKR-per-node metric introduced here provides a complementary measure to the conventional bottleneck SKR by asking how effectively the available infrastructure is converted into secure key capacity.

The behavior of node reuse provides a related indication of network pressure. Repeated use of intermediate nodes can allow a route to remain feasible when trusted connectivity becomes limited, but increasing reuse also indicates that fewer structurally distinct alternatives remain. Node reuse should therefore be interpreted both as a mechanism for maintaining connectivity and as an indicator that the routing space is becoming constrained. This becomes particularly relevant for multipaths, where excessive consumption of useful nodes by an early route can reduce the opportunities available to later routes.

The hybrid trusted–untrusted architecture also has an important network design implication. An untrusted node does not necessarily need to be removed from the communication infrastructure. By allowing such nodes to participate through public CV Bell measurements, they can continue to support connectivity without being entrusted with the final end-to-end key. The ultimate limitation is then determined by the availability and spatial distribution of trusted nodes and the feasibility of relay assisted transitions. Consequently, the placement of trusted stations may be as important as their overall fraction when designing large QKD networks.

The distinction between $\mathrm{QUR}_{\mathrm{go}}$ and $\mathrm{QUR}_{\mathrm{fw}}$ is primarily operational. Graph specific optimization provides a useful reference for the best QUR performance identified within the 8,000 sample search budget. The idea is motivated from the fact that repeating a large weight search whenever network conditions change would be difficult in practice. $\mathrm{QUR}_{\mathrm{fw}}$ instead suggests that the expensive optimization stage may be moved offline,

leaving only the route search and transition rate updates to be performed during network operation. This is particularly relevant to dynamic CV-QKD networks, where channel transmissivity and therefore the available SKR can vary with time [15].

The score function should be viewed as a network policy choice rather than a fixed objective. In the present implementation, the optimization score favors reduced node use while discouraging solutions with poor key rate. A network operator could instead emphasize other requirements, such as relay avoidance, higher key throughput, lower latency, multipath availability or resilience. The weighted formulation therefore provides a mechanism for incorporating network-level priorities without modifying the underlying physical-layer security model.

The fixed-weight vector is learned and evaluated on the same ensemble of Waxman network. Present results establish that a common weight vector can represent that ensemble. The methodology described here can extend to unseen topologies. We can also extend the simulations to include other factors such as heterogeneous equipment, time dependent channels and trust assignments based on ownership, location or operational history.

In the present framework, composability is applied at the transition level. Recent work has shown that end-to-end composable security introduces a security–rate trade off in which a total security budget must be distributed across all of the transitions forming the route [29]. This creates a coupling between routing and security allocation: the number and type of transitions are not known until a route has been constructed, while their achievable SKRs depend on the security budget assigned to all of them. This coupling is avoided by assigning a fixed and relatively strict security parameter to each transition. For a route $\mathcal{R}$ containing $L_{\mathcal{R}}$ transitions, composable security gives the route-level bound

$$\epsilon_{\mathcal{R}} \leq \sum_{\ell \in \mathcal{R}} \epsilon_{\mathrm{tr},\ell}. \tag{7}$$

For the uniform transition security used here, this reduces to $\epsilon_{\mathcal{R}} \leq L_{\mathcal{R}} \epsilon_{\mathrm{tr}}$. Consequently, the end-to-end security parameter is not fixed a priori, but increases with the number of transitions forming the selected route.

Methods

Transition rate evaluation and parameters

We evaluate direct and relay assisted transitions using the established Gaussian network security model under collective Gaussian attacks [27]. The covariance matrix and composable finite size expressions are given in Supplementary Note 1. A transition contains $L_\ell = 1, \ldots, 4$ physical links and $L_\ell - 1$ untrusted relays. The noise ξ is output referenced, η_{B} is the Bell-measurement efficiency, and the fiber attenuation is $\alpha = 0.2$ dB/km.

Throughout the work, we use

$$\mathcal{N} = 10^9, \quad f_{\mathrm{PE}} = 0.10, \quad \beta = 0.98, \quad \mathrm{FER} = 0.10, \quad |\mathcal{X}| = 32, \quad \xi = 0.01, \quad \eta_{\mathrm{B}} = 0.98, \tag{8}$$

and

$$\epsilon_{\mathrm{PE}} = \epsilon_s = \epsilon_h = \epsilon_c = 10^{-10}. \tag{9}$$

For the parameters used here, this gives $\epsilon_{\mathrm{tr}} = 3.9 \times 10^{-10}$. For P2P transitions, the SKR is optimized over $r_{\mathrm{dB}} \in [1, 30]$ dB.

Optimizing nested relay transitions is computationally intensive, so we use a discrete hill climb search. We initialize at $r_{\mathrm{dB}} = 14$ dB and evaluate the SKR at the neighboring values $r_{\mathrm{dB}} \pm 1$ dB. If either neighbor give a higher SKR, the search proceeds in that direction in increments of 1 dB until no further improvement is obtained or the boundary $1 \leq r_{\mathrm{dB}} \leq 30$ is reached.

Network model and simulation parameters

We use an undirected Waxman graph $G = (V, E)$ with $N_V = |V|$ nodes distributed uniformly within a disk of radius R. The sampling and connection probability formulas are given in Supplementary Note 2 [18].

Node trust is assigned from its degree, with node v treated as untrusted when

$$\deg(v) > \mathcal{D}, \tag{10}$$

where $\mathcal{D}$ is the trust threshold.

The default settings are

$$N_V = 1000, \qquad R = 50\,\text{km}, \qquad \beta_W = 1, \qquad r_0 = 0.07R, \qquad \mathcal{D} = 10, \qquad M = 5, \tag{11}$$

where M is the maximum number of routes considered.

To examine the effects of network scale and trust, we independently vary

$$R \in 50, 100, 150, 200, 250 \text{ km}, \tag{12}$$

and

$$\mathcal{D} \in 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, \tag{13}$$

while keeping all remaining parameters fixed at their default values. Each parameter point is averaged over 50 independently generated graphs per setting.

Route and multipath construction

A route $\mathcal{R}$ is formed as a sequence of transitions between trusted nodes. Once a route is found, each transition is expanded into its underlying node sequence to obtain the corresponding path p, defined as the ordered sequence of all nodes used by that route.

For a route $\mathcal{R}$ consisting of transitions ℓ, the end-to-end SKR is limited by its weakest transition,

$$K(\mathcal{R}) = \min_{\ell \in \mathcal{R}} K_\ell. \tag{14}$$

Rates are expressed in secret bits per transition use. The path p inherits the same end-to-end rate.

Multipath construction is iterative. After a route is accepted, it is expanded into its path and every node and edge appearing in that path is removed from the network. The source and destination are retained. The routing search is then repeated until M routes are found or no positive rate route remains. Accepted paths are therefore internally node and edge disjoint from one another.

For m accepted routes $\mathcal{R}_j$ and their corresponding paths p_j, we define the following quantities

$$\begin{aligned} K_{\text{sum}} &= \sum_{j=1}^{m} K(\mathcal{R}_j), \\ \overline{K} &= \frac{1}{m} K_{\text{sum}}, \\ \overline{n} &= \frac{1}{m} \sum_{j=1}^{m} n(p_j), \end{aligned} \tag{15}$$

where K_{sum} is the sum of bottleneck key rates, $\overline{K}$ is the average SKR per path, and $n(p_j)$ is the number of nodes in the path, including repeated appearances of the same node. This gives us $\overline{n}$ as the average number of nodes used by those paths. The source and destination are excluded from this count. A simulation in which no route is found is recorded with $\overline{K} = \overline{n} = p = 0$.

The reference algorithms are geometric A* routing [32] and maximum bottleneck Max–Min routing [33], described in Supplementary Notes 3 and 4, respectively. Both use the same transition rates and iterative resource removal procedure defined above.

Quantum Utility Routing cost function and search priority

For a transition $\ell = (u, v)$, where u is the current trusted node and v is a reachable trusted node, we define

$$\begin{aligned} q_\ell &= \max(0, -\log_{10} K_\ell), \\ r_\ell &= \text{number of untrusted relays in } \ell, \\ \delta_\ell &= \frac{1}{R} \max(0, d_\ell - \|\boldsymbol{x}_u - \boldsymbol{x}_v\|_2), \\ a_\ell &= \begin{cases} 0, & r_\ell = 0, \\ \dfrac{1}{R} \displaystyle\sum_j \max[0, \|\boldsymbol{x}_{j+1} - \boldsymbol{x}_T\|_2 - \|\boldsymbol{x}_j - \boldsymbol{x}_T\|_2], & r_\ell \geq 1, \end{cases} \end{aligned} \tag{16}$$

where d_ℓ is the total physical length of the transition and the index j runs over its ordered node sequence. Physically, q_ℓ penalizes low SKR, r_ℓ counts relay use, a_ℓ penalizes backward movement within relay transitions, and δ_ℓ penalizes relay detour from the direct distance between the trusted endpoints. For a direct P2P transition, $r_\ell = a_\ell = \delta_\ell = 0$. From this, the QUR cost assigned to transition ℓ is given as

$$c_\ell^{\mathrm{QUR}} = q_\ell + w_d \frac{d_\ell}{R} + w_R r_\ell + w_H + w_A a_\ell + w_D \delta_\ell. \tag{17}$$

Here, the constant term w_H assigns one hop cost to each transition. For a partial route $\mathcal{R}_{S\to u}$, extending it from u to v through $\ell = (u, v)$ gives

$$C_{\mathrm{QUR}}(v) = \sum_{\ell \in \mathcal{R}_{S\to u}} c_\ell^{\mathrm{QUR}} + c_\ell^{\mathrm{QUR}}(v). \tag{18}$$

The search priority assigned to the reachable node v is

$$f_{\mathrm{QUR}}(v) = C_{\mathrm{QUR}}(v) + w_T \frac{\|\boldsymbol{x}_v - \boldsymbol{x}_T\|_2}{R}. \tag{19}$$

QUR is a heuristic search whose additive cost represents the desired trade off between transition quality and resource use. For each reachable node, the route with the lower accumulated QUR cost is retained. If the costs are equal, the route with the larger bottleneck SKR is preferred. The reachable node with the lowest $f_{\mathrm{QUR}}(v)$ is explored next. The six free QUR weights that control the penalties are

$$\boldsymbol{w} = (w_d, w_R, w_H, w_A, w_D, w_T). \tag{20}$$

Monte Carlo selection of graph specific QUR weights

Every weight in $\boldsymbol{w}$ is independently sampled from

$$\mathcal{W} = \{0.00, 0.01, \ldots, 1.00\}. \tag{21}$$

This creates a space containing

$$|\mathcal{W}|^6 = 101^6 = 1{,}061{,}520{,}150{,}601 \tag{22}$$

possible vectors. Checking every combination is hence impractical. For each graph, 8,000 distinct vectors are sampled from this grid and evaluated by the QUR routing model. Every weight vector searches for up to M routes.

We associate a score $S(\boldsymbol{w})$ with the performance of each sampled wight vector. A vector with no positive rate route receives score $-\infty$. For $0 < \overline{K} < 1$ and $\overline{n} > 0$, the base score is

$$S_0(\boldsymbol{w}) = -\frac{1}{\overline{n}(\boldsymbol{w}) \log_{10} \overline{K}(\boldsymbol{w})}. \tag{23}$$

If a direct source–destination route gives $\overline{n} = 0$, we assign an infinite base score and hence we always prefer it.

The A* rate penalty is

$$\delta_{A^*}(\boldsymbol{w}) = \max\left[0, \frac{\overline{K}_{A^*} - \overline{K}(\boldsymbol{w})}{\overline{K}_{A^*}}\right], \tag{24}$$

where $\overline{K}_{A^*}$ is the mean bottleneck SKR of the routes returned by A*, and $\overline{K}(\boldsymbol{w})$ is the corresponding mean bottleneck SKR obtained by QUR using the weight vector $\boldsymbol{w}$. The final score is

$$S(\boldsymbol{w}) = S_0(\boldsymbol{w}) \exp[-\gamma \delta_{A^*}(\boldsymbol{w})], \qquad \gamma = 0.5. \tag{25}$$

The parameter γ controls the strength of the penalty applied when the QUR mean bottleneck SKR falls below the A* reference value. This score function is used in Algorithm 2 to pick weight vectors that follow our defined criteria. An example of $S(\boldsymbol{w})$ for a fixed network configuration is shown in Figure 6.

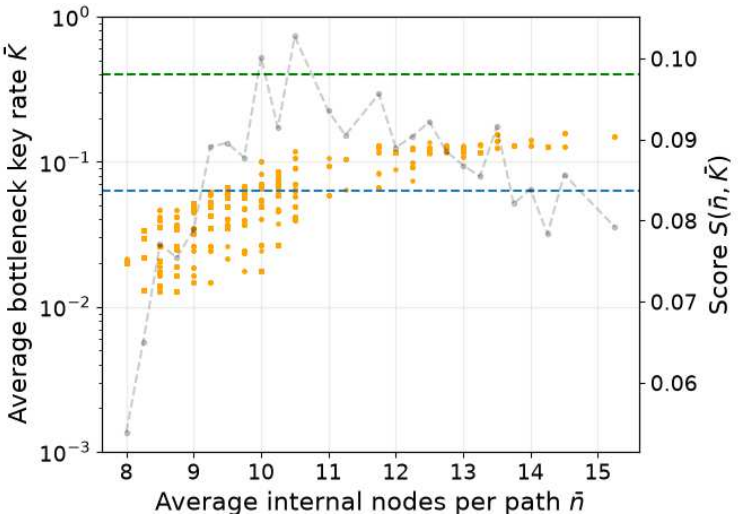


Figure 6: Example Monte Carlo weight search on one graph. Each orange point is one sampled QUR weight vector, shown by its mean node $\bar{n}$ count and mean bottleneck SKR $\bar{K}$. They gray dashed line is the score value plotted for the largest $\bar{K}$ at each $\bar{n}$. The A* and Max–Min mean rates are shown as the dashed blue and green reference lines respectively. In this simulation, 8,000 vectors are sampled and evaluated on a graph created at default values. On average, A* used 11.75 nodes and Max–Min used 49.

Machine-learned selection of fixed QUR weights

QUR_{go} selects the highest scoring weight vector among the tested samples, but requires a new 8,000 random weight search for every graph. QUR_{fw} is introduced to avoid this repeated optimization. It uses exactly the same QUR routing cost in Eqs. 1–19, but replaces the graph specific weights with one weight vector selected from the ensemble of 50 graphs per configuration.

The machine learning model learns how the QUR score changes with the graph properties and the six free QUR weights. For graph g, we denote the graph information by $\boldsymbol{y}_g$ and the ith tested weight vector by $\boldsymbol{w}_{g,i}$. The input to the model is therefore

$$\boldsymbol{z}_{g,i} = \begin{bmatrix} \boldsymbol{y}_g \\ \boldsymbol{w}_{g,i} \end{bmatrix}, \tag{26}$$

where $\boldsymbol{w}_{g,i} \in \mathbb{R}^6$ contains the six free QUR weights.

The vector $\boldsymbol{y}_g$ contains information about the graph on which the weights are tested. This includes quantities already defined above, such as the number of nodes N_V, network radius R, degree threshold $\mathcal{D}$, maximum number of paths M, and the number of edges. It also contains quantities calculated from the particular graph, including the fractions of trusted and untrusted nodes, node degree and fiber lengths, the source–destination separation, and the corresponding A* results. Thus, $\boldsymbol{y}_g$ describes the network conditions under which a QUR weight vector is tested.

For every graph, the 8,000 QUR weight vectors are first evaluated using the exact routing model. Each graph–weight pair has an exact score $S_{g,i}$ from Eq. 25. The machine-learning model is trained to reproduce how this score changes with $\boldsymbol{y}_g$ and $\boldsymbol{w}_{g,i}$. Before training, the score is transformed according to

$$S'_{g,i} = \begin{cases} \ln\left(1 + 10^6\right), & S_{g,i} = +\infty, \\ \ln(1 + S_{g,i}), & 0 < S_{g,i} < \infty, \\ 0, & \text{otherwise}, \end{cases} \tag{27}$$

which reduces the effect of very large score values during training. When all 50 graphs provide usable QUR results, the model is trained using up to 50 × 8,000 exact graph–weight evaluations.

We use histogram based gradient boosting [30, 31], whose standard regression tree formulation is summarized in Supplementary Note 5. The settings are a maximum of $T_{\text{ML}} = 300$ trees, a learning rate $\lambda_{\text{ML}} = 0.06$, at most 31 leaf nodes, at least 20 training samples per leaf, and an ℓ_2 regularization value of 1.

The trained model does not directly generate the fixed QUR weights. Instead, it is used to compare a finite selection of potential weight vectors. For each graph, the 500 highest scoring vectors from the 8,000 QUR trials are retained. A further 10,000 randomly sampled vectors are added to explore weight combinations that may perform well across several graphs even if they were not among the highest scoring vectors of any individual graph. The resulting fixed weight vector is chosen from

$$\mathcal{W}_{\text{fw}} = \bigcup_{g=1}^{G' \leq 50} \text{Top}_{500}(\mathcal{W}_g) \cup \mathcal{W}_{\text{rand}}, \tag{28}$$

where G' is the number of usable graphs, $\mathcal{W}_g$ is the set of 8,000 weight vectors tested on graph g and $\mathcal{W}_{\text{rand}}$ contains the additional 10,000 random weight vectors. The model predicts the score of every weight vector in $\mathcal{W}_{\text{fw}}$ for each graph. These predictions are normalized by the highest predicted score for that graph, and the vector with the highest average normalized score is selected.

Averaging and plotted comparisons

For each graph $g = 1, \ldots, 50$, two measures of SKR per node are calculated. The first measure uses the mean number of nodes per path,

$$\eta_{\text{int},g} = \frac{\overline{K}_g}{\overline{n}_g}, \tag{29}$$

where $\overline{n}_g$ is the mean number of nodes per accepted path, which includes repeated nodes, and $\overline{K}_g$ is the average bottleneck SKR.

The second measure uses the number of unique nodes across all accepted paths,

$$\eta_{\text{unique},g} = \frac{K_{\text{sum},g}}{n_{\text{unique},g}}, \tag{30}$$

where $n_{\text{unique},g}$ is the number of unique nodes used across all accepted paths.

The average SKR-per-node measure is then

$$\eta_{\text{avg},g} = \frac{\eta_{\text{int},g} + \eta_{\text{unique},g}}{2}. \tag{31}$$

These quantities are calculated separately for each graph before averaging over the graph ensemble. The error bars shown for the curves are the standard errors calculated from the 50 graph values.

Node saving is calculated separately using the mean number of nodes per path and the number of unique nodes,

$$\begin{aligned} s_{\text{int}} &= 100 \left(1 - \frac{\langle \overline{n} \rangle_{\text{QUR}}}{\langle \overline{n} \rangle_{\text{ref}}}\right), \\ s_{\text{unique}} &= 100 \left(1 - \frac{\langle n_{\text{unique}} \rangle_{\text{QUR}}}{\langle n_{\text{unique}} \rangle_{\text{ref}}}\right). \end{aligned} \tag{32}$$

Here, $\langle \cdot \rangle$ denotes the average of the corresponding quantity over the 50 graph ensemble. The subscript "ref" denotes the reference routing algorithm, either A* or Max–Min. The average node saving shown in Figure 2 is

$$s_{\text{avg}} = \frac{s_{\text{int}} + s_{\text{unique}}}{2}. \tag{33}$$

Repeated node use is determined by comparing the total number of nodes used across all accepted paths with the number of unique nodes. For each plotted configuration, the repeated node percentage is

$$P_{\text{reuse}} = 100 \; \frac{\sum_{g=1}^{50} (m_g \overline{n}_g - n_{\text{unique},g})}{\sum_{g=1}^{50} m_g \overline{n}_g}, \tag{34}$$

where m_g is the number of accepted paths in graph g. Because accepted paths are internally node disjoint, this quantity measures repeated nodes within individual paths rather than reuse between different accepted paths. Thus, $P_{\text{reuse}} = 0$ means that every node used across the accepted paths is unique. A larger value indicates that some nodes are used more than once.

Use of generative artificial intelligence

ChatGPT (OpenAI), Gemini (Google), and Claude (Anthropic) were used to assist with code optimization, multiprocessing and caching, development of the machine-learning implementation and figure aesthetics. All AI-assisted outputs were reviewed, tested, and verified by the authors, who retained full responsibility for the final work.

Data availability

The data generated during the study is available from the corresponding author upon reasonable request.

Code availability

The custom Python code used to generate the results reported in this study is publicly available at https://github.com/Farsad-A/qur.

References

[1] Valerio Scarani, Helle Bechmann-Pasquinucci, Nicolas J. Cerf, Miloslav Dušek, Norbert Lütkenhaus, and Momtchil Peev. The security of practical quantum key distribution. *Rev. Mod. Phys.*, 81:1301–1350, Sep 2009. doi: 10.1103/RevModPhys.81.1301. URL https://link.aps.org/doi/10.1103/RevModPhys.81.1301.

[2] Frédéric Grosshans and Philippe Grangier. Continuous variable quantum cryptography using coherent states. *Phys. Rev. Lett.*, 88:057902, Jan 2002. doi: 10.1103/PhysRevLett.88.057902. URL https://link.aps.org/doi/10.1103/PhysRevLett.88.057902.

[3] Christian Weedbrook, Stefano Pirandola, Raúl García-Patrón, Nicolas J Cerf, Timothy C Ralph, Jeffrey H Shapiro, and Seth Lloyd. Gaussian quantum information. *Reviews of Modern Physics*, 84(2):621, 2012.

[4] T. C. Ralph. Security of continuous-variable quantum cryptography. *Phys. Rev. A*, 62:062306, Nov 2000. doi: 10.1103/PhysRevA.62.062306. URL https://link.aps.org/doi/10.1103/PhysRevA.62.062306.

[5] Fabian Laudenbach, Christoph Pacher, Chi-Hang Fred Fung, Andreas Poppe, Momtchil Peev, Bernhard Schrenk, Michael Hentschel, Philip Walther, and Hannes Hübel. Continuous-variable quantum key distribution with gaussian modulation—the theory of practical implementations. *Advanced Quantum Technologies*, 1(1), June 2018. ISSN 2511-9044. doi: 10.1002/qute.201800011. URL http://dx.doi.org/10.1002/qute.201800011.

[6] Vladyslav C. Usenko, Antonio Acín, Romain Alléaume, Ulrik L. Andersen, Eleni Diamanti, Tobias Gehring, Adnan A. E. Hajomer, Florian Kanitschar, Christoph Pacher, Stefano Pirandola, and Valerio Pruneri. Continuous-variable quantum communication. *Rev. Mod. Phys.*, 98:015003, Mar 2026. doi: 10.1103/mgj7-t6d3. URL https://link.aps.org/doi/10.1103/mgj7-t6d3.

[7] G. Zhang, J. Y. Haw, H. Cai, F. Xu, S. M. Assad, J. F. Fitzsimons, X. Zhou, Y. Zhang, S. Yu, J. Wu, W. Ser, L. C. Kwek, and A. Q. Liu. An integrated silicon photonic chip platform for continuous-variable quantum key distribution. *Nature Photonics*, 13(12):839–842, August 2019. ISSN 1749-4893. doi: 10.1038/s41566-019-0504-5. URL http://dx.doi.org/10.1038/s41566-019-0504-5.

[8] Nitin Jain, Hou-Man Chin, Hossein Mani, Cosmo Lupo, Dino Solar Nikolic, Arne Kordts, Stefano Pirandola, Thomas Brochmann Pedersen, Matthias Kolb, Bernhard Ömer, Christoph Pacher, Tobias Gehring, and Ulrik L. Andersen. Practical continuous-variable quantum key distribution with composable security. *Nature Communications*, 13(1), August 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-32161-y. URL http://dx.doi.org/10.1038/s41467-022-32161-y.

[9] Hans H. Brunner, Chi-Hang Fred Fung, Momtchil Peev, Rubén B. Méndez, Laura Ortiz, Juan P. Brito, Vicente Martín, José M. Rivas-Moscoso, Felipe Jiménez, Antonio A. Pastor, and Diego R. López. Demonstration of a switched cv-qkd network. *EPJ Quantum Technology*, 10(1), 2023. ISSN 2196-0763. doi: 10.1140/epjqt/s40507-023-00194-x. URL http://dx.doi.org/10.1140/epjqt/s40507-023-00194-x.

[10] Chip Elliott, Alexander Colvin, David Pearson, Oleksiy Pikalo, John Schlafer, and Henry Yeh. Current status of the darpa quantum network, 2005. URL https://arxiv.org/abs/quant-ph/0503058.

[11] M. Sasaki, M. Fujiwara, H. Ishizuka, W. Klaus, K. Wakui, M. Takeoka, S. Miki, T. Yamashita, Z. Wang, 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*, 19(11):10387–10409, May 2011. doi: 10.1364/OE.19.010387. URL https://opg.optica.org/oe/abstract.cfm?URI=oe-19-11-10387.

[12] 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 Journal of Physics*, 11(7):075001, jul 2009. doi: 10.1088/1367-2630/11/7/075001. URL https://doi.org/10.1088/1367-2630/11/7/075001.

[13] Stefano Pirandola. End-to-end capacities of a quantum communication network. *Communications Physics*, 2(1), May 2019. ISSN 2399-3650. doi: 10.1038/s42005-019-0147-3. URL http://dx.doi.org/10.1038/s42005-019-0147-3.

[14] Omar Amer, Walter O. Krawec, and Bing Wang. Efficient routing for quantum key distribution networks. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 137–147, 2020. doi: 10.1109/QCE49297.2020.00027.

[15] Mikhael T. Sayat, Sebastian P. Kish, Ping Koy Lam, Nicholas J. Rattenbury, and John E. Cater. Dynamic continuous variable quantum key distribution for securing a future global quantum network. *Advanced Quantum Technologies*, 8(10), 2025. ISSN 2511-9044. doi: 10.1002/qute.202500135. URL http://dx.doi.org/10.1002/qute.202500135.

[16] Jiameng Yao, Yaxing Wang, Qiong Li, Haokun Mao, Ahmed A. Abd El-Latif, and Nan Chen. An efficient routing protocol for quantum key distribution networks. *Entropy*, 24(7), 2022. ISSN 1099-4300. doi: 10.3390/e24070911. URL https://www.mdpi.com/1099-4300/24/7/911.

[17] Tommy van Duijn, Sebastian Verschoor, Simon Rommel, and Idelfonso Tafur Monroy. Routing strategies for quantum key distribution networks based on trusted relay nodes. In *International Conference on Optical Network Design and Modeling (ONDM)*, 2024.

[18] Cillian Harney and Stefano Pirandola. Practical routing and criticality in large-scale quantum communication networks. *Phys. Rev. Res.*, 7:043168, Nov 2025. doi: 10.1103/vy37-28jc. URL https://link.aps.org/doi/10.1103/vy37-28jc.

[19] Anthony Leverrier. Composable security proof for continuous-variable quantum key distribution with coherent states. *Phys. Rev. Lett.*, 114:070501, Feb 2015. doi: 10.1103/PhysRevLett.114.070501. URL https://link.aps.org/doi/10.1103/PhysRevLett.114.070501.

[20] Cheng Liu, Xuanxuan Che, Jianshe Xie, and Yumin Dong. A multi-path qkd algorithm with multiple segments. *Journal of Cyber Security and Mobility*, 13(2):193–214, 2024. doi: 10.13052/jcsm2245-1439.1321.

[21] Evan Sutcliffe and Alejandra Beghelli. Multiuser entanglement distribution in quantum networks using multipath routing. *IEEE Transactions on Quantum Engineering*, 4:1–15, 2023. doi: 10.1109/TQE.2023.3329714.

[22] Joy Halder, Akhmadjon Rajabov, Riccardo Bassoli, Frank H. P. Fitzek, and Gerhard P. Fettweis. Optimal routing and end-to-end entanglement distribution in quantum networks. *Scientific Reports*, 14 (1), August 2024. ISSN 2045-2322. doi: 10.1038/s41598-024-70114-1. URL http://dx.doi.org/10.1038/s41598-024-70114-1.

[23] Hoi-Kwong Lo, Marcos Curty, and Bing Qi. Measurement-device-independent quantum key distribution. *Phys. Rev. Lett.*, 108:130503, Mar 2012. doi: 10.1103/PhysRevLett.108.130503. URL https://link.aps.org/doi/10.1103/PhysRevLett.108.130503.

[24] Zhengyu Li, Yi-Chen Zhang, Feihu Xu, Xiang Peng, and Hong Guo. Continuous-variable measurement-device-independent quantum key distribution. *Physical Review A*, 89(5):052301, 2014. doi: 10.1103/PhysRevA.89.052301.

[25] Stefano Pirandola, Carlo Ottaviani, Gaetana Spedalieri, Christian Weedbrook, Samuel L. Braunstein, Seth Lloyd, Tobias Gehring, Christian S. Jacobsen, and Ulrik L. Andersen. High-rate measurement-device-independent quantum cryptography. *Nature Photonics*, 9(6):397–402, May 2015. ISSN 1749-4893. doi: 10.1038/nphoton.2015.83. URL http://dx.doi.org/10.1038/nphoton.2015.83.

[26] Cosmo Lupo, Carlo Ottaviani, Panagiotis Papanastasiou, and Stefano Pirandola. Continuous-variable measurement-device-independent quantum key distribution: Composable security against coherent attacks. *Phys. Rev. A*, 97:052327, May 2018. doi: 10.1103/PhysRevA.97.052327. URL https://link.aps.org/doi/10.1103/PhysRevA.97.052327.

[27] Masoud Ghalaii, Panagiotis Papanastasiou, and Stefano Pirandola. Composable end-to-end security of gaussian quantum networks with untrusted relays. *npj Quantum Information*, 8:105, 2022. doi: 10.1038/s41534-022-00620-5.

[28] Shifeng Ding, Gangxiang Shen, Fengxian Tang, and Calvin Chun-Kit Chan. Noise-aware resource allocation with integrated key generation and consumption for cv-qkd over wdm networks. *J. Opt. Commun. Netw.*, 16(1):29–44, Jan 2024. doi: 10.1364/JOCN.505197. URL https://opg.optica.org/jocn/abstract.cfm?URI=jocn-16-1-29.

[29] Stefano Pirandola, Cillian Harney, and Gaetana Spedalieri. Security-rate trade-off in quantum key distribution: From individual links to large-scale networks. *Phys. Rev. Res.*, 8:033221, Aug 2026. doi: 10.1103/7839-3xhy. URL https://link.aps.org/doi/10.1103/7839-3xhy.

[30] Duo Ma, Yue Li, Haoyang Wang, Kaiyi Shi, Chang Liu, Yujia Zhang, Fangze Ma, and Haiqiang Ma. Parameter optimization for mode-pairing quantum key distribution based on lightgbm. *Optics Continuum*, 5:611–620, 02 2026. doi: 10.1364/OPTCON.581743.

[31] Qin Dong, Guoqi Huang, Wei Cui, and Rongzhen Jiao. Optimization parameter prediction-based xgboost of tf-qkd. *Quantum Information Processing*, 21(7), 2022. ISSN 1573-1332. doi: 10.1007/s11128-022-03579-6. URL http://dx.doi.org/10.1007/s11128-022-03579-6.

[32] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. doi: 10.1109/TSSC.1968.300136.

[33] Maurice Pollack. Letter to the editor—the maximum capacity through a network. *Operations Research*, 8(5):733–736, October 1960. ISSN 1526-5463. doi: 10.1287/opre.8.5.733. URL http://dx.doi.org/10.1287/opre.8.5.733.

Acknowledgements

The authors gratefully acknowledge Adil Umar Khawaja for assistance in developing the Monte Carlo simulation framework used in this work.

Author contributions

F.A. developed the network model, routing framework, simulations and manuscript. A.K. supervised the project and reviewed the manuscript. Both authors discussed the results and approved the final manuscript.

Competing interests

The authors declare no competing interests.

Funding Declaration

The project is not funded by any resource.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Supplementary.pdf