ARTICLE OPEN

Quantum key distribution in a packet-switched network

Reem Mandil[1,2], Stephen DiAdamo[3]✉, Bing Qi[1] and Alireza Shabani[1]

Packet switching revolutionized the Internet by allowing the efficient use of network resources for data transmission. In a previous work, we introduced packet switching in quantum networks as a path to the Quantum Internet and presented a proof-of-concept for its application to quantum key distribution (QKD). In this paper, we outline a three-step approach for key rate optimization in a packet-switched network. Our simulated results show that practical key rates may be achieved in a sixteen-user network with no optical storage capacity. Under certain network conditions, we may improve the key rate by using an ultra-low-loss fiber delay line to store packets during network delays. We also find that implementing cut-off storage times in a strategy analogous to real-time selection in free-space QKD can significantly enhance performance. Our work demonstrates that packet switching is imminently suitable as a platform for QKD, an important step towards developing large-scale and integrated quantum networks.

npj Quantum Information (2023)9:85 ; https://doi.org/10.1038/s41534-023-00757-x

INTRODUCTION

Packet-switched communication networks were introduced as an efficient and scalable alternative to circuit switching in the early sixties[1,2]. Today, packet switching is the dominant mode of operation in the Internet. Recently we have introduced packet switching as a paradigm for quantum networks using hybrid (classical-quantum) data frames[3]. Inside a frame, a quantum payload is prepended with a classical header containing information for routing and more. Frames travel from sender to receiver through a series of routers, which process the header to determine the channel forward based on the current conditions of the network (Fig. 1a). This is in contrast to a circuit-switched network where a dedicated channel is established between sender and receiver and reserved until communication is complete (Fig. 1b).

There are important considerations to be made when deciding whether packet switching or circuit switching is best suited for a network application. In a circuit-switched network, communication across multiple user pairs must be done in a coordinated fashion in order to enable bandwidth sharing (e.g., via time or wavelength-division multiplexing). In a packet-switched network, the communication need not be coordinated in advance. However, frames will experience delays at the intermediate nodes between users due to finite header processing times and, under some traffic conditions, queuing times. For this reason, packet switching is generally advantageous over circuit switching when the traffic generated by network users is *bursty*, characterized by intervals of activity and intervals of inactivity.

One important application in a quantum network is quantum key distribution (QKD), a procedure that allows two remote users (e.g., Alice and Bob) to establish shared encryption keys with information-theoretic security[4,5]. An important feature of QKD is that it is robust against loss in transmission, meaning that a secure key can still be established even when most of the transmitted signals are lost. This suggests that data loss due to delays in a packet-switched network may be tolerated even without any storage of QKD signals at the routers. Moreover, the optical loss introduced by an imperfect storage medium may also be tolerated. Another important feature of QKD is that key generation is not time-critical, meaning that secure keys need not be generated immediately before their consumption. This implies that bursty frame generation may be sufficient since users can establish and store keys for later use.

These features motivate our hypothesis that packet switching is imminently suitable as a platform for QKD. One may of course imagine a scenario where network users prefer access to a dedicated quantum channel for their key distribution (e.g., urgent requests or large size requirement for encryption keys). Furthermore, most existing demonstrations of multi-user QKD are conducted over dedicated networks[6–12] where QKD is the sole task. In this case, it may be beneficial to have a central controller to coordinate QKD among different user pairs, in a fashion similar to circuit switching. However, if we wish to integrate QKD with existing classical networks in order to extend its applications, packet switching is a natural choice.

Packet switching in quantum networks is a relatively unexplored topic, but has been proposed as a solution for overcoming scalability issues in previous works[13,14]. Moreover, ref. [15] has investigated using leading classical signals to make routing decisions in a QKD network, although packet switching is not considered in their approach. In our previous work[3], we presented a proof-of-concept for QKD in a packet-switched tandem network, and considered a basic model for a two-user scenario where the routers had no optical storage capacity. While this work captured certain features of packet switching by introducing dynamic switches between the sender and receiver, it did not consider any non-trivial aspects of a packet-switched network; namely, the network delays and contentions when many users are present, the use of storage at the intermediate nodes, and the use of varying parameters for generating and sending the hybrid frames. Furthermore, the QKD protocol considered in our previous work was single-photon BB84 in the asymptotic regime. In this work, we analyze a sixteen-user network with and without optical storage capacity at the routers, and consider a finite-size security analysis for decoy-state BB84.

The goal of this paper is to demonstrate the feasibility of performing QKD in a packet-switched network. To meet this goal, we take a three-step approach. First, we choose a network routing protocol that describes how a router handles a frame during

[1]Cisco Quantum Lab, Los Angeles, California, USA. [2]University of Toronto, Toronto, Ontario, Canada. [3]Cisco Quantum Lab, Garching bei München, Bavaria, Germany.
✉email: sdiadamo@cisco.com

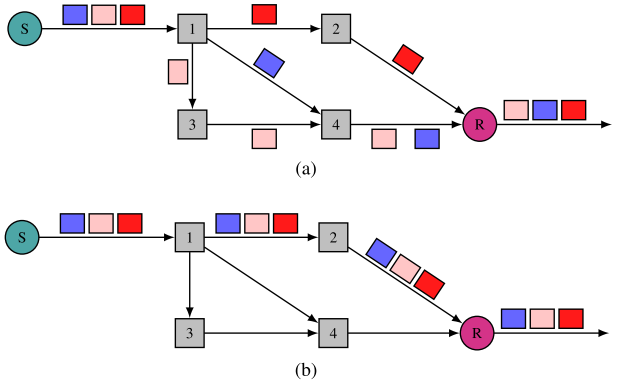


Fig. 1 Difference between packet switching and circuit switching. **a** Packet-switched network. The channel between sender (S) and receiver (R) is not predetermined and can be dynamically reconfigured. **b** Circuit-switched network. A dedicated channel between sender and receiver is set up before data is transferred between them.

network delays. In this paper, we will investigate three different routing protocols based on varying optical storage capacity. Second, we simulate the transport of frames in a network operating under a given routing protocol and traffic model. The simulation provides us with statistics for the dynamic channel between each Alice-Bob pair. Lastly, we use the simulated network statistics to predict the maximum secure key rate for each user pair in the network by performing a finite-key analysis. Our results show that QKD is feasible in a packet-switched network with today's commercial technology and that optical storage can be used to improve its performance under certain conditions.

This paper is organized as follows. In section "Network delays and routing protocols", we review the frame structure then outline the delays in a packet-switched network and the routing strategies considered in this work. In section "Router hardware", we propose a router hardware design based on current technology. Simulated key rates for a sixteen-user network are presented in section "Simulation results". We discuss considerations for future work and conclude in section "Discussion".

RESULTS

Network delays and routing protocols

Figure 2 depicts a hybrid frame with a quantum payload consisting of weak laser pulses with repetition rate R_t (Hz). The classical trailer is used to indicate the end of the payload. The frame may be configured to include a time delay between the end of the header and the beginning of the payload, referred to as the guard time, T_g. The total time a frame needs to move through a router is the sum of three sources of delay. First, there is the processing delay, d_{proc}, which is the time to process the classical header and determine the next action for the frame as well as regenerate the header when needed. Depending on the network complexity, this delay can range from 10 to 1000 μs[16]. In this work, we assume a d_{proc} of 100 μs. Second, there is the queuing delay, d_{queue}, which is the time the frame must wait before it can be forwarded from a router (after the header has been processed). This quantity depends on the traffic conditions of the network and can range from zero to infinity. Lastly, there is the transmission delay, d_{trans}, which is the time required to transmit the entire frame onto an outgoing link. This is equal to the temporal frame length, T_f, which may shrink at each router it traverses depending on the routing protocol employed.

The network routing protocol specifies what happens to a frame during the network delays d^i_{proc} and d^i_{queue}, where the superscript *i* is used to index each router in the frame's path from sender to receiver. In general, our network routing protocols fall into one of two categories based on the capacity to store frames at the routers. For protocols without storage, d^i_{trans} ($= T^i_f$) will shrink by a duration equal to $d^i_{proc} + d^i_{queue}$ at each router the frame traverses. If $T^i_g = 0$, this corresponds to the discarding of $R_t(d^i_{proc} + d^i_{queue})$ pulses in the leading portion of the payload (note that we consider the lengths of the classical header and trailer to be negligible compared to the quantum payload). If T^i_g>0, then it will serve as a buffer to reduce the number of pulses that are lost (i.e., if T^i_g>$d^i_{proc} + d^i_{queue}$, then no pulses are discarded as the frame shrinks but T^i_g will decrease accordingly). Note that in each routing protocol we consider, the guard time is not reset at each router. This alternative approach may be useful for a quantum network application in which the payload carries information that should not be lost.

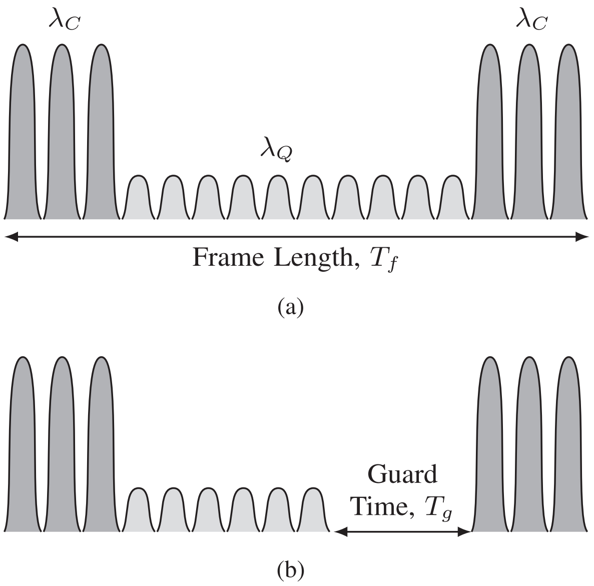


Fig. 2 Hybrid frame structure. **a** The classical header and trailer (λ_C) and the quantum payload (λ_Q) are generated from a laser source and multiplexed into a hybrid data frame using time-division and wavelength-division multiplexing (not shown to scale). **b** The hybrid frame includes guard time—a time delay between the end of the header and the beginning of the payload.

For protocols with storage, the frame will enter a fiber delay line for a storage time $T^i_s \leq d^i_{proc} + d^i_{queue}$. During T^i_s, no pulses are discarded from the payload, but they will be subject to the attenuation of the fiber delay line. If T^i_g>0, then it will again serve as a buffer to reduce T^i_s (i.e., if T^i_g>$d^i_{proc} + d^i_{queue}$, then $T^i_s = 0$ but T^i_g will decrease accordingly). Note that the header may be configured to include a field that tracks the cumulative time spent in storage as a frame traverses the network. In this work, we investigate the following three routing protocols.

1. *No storage during delays*. At each router, a frame will have its payload discarded for a time $d^i_{proc} + d^i_{queue}$ and d^i_{trans} will shrink by the same amount. If d^i_{trans} reaches zero, then the frame is discarded from the network.
2. *Storage during delays (unlimited)*. At each router, a frame will enter a fiber delay line for a storage time $T^i_s = \max(0, d^i_{proc} + d^i_{queue} - T^i_g)$ and d^i_{trans} will shrink by $\min(T^i_g, d^i_{proc} + d^i_{queue})$.
3. *Storage during delays (limited)*. At each router, a frame will enter a fiber delay line for a storage time $T^i_s = \max(0, d^i_{proc} + d^i_{queue} - T^i_g)$ and d^i_{trans} will shrink by

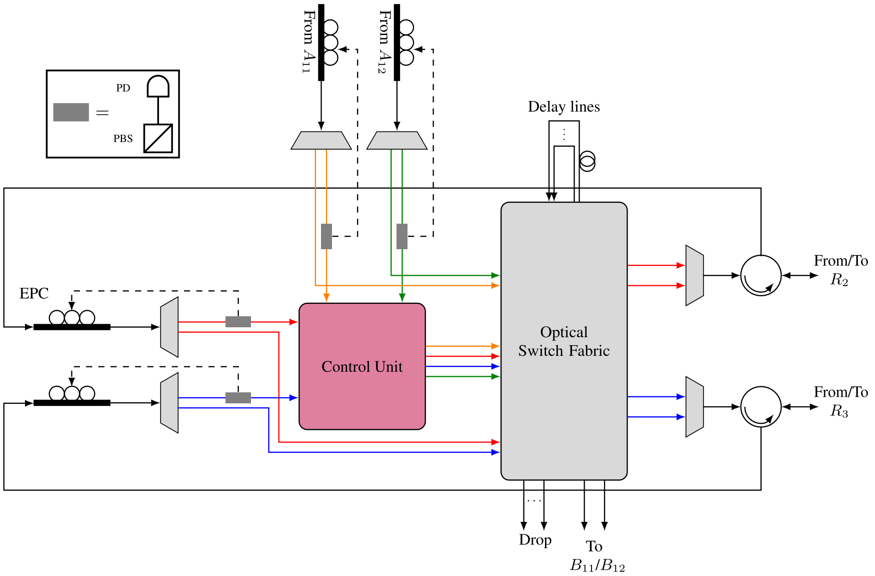


Fig. 3 Inside the routers. Hardware design of router R_1 in the packet-switched network depicted in Fig. 4. A frame arrives at the router from senders A_{11}, A_{12} and from routers R_2, R_3. An arriving frame passes through a wavelength-division multiplexer to separate the classical and quantum information. The classical information is processed in the control unit, which signals to the optical switch fabric where to route the frame (i.e., to a router R_2, R_3 or to a receiver B_{11}, B_{12}) and regenerates the header prior to transmitting the frame. A circulator is used to allow for bidirectional transmission in the channels linking to routers.The optical switch may also access variable optical fiber delay lines for storing frames. Drop channels are used for discarding pulses or entire frames. The polarization drift of the classical signals is measured and used to compensate the drift of the quantum signals. PD photodiode, PBS polarizing beam splitter, EPC electronic polarization controller.

$\min(T^i_g, d^i_{proc} + d^i_{queue})$. If the total time a frame has spent in storage reaches a predetermined storage time limit, the frame is immediately discarded from the network.

In the no storage routing protocol, network delays introduce a controlled photon loss as a portion of the payload is discarded. In the storage routing protocols, network delays introduce random photon loss in the payload due to the attenuation of the fiber delay line. The regime in which one strategy may dominate over the other therefore depends on factors such as the frame length, the network delays, and the attenuation of the storage line. A more detailed motivation for the two types of routing protocols is provided in Supplementary Note 1.

To motivate the limited storage routing protocol, we make the observation that the dynamic channel conditions in a packet-switched network are analogous to those in free-space QKD under turbulent conditions. In such scenarios, it has been shown that the key rate can be improved by rejecting key bits when the channel's transmittance is below a threshold[17–19]. In our case, since the routing history is recorded in the classical header, we can discard frames en-route, which has the additional benefit of reducing network congestion. Another option, more analogous to the technique used in free-space QKD, is to allow all frames to reach the receiver end via the unlimited storage routing protocol, but enforce a storage time limit (STL) in post-processing. That is, frames for which $\sum_i T^i_s > STL$ will be excluded from key generation. In this work, we compare both options for implementing a cut-off channel transmittance.

Router Hardware

A conceptual router design is shown in Fig. 3. This router behaves as a quantum version of a reconfigurable optical add drop multiplexer (ROADM). A ROADM is a routing device that can accept incoming frames from multiple links (senders, other routers), direct outgoing frames to multiple links (receivers, other routers), discard frames from the network using fibers labeled as 'Drop' channels, and move frames into and out of fiber delay lines used as storage. The links to adjacent routers contain circulators to support bidirectional transmission over the same fibers. Frames arriving at a router will pass a wavelength-division multiplexer that is used to separate the quantum payload from the classical header and trailer. The header is then fed into a control unit to decide how to further process the frame. Once the header has been processed, the frame will be forwarded towards the next node in the network (i.e., to another router or to a receiver). The control unit will regenerate the header with updated fields for the quantum payload duration, guard time, and time spent in storage prior to transmitting the frame to the next node.

We assume the control unit is capable of processing up to k headers simultaneously and that the router has access to q variable optical fiber delay lines. To achieve an arbitrary delay, each fiber delay line can be combined with an active optical switch (not illustrated in figure). The router can also discard frames or partially discard the quantum payload via its Drop channels. The use of these channels depends on the network routing protocol being implemented.

We also assume the router to have a minimum insertion loss of 4 dB, which accounts for the circulators, multiplexers, and optical switch fabric (excludes the fiber delay lines). This value is a conservative estimate based on the current performance of commercial devices. As the performance of these components improves in the future, we also expect better QKD performance. The total loss (dB) at each router is thus given by

$$loss^i_r = T^i_s v_g \alpha_s + 4dB, \quad (1)$$

where v_g is the speed of light in fiber and α_s is the attenuation coefficient (dB/km) of the fiber storage line. Furthermore, we assume the router may compensate the polarization drift of all incoming channels by using a feedback signal generated from the measured drift of the classical pulses in the header.

Lastly, we note that this router design is directly suitable for the network configuration in Fig. 4 although additional links may be added to the router depending on the desired connectivity of the network. We also consider hardware that is directly suitable for the hybrid frame in Fig. 2 although the hardware can be modified according to the multiplexing scheme employed for the frame.

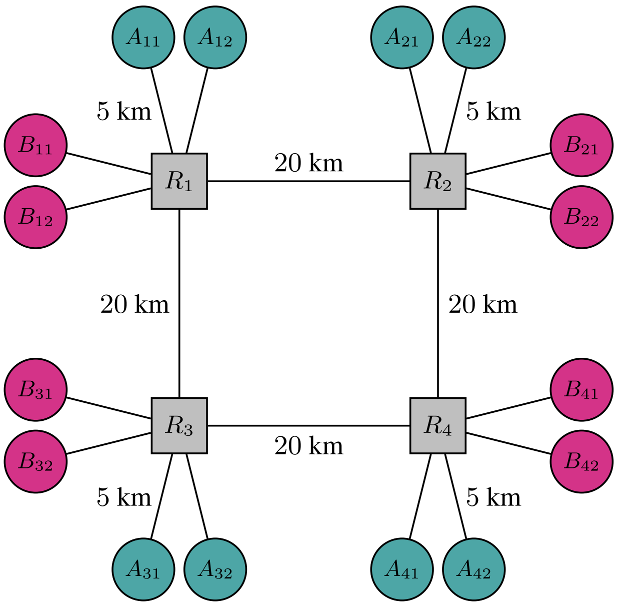


Fig. 4 Network topology. Sixteen-user network for simulation. Each of the four routers are connected to two Alices and two Bobs. The links are assumed to be standard single-mode optical fiber (0.2 dB/km) spanning 20 km between routers and 5 km between each user and their default router.

Simulation results

In order to demonstrate the feasibility of performing QKD in a packet-switched network, we analyze the network shown in Fig. 4. We choose this topology as it combines properties of star, ring, and dumbbell networks. We emphasize, however, that our approach may be used to test an arbitrary network configuration. In our simulated network, sixteen users are connected through four routers by standard single-mode fiber. In practice each user can operate as a sender or a receiver, but we assume that users do not operate in both modes simultaneously. Thus, half of the users are designated as senders ("Alices") and half as receivers ("Bobs"). In this section, we present the secure key rates in bits per second for Alice-Bob pairs separated by one, two, and three routers. We test each of the three routing protocols outlined in section "Network delays and routing protocols". Details on the QKD security analysis and network simulation are provided in section "Methods".

In Fig. 5, we show the key rate performance in a network with no storage during delays. We fix the number of frames sent between each user pair and examine the effects of the average frame inter-arrival time $1/\gamma$, the initial frame length T_f^0, and the initial guard time T_g^0. In this routing protocol, these parameters affect the data size, N, for key generation. The top and subsequent rows contain the results for zero and non-zero guard times, respectively. The columns from left to right show the results for a user pair separated by one, two, and three routers. Note that since there are random idle times between packets to simulate bursty network traffic, in the key rate we consider only the time during which Alice is transmitting frames. That is, to obtain the key rate in bits per second, we multiply the key rate in bits per sent pulse by $(1 - T_g^0/T_f^0)R_t$, where $R_t = 1$ GHz. We discuss other formats for this figure of merit in section "Discussion".

We interpret these results as follows. Firstly, the secure key rate is expected to decrease with higher channel loss. Therefore, we observe the highest key rates for A_{31} and B_{32} and the lowest for A_{22} and B_{31}. We note that due to the symmetry of the network configuration, there are negligible differences between the results of different user pairs with the same separation. For small values of $1/\gamma$, higher network traffic results in larger d_{queue} leading to more pulses being discarded and thus smaller N. As a result, we observe a decrease in the key rate as $1/\gamma$ decreases. In Fig. 5a–c, we observe the effect of T_f^0. As this parameter increases, more pulses are generated. However, longer frames have a larger d_{trans} which increases the time for which the server is occupied at each router and therefore increases d_{queue}. Thus we expect the upwards trend in the key rate to eventually stop, as is observed in Fig. 5c. In Fig. 5d–f, we observe the effect of T_g^0 for a fixed T_f^0. A larger guard time means fewer pulses are discarded during delays but smaller payloads are generated. Due to this effect, a non-zero guard time is shown to slightly enhance the key rate, albeit only under high network traffic (small $1/\gamma$). Ultimately, these results suggest that QKD is feasible in a packet-switched network even without any optical storage capacity at the routers.

In Fig. 6, we show the key rate performance in a network with storage during delays, where frames have no storage time limit. We fix the number of frames sent between each user pair and examine the effect of the attenuation coefficient, α_s, for the fiber delay lines used as storage at the routers which will determine $\langle\eta_{tot}\rangle$ for the QKD channel. The top and bottom rows consider scenarios of long and short frame lengths, respectively, where the ratio of frame length to $1/\gamma$ is fixed in each such that the average network traffic is the same. The left and right columns consider zero and non-zero guard times, respectively. For each user pair, we compare the results of this routing protocol to the no storage routing protocol under the same network parameters.

We interpret these results as follows. Firstly, the secure key rate decreases exponentially with α_s, as expected. A non-zero guard time is again shown to enhance the key rate since it reduces the storage time of each payload, which increases $\langle\eta_{tot}\rangle$. Guard time also reduces d_{queue} since it shrinks d_{trans} at each router. The enhancement is more pronounced in the long frames scenario since the guard time is $\gg d_{proc}$ in this case. We observe that the short frames scenario is generally more robust to increasing α_s, which can be attributed to smaller storage times due to a smaller d_{trans}. The distributions of the storage time in the long and short frames scenarios are shown in Fig. 8 for the case of zero guard time. In Fig. 6a, b, we observe that the no storage routing protocol is generally superior when $\alpha_s > 0.01$ dB/km. We note that while attenuation coefficients as low as 0.14 dB/km have been achieved over telecom wavelengths using state-of-the-art technology[20], it is unrealistic to consider an attenuation much smaller than this. For a more efficient storage medium, we require long-lived quantum memories. In Fig. 6c, d, we do not extract any secure keys with the no storage routing protocol except in the case of one router separating users. This can be explained since the frame length is on the order of d_{proc}, so there are zero to few non-discarded pulses from each payload. Our results suggest that, for short frames, storage during network delays is a better strategy than discarding pulses. The opposite holds true for frame lengths $\gg d_{proc}$ when we consider realistic fibers as our storage medium. This finding is important since frame lengths in a packet-switched network may have practical constraints.

As mentioned previously, we may enforce a STL in post-processing, analogous to applying a cut-off $\langle\eta_{tot}\rangle$, in order to improve the key rate. Figure 7 shows the results for the same parameters as in Fig. 6, but with frames excluded from key generation if their storage time reached the STL. We consider an ultra-low-loss fiber with $\alpha_s = 0.16$ dB/km as our storage medium and examine the effect of the STL duration. It is clear that implementing a STL enhances the key rate in each scenario considered, and most significantly for frame lengths $\gg d_{proc}$. In Fig. 7a, the optimal STL for users separated by one, two, and three routers is 200, 300, and 400 μs, respectively. From Fig. 8a, we see that these STLs preserve 82, 70, and 58% of frames across the user pairs. In Fig. 7b, the optimal STL is roughly 150 μs for all user pairs

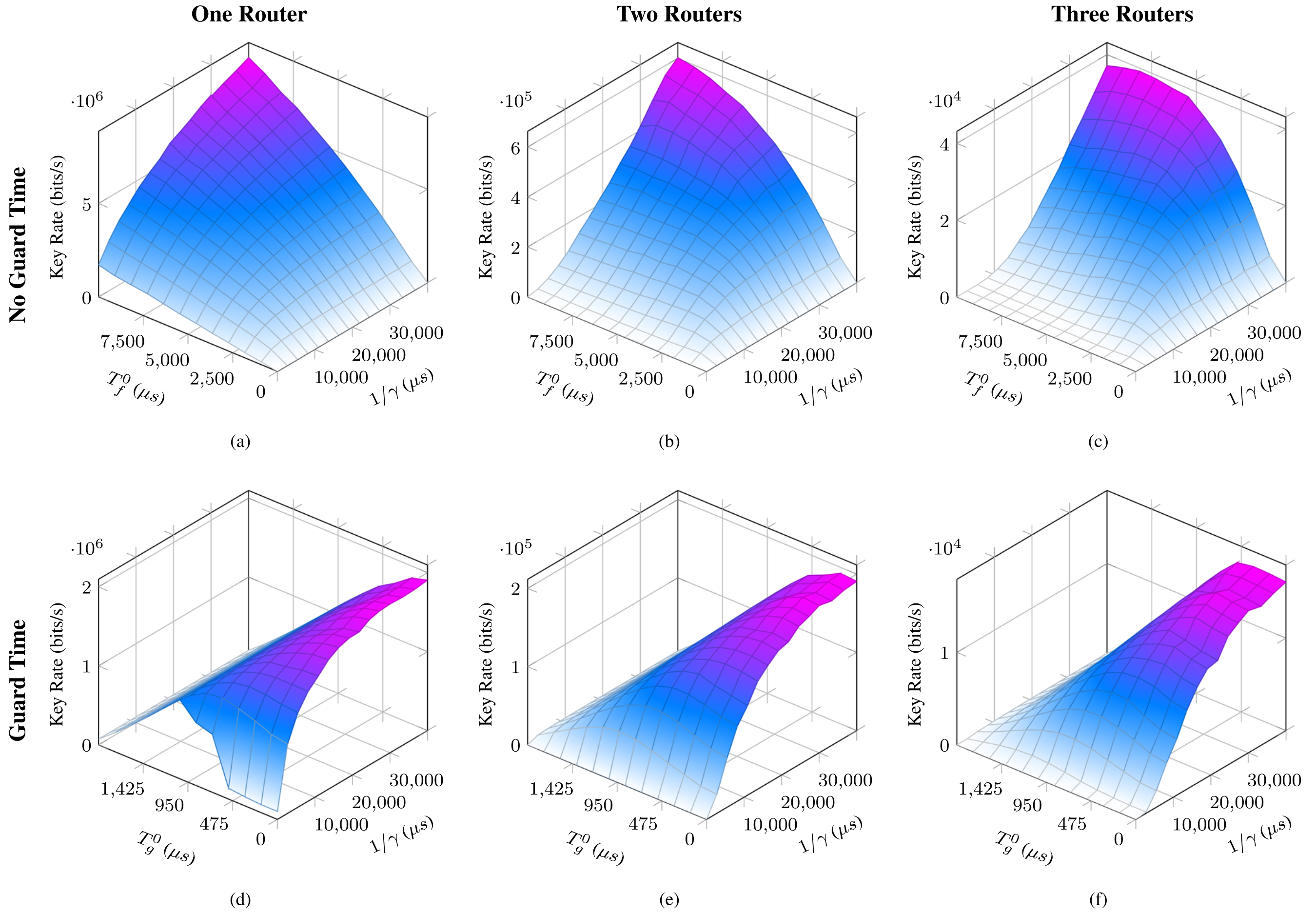


Fig. 5 **Secure key rates in a network with no storage during delays.** A total of 18,750 frames are generated by Alice in each user pair. The finite-size is $N \approx 10^{12}$. In plots (**a**)–(**c**), we fix the guard time to be zero and vary the initial frame length and average frame inter-arrival time, $1/\gamma$. In plots (**d**)–(**f**), we fix the initial frame length to 2000 μs and vary the initial guard time and $1/\gamma$. Columns (left to right) are for user pairs A_{31} and B_{32}, A_{42} and B_{22}, and A_{22} and B_{31}. Color map changes from white to purple as the key rate increases.

and the key rates approach those of the no storage routing protocol.

In Fig. 9, we show the key rate performance in a network with storage during delays, where frames have a storage time limit. Once again, we fix the number of frames sent between each user pair and consider $\alpha_s = 0.16$ dB/km. We examine the effect of the STL duration under various network parameters and in each case we compare the results with the unlimited storage routing protocol where a STL is implemented in post-processing. Note that for the network parameters in the previous subsection, the two methods for implementing a cut-off transmittance produce very similar results. Here we show scenarios in which discarding frames en-route provides a significant advantage due to its mitigation of network congestion.

DISCUSSION

In this work, we have developed a framework for key rate optimization in a packet-switched network and assessed QKD performance in relation to several network parameters such as frame length, guard time, frame inter-arrival time, and storage efficiency. Notably, we found that practical secure key rates can be achieved without any optical storage capacity in the network and that guard time can generally be used to mitigate the effects of network delays. We also found that the transmittance threshold strategy used in free-space QKD can be applied in a packet-switched network to significantly enhance the key rate by limiting the permissible storage time of frames. We believe our results pave the way for future exploration of quantum applications in a packet-switched network.

The figure of merit we have used for QKD in a packet-switched network is the key rate per unit time that Alice is transmitting frames for this task. Recall that one motivation of packet switching is to allow network users to perform multiple tasks, so during the idle time of QKD, Alice may do something else. Thus, it is reasonable for the key rate to consider only the time Alice actually spends on QKD. Depending on the desired applications of the network, it may be useful to investigate the number of secret keys that can be generated over a given time window. Results for the key rate in bits per pulse sent by Alice are provided in Supplementary Note 2 and exhibit very similar trends.

Regarding the use of hybrid frames in the network, an important consideration to be made is that nonlinear effects in fiber may greatly reduce the signal-to-noise ratio of QKD signals when they are co-propagating with the bright classical signals of the header and trailer[21]. These effects can be mitigated via careful wavelength assignments for the classical and quantum signals, and applying spectral and temporal filtering. Indeed, such strategies have been used to successfully integrate QKD with high-power classical communications over metropolitan-scale distances[22–24]. Improved versions of our present work may explicitly account for cross-talk in the noise model for QKD.

In the future, it may be interesting to consider the application of measurement-device-independent (MDI) QKD[25] in a packet-

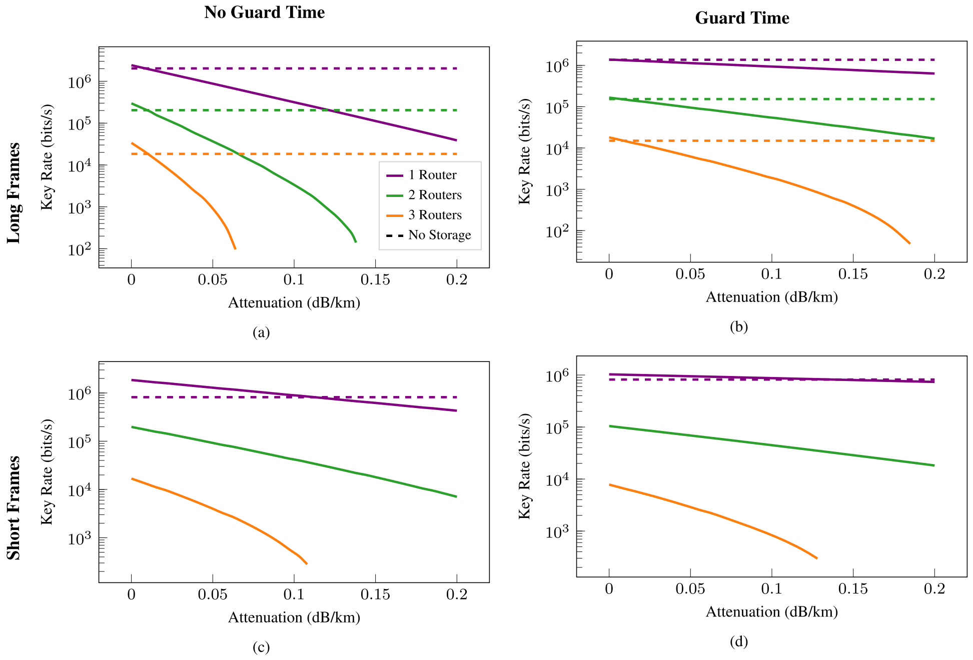


Fig. 6 Secure key rates in a network with storage during delays (unlimited). In each plot, we fix the network parameters and vary the attenuation of the fiber storage lines. A total of 37,500 frames are generated by Alice in each user pair. The finite data size is $N \approx 10^{12}$. Unless displayed, the no storage routing protocol fails to produce a secure key. **a** $T_g^0 = 0$, $1/\gamma = 30{,}000\,\mu s$, $T_f^0 = 2000\,\mu s$. **b** $T_g^0 = 800\,\mu s$, $1/\gamma = 30{,}000\,\mu s$, $T_f^0 = 2000\,\mu s$. **c** $T_g^0 = 0$, $1/\gamma = 3000\,\mu s$, $T_f^0 = 200\,\mu s$. **d** $T_g^0 = 80\,\mu s$, $1/\gamma = 3000\,\mu s$, $T_f^0 = 200\,\mu s$.

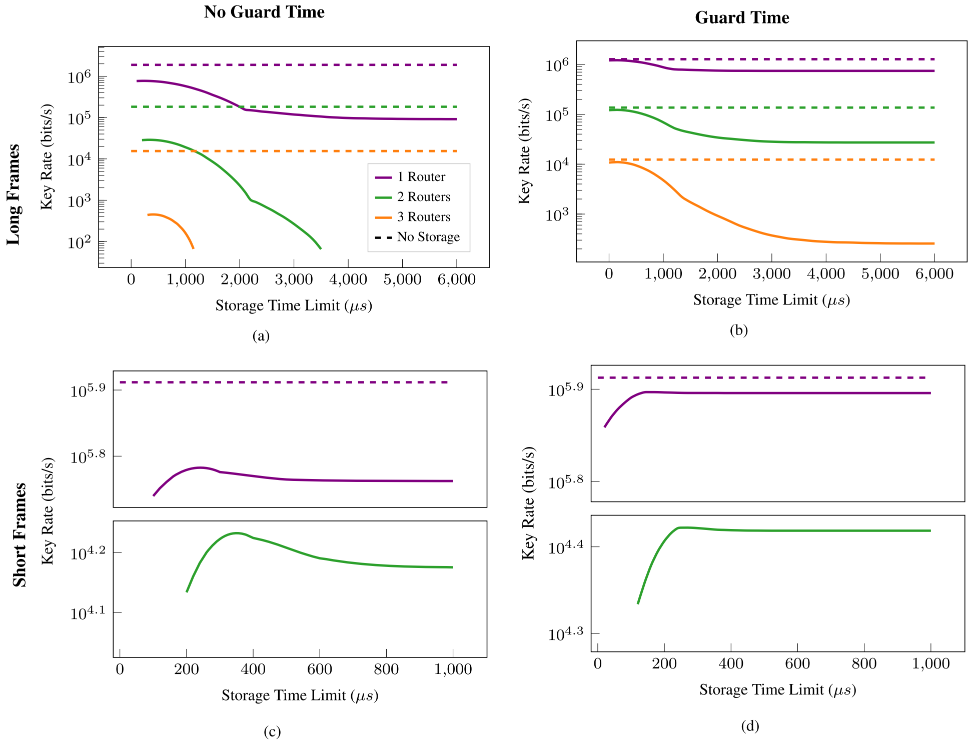
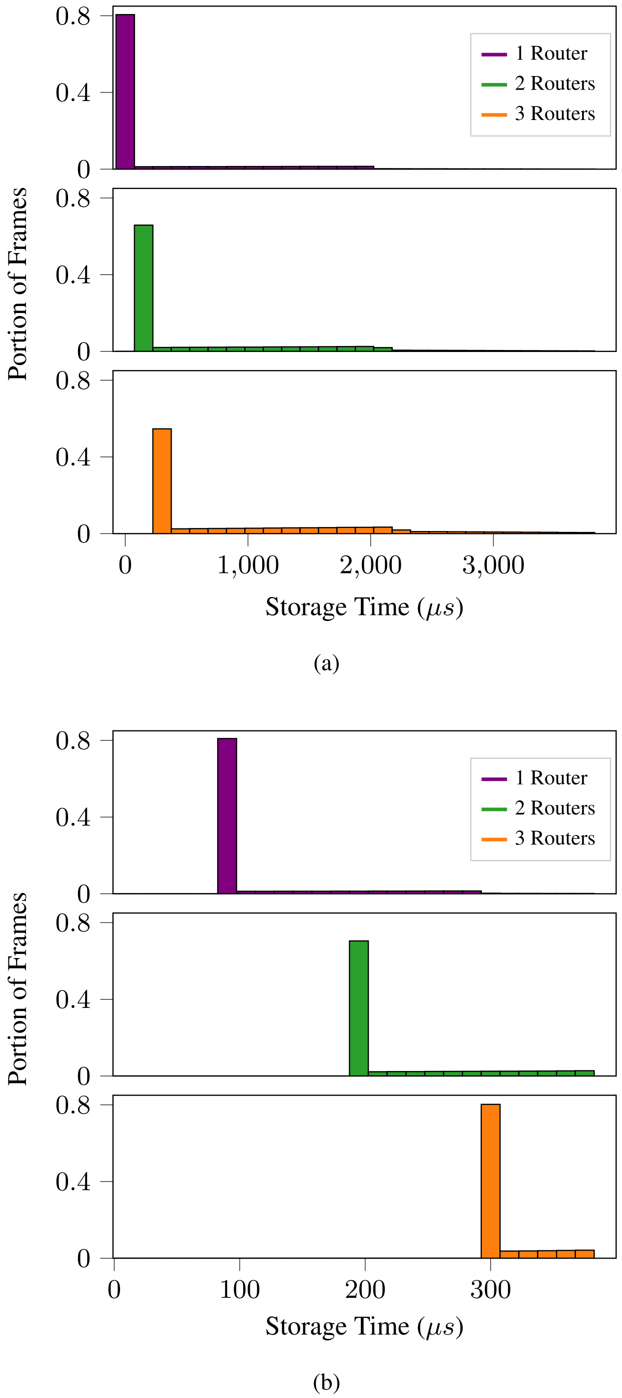


Fig. 7 Secure key rates in a network with storage during delays (unlimited) and STL implemented in post-processing. In each plot, we fix the network parameters and vary the STL duration. The attenuation of the fiber storage lines is fixed at 0.16 dB/km. The network parameters for panels (**a** $T_g^0 = 0$, $1/\gamma = 30{,}000\,\mu s$, $T_f^0 = 2000\,\mu s$. **b** $T_g^0 = 800\,\mu s$, $1/\gamma = 30{,}000\,\mu s$, $T_f^0 = 2000\,\mu s$. **c** $T_g^0 = 0$, $1/\gamma = 3000\,\mu s$, $T_f^0 = 200\,\mu s$. **d** $T_g^0 = 80\,\mu s$, $1/\gamma = 3000\,\mu s$, $T_f^0 = 200\,\mu s$) are identical to the respective panels in Fig. 6.



(a)

(b)

Fig. 8 Distribution of storage times in a network with storage during delays (unlimited). The *y*-axis denotes the fraction out of all frames traversing the indicated number of routers. $T_g^0 = 0$. **a** $1/\gamma = 30{,}000\,\mu\text{s}$, $T_f^0 = 2000\,\mu s$. **b** $1/\gamma = 3000\,\mu\text{s}$, $T_f^0 = 200\,\mu s$.

switched network. Due to its invulnerability to detector side-channel attacks and natural compatibility with a star network topology, MDI-QKD is an appealing choice for secure communication networks[11,26]. However, this protocol requires that signals from Alice and Bob arrive simultaneously at an intermediate node in order to exhibit Hong-Ou-Mandel interference, posing a significant challenge under packet switching since Alice's and Bob's channels are subject to independent dynamic routing conditions. If the intermediate node has access to reliable quantum memories, then memory-enhanced MDI-QKD[27] may be used to overcome this challenge. Nonetheless, current state-of-the-art commercial QKD devices operate under the decoy-state BB84 protocol, making it an important application for packet-switched networks in the near-term.

Other future areas of investigation may include examining more complex network topologies and perhaps a topology deployed in the field. Given that our simulation tool can accommodate arbitrary network configurations, hardware specifications, and traffic models, it can be used to establish a performance benchmark for real-world systems. The simulation tool, which we aim to make publicly available in the near future, can also be extended to examine the performance of other quantum communication tasks besides QKD such as entanglement distribution.

Lastly, an interesting question to address is how QKD in a packet-switched network compares to a circuit-switched network. While we have a general idea of when packet switching outperforms circuit switching based on classical networks, determining specific conditions for this advantage in a quantum network may be useful.

METHODS

QKD security analysis

Practical implementations of QKD adopt the decoy-state method[28–31] to allow for use of a weak pulsed laser source instead of an ideal single-photon source. In this work, we consider a decoy-state asymmetric coding BB84 protocol[32] and we adopt the finite-size security analysis in ref. [33] to calculate the secure key rate. In this section, we provide a brief summary of the QKD protocol and then describe our strategy for key rate optimization in a packet-switched network.

Preparation. Alice chooses a bit value b_A uniformly at random. Then, she selects a basis $\in \{X, Z\}$ with probabilities q_x and $1 - q_x$, respectively, and an intensity $k_i \in K := \{\mu_1, \mu_2, \mu_3\}$ with probabilities p_{μ_1}, p_{μ_2}, and $p_{\mu_3} = 1 - p_{\mu_1} - p_{\mu_2}$, respectively. Lastly, she prepares a weak laser pulse based on the chosen values and sends it to Bob.

Measurement. Bob selects a basis $\in \{X, Z\}$ with probabilities q_x and $1 - q_x$, respectively. Then, he performs a measurement in the chosen basis and records the outcome in a bit value b_B. The measurement device typically employs two single-photon detectors. More precisely, Bob assigns $b_B = 0$ for a click in single-photon detector D_0 and $b_B = 1$ for a click in detector D_1. If both detectors click, he assigns a random value to b_B. If neither detector clicks, he does not assign any value.

Basis reconciliation. Alice and Bob announce their basis and intensity choices over an authenticated public channel. Based on the information announced, Alice and Bob identify their raw keys b_A and b_B from the instances where they both chose basis X and Bob observed a detection event. Note that all intensity levels are used for the key generation[33]. They use the instances where they both chose basis Z and Bob observed a detection event for phase error estimation.

Post-processing. Alice and Bob perform classical error correction and privacy amplification on their raw key pair to extract a secure key.

A convenient feature of standard security proofs for BB84, including the one adopted in this work, is that it is assumed that an adversary may have full control over which signals arrive at Bob and which do not, without affecting the security of the protocol. This assumption is secure since the adversary cannot distinguish between the four BB84 states and therefore gains no advantage by blocking certain signals from arriving at Bob. Since Bob post-

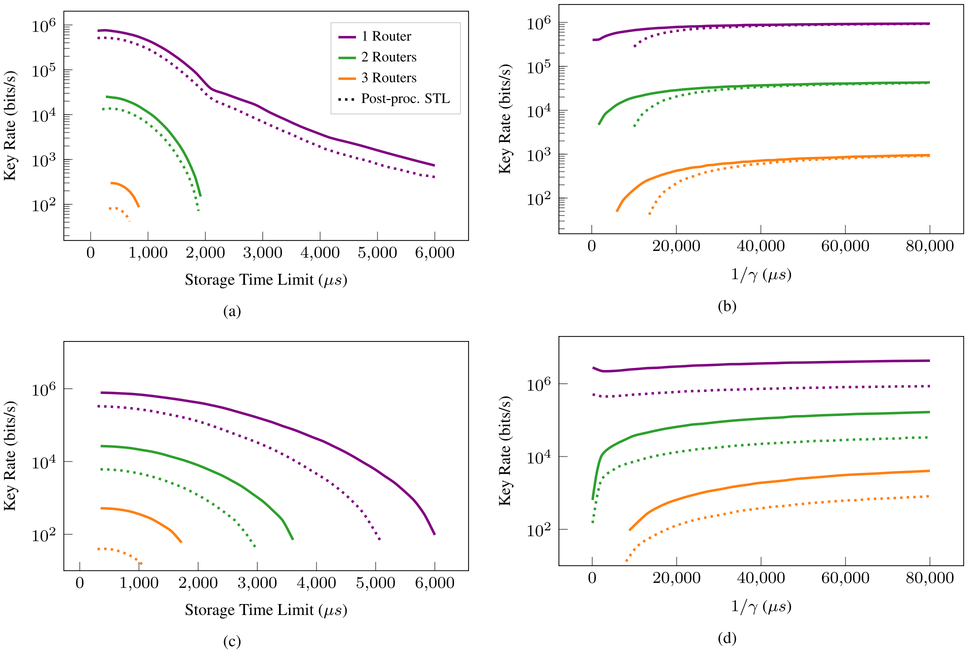


Fig. 9 Secure key rates in a network with storage during delays (limited). The attenuation of the fiber storage lines is fixed at 0.16 dB/km. A total of 37,500 frames are generated by Alice in each user pair. $T_g^0 = 0$. **a** $1/\gamma = 15{,}000\ \mu s$, $T_f^0 = 2000\,\mu s$. **b** STL $= 320\ \mu s$, $T_f^0 = 2000\,\mu s$. **c** $1/\gamma = 50{,}000\ \mu s$, $T_f^0 = 10{,}000\,\mu s$. **d** STL $= 550\ \mu s$, $T_f^0 = 10{,}000\,\mu s$.

selects on the events where he observed a detection, this assumption amounts to the adversary having full control over Bob's post-selection process. In the packet switching scenario, the channel consists of routers where signals may be discarded for legitimate reasons, and the routers communicate to Bob (via the header) which signals have been discarded. This is equivalent to the routers (which may very well be under the control of an adversary) steering Bob's post-selection. As explained, this falls within the assumptions for the security proof of BB84. Thus, we need not trust the routers or develop a new security proof for QKD in a packet-switched network.

Nonetheless, packet switching poses a unique challenge to QKD due to the dynamic nature of the quantum channel between users. In order to maximize the secure key rate in the decoy-state protocol described above, we must optimize over the free parameters $\{q_x, p_{\mu_1}, p_{\mu_2}, \mu_1, \mu_2\}$[33] which requires knowledge of the average channel transmittance, $\langle\eta_{tot}\rangle$, where the average is taken over all frames contributing to the key. Furthermore, in order to conduct a finite-size analysis, we must determine the total number of QKD states, N, passed to Bob. Depending on the network routing protocol employed, this may not be equivalent to the number of states transmitted by Alice, N_0, due to discarding at the routers. Therefore, in order to predict the maximum secure key rates from QKD in a packet-switched network, we need a tool for assessing $\langle\eta_{tot}\rangle$ and N for each user pair. One may consider an analytic approach to gathering these statistics, however this quickly becomes infeasible for increased complexity of the network. The theory of Jackson networks[34] allows us to calculate the average queuing delay at each router quite simply, but only if the network obeys a specific traffic model. Instead, we build a network simulation tool to numerically determine the channel statistics. Details of the key rate analysis, including noise and detection parameters, are given in the Supplementary Methods.

Network simulation

In this section, we first provide a high-level description for the sequence of events that occur as a frame travels from sender to receiver in a packet-switched network and then describe our software tool for simulating these events in order to extract the dynamic channel statistics.

We model the arrival of frames into the network as follows. Each sender is allowed to transmit frames one at a time, following an exponentially distributed inter-arrival time with an average $1/\gamma$. Note that all senders can be active simultaneously. We assume a repetition rate $R_t = 1$ GHz for the signals in the quantum payload. The destination for each frame is assigned randomly from the list of all receivers in the network.

A frame travels from a sender towards its default router (i.e., the router to which the sender is directly connected). The default router and all subsequent routers a frame encounters will forward the frame according to the path determined by the routing algorithm for the network. The routing algorithm calculates the least-cost path from sender to receiver, where the cost of a path is the sum of the link costs along the path. In this work, we consider a load-insensitive routing algorithm, meaning the cost of each link in the network does not reflect its level of congestion and is determined solely by its physical length. Therefore, the least-cost path is simply the shortest path. Note that in the case of multiple least-cost paths, the router will select one at random. In general, the shortest path may not have the highest expected transmittance, depending on the number of routers it contains. In this case, the cost of the path may be modified to include router loss, although this scenario is not applicable in this work.

A frame can be forwarded from a router only if there are fewer than c frames simultaneously being forwarded from the router and there are no frames preceding it in the queue (we refer to c as the number of servers for the queue); otherwise, the frame must join the queue. A frame may join the queue only if there are fewer than q frames already in the queue (we refer to q as the capacity of the queue); otherwise, the frame will be discarded. Frames will be forwarded from the queue according to a first-come first-served discipline.

In order to simulate these events in a network, we developed a software tool based on a simulation method known as discrete-event simulation (DES)[35]. We build on the DES Python package *SimPy*[36] for the timing and resource management aspects of the network. For the network configuration, including path calculations and topology initialization, we use the Python package *NetworkX*[37].

The first step in using our simulation is to configure a topology of nodes (i.e., users and routers) and links (i.e., connections between nodes). Each node is able to generate frames as well as process any incoming frames. If the node is a sender, frames at the node do not undergo header processing and the frame need only wait to be sent into the network according to the frame arrival model. If the node is a router or a receiver, frames at the node will undergo a processing delay. In our simulation, routers can process $k \gg 1$ headers simultaneously. In general, if k is small, the frames may experience a queuing delay prior to header processing. In our simulation, the queue in each router has $c = 1$ server and unlimited storage capacity ($q \to \infty$). The actions on the frame during the processing and queuing delays will depend on the network routing protocol, as outlined in the "Results" section.

Each frame in the network holds attributes (corresponding to header fields) for the storage time limit, how long it has spent in storage, the temporal frame length, the guard time, the path it has traveled, and its status (in transit, arrived, or discarded). We can simulate the network dynamics for a specified duration and collect data on the number of routed QKD signals, N, as well as the path they have traveled, i.e., the number of routers traversed and the average total time spent in storage, $\langle \sum_i T_s^i \rangle$. Note that signals from different frames will have a different total storage time, and so we take an average over all frames. We may then determine the average channel transmittance for each user-pairing,

$$\langle \eta_{tot} \rangle = 10^{\left(-\alpha L - \left\langle \sum_i loss_r^i \right\rangle\right)/10}, \tag{2}$$

where α is the attenuation coefficient (dB/km) of the network links, L is the distance between sender and receiver, and $\langle \sum_i loss_r^i \rangle$ is the average loss over all routers in the channel, found by Eq. (1).

The simulated N and $\langle \eta_{tot} \rangle$ may then be used by senders in the network to optimize their decoy-state parameters. Note that the network statistics correspond to a particular network configuration; namely, the topology, number of users, frame inter-arrival time, and routing protocol. Thus, these parameters must be known and fixed prior to a QKD session in order for user pairs to have accurate knowledge of their transmittance statistics. This is feasible in practice. For example, the network can employ traffic shaping[38] to ensure that frames from each sender arrive one at a time with inter-arrival times following the intended distribution. The remaining parameters typically do not change very frequently and their status can be updated as needed to all network users.

DATA AVAILABILITY

The raw data supporting the findings of this study are available from S.D. upon reasonable request.

CODE AVAILABILITY

The simulation code of this study is available from S.D. upon reasonable request.

Received: 21 March 2023; Accepted: 31 August 2023;
Published online: 09 September 2023

REFERENCES

1. Baran, P. *On Distributed Communications Networks* (RAND Corporation, 1962).
2. Abbate, J. *Inventing the Internet* (MIT Press, 1999).
3. DiAdamo, S., Qi, B., Miller, G., Kompella, R. & Shabani, A. Packet switching in quantum networks: a path to the quantum internet. *Phys. Rev. Res.* **4**, 043064 (2022).
4. Bennett, C.H. & Brassard, G. Quantum cryptography: public key distribution and coin tossing. *Proc. IEEE Int. Conf. Comp. Systems Signal Processing*, pp. 175–179 (1984).
5. Ekert, A. K. Quantum cryptography based on Bell's theorem. *Phys. Rev. Lett.* **67**, 661–663 (1991).
6. Townsend, P. D. Quantum cryptography on multiuser optical fibre networks. *Nature* **385**, 47–49 (1997).
7. Elliott, C. et al. Current status of the DARPA quantum network. *Proc. SPIE 5815, Quantum Information and Computation III* (2005).
8. Peev, M. et al. The SECOQC quantum key distribution network in Vienna. *N. J. Phys.* **11**, 075001 (2009).
9. Sasaki, M. et al. Field test of quantum key distribution in the Tokyo QKD Network. *Opt. Express* **19**, 10387–10409 (2011).
10. Fröhlich, B. et al. A quantum access network. *Nature* **501**, 69–72 (2013).
11. Tang, Y.-L. et al. Measurement-device-independent quantum key distribution over untrustful Metropolitan network. *Phys. Rev. X* **6**, 011024 (2016).
12. Chen, Y.-A. et al. An integrated space-to-ground quantum communication network over 4,600 kilometres. *Nature* **589**, 214–219 (2021).
13. Munro, W. J., Piparo, N. L., Dias, J., Hanks, M. & Nemoto, K. Designing tomorrow's quantum internet. *AVS Quantum Sci.* **4**, 020503 (2022).
14. Yoo, S. J. B. & Kumar, P. Quantum Wrapper Networking. In *2021 IEEE Photonics Conference (IPC)*, pp. 1–2. (IEEE, 2021).
15. Singal, A., Iyengar, S. S., Kumar, L. & Madni, A. M. Hardware routed quantum key distribution networks. *IET Quantum Commun* **3**, 127–138 (2022).
16. Ramaswamy, R., Weng, N. & Wolf, T. Characterizing network processing delay. In *IEEE Global Telecommunications Conference, 2004. GLOBECOM'04.*, **3**, 1629–1634 (2004).
17. Erven, C. et al. Studying free-space transmission statistics and improving free-space quantum key distribution in the turbulent atmosphere. *N. J. Phys.* **14**, 123018 (2012).
18. Vallone, G. et al. Adaptive real time selection for quantum key distribution in lossy and turbulent free-space channels. *Phys. Rev. A* **91**, 042320 (2015).
19. Wang, W., Xu, F. & Lo, H.-K. Prefixed-threshold real-time selection method in free-space quantum key distribution. *Phys. Rev. A* **97**, 032337 (2018).
20. Tamura, Y. et al. The first 0.14-dB/km loss optical fiber and its impact on submarine transmission. *J. Lightwave Technol.* **36**, 44–49 (2018).
21. Chapuran, T. E. et al. Optical networking for quantum key distribution and quantum communications. *N. J. Phys.* **11**, 105001 (2009).
22. Geng, J.-Q. et al. Quantum key distribution integrating with ultra-high-power classical optical communications based on ultra-low-loss fiber. *Opt. Lett.* **46**, 6099–6102 (2021).
23. Eriksson, T. A. et al. Wavelength division multiplexing of continuous variable quantum key distribution and 18.3 Tbit/s data channels. *Commun. Phys.* **2**, 9 (2019).
24. Mao, Y. et al. Integrating quantum key distribution with classical communications in backbone fiber network. *Opt. Express* **26**, 6010–6020 (2018).
25. Lo, H.-K., Curty, M. & Qi, B. Measurement-device-independent quantum key distribution. *Phys. Rev. Lett.* **108**, 130503 (2012).
26. Wang, C. et al. Measurement-device-independent quantum key distribution robust against environmental disturbances. *Optica* **4**, 1016–1023 (2017).
27. Bhaskar, M. K. et al. Experimental demonstration of memory-enhanced quantum communication. *Nature* **580**, 60–64 (2020).
28. Hwang, W.-Y. Quantum key distribution with high loss: Toward global secure communication. *Phys. Rev. Lett.* **91**, 057901 (2003).
29. Lo, H.-K., Ma, X. & Chen, K. Decoy state quantum key distribution. *Phys. Rev. Lett.* **94**, 230504 (2005).
30. Ma, X., Qi, B., Zhao, Y. & Lo, H.-K. Practical decoy state for quantum key distribution. *Phys. Rev. A* **72**, 012326 (2005).
31. Wang, X.-B. Beating the photon-number-splitting attack in practical quantum cryptography. *Phys. Rev. Lett.* **94**, 230503 (2005).
32. Lo, H.-K., Chau, H. F. & Ardehali, M. Efficient quantum key distribution scheme and a proof of its unconditional security. *J. Cryptol.* **18**, 133–165 (2005).
33. Lim, C. C. W., Curty, M., Walenta, N., Xu, F. & Zbinden, H. Concise security bounds for practical decoy-state quantum key distribution. *Phys. Rev. A* **89**, 022307 (2014).
34. Jackson, J. R. Networks of waiting lines. *Oper. Res.* **5**, 518–521 (1957).
35. Matloff, N. Introduction to discrete-event simulation and the SimPy language. *Dept. Computer Sci., Univ. Calif. Davis* **2**, 1–33 (2008).
36. Muller, K. & Vignaux, T. SimPy: Simulating Systems in Python. *O'Reilly*, 650 (2003).
37. Hagberg, A., Swart, P. & Chult, D. S. Exploring network structure, dynamics, and function using NetworkX. Technical report, Los Alamos National Lab (LANL), (2008).
38. Noormohammadpour, M. & Raghavendra, C. S. Datacenter traffic control: understanding techniques and tradeoffs. *IEEE Commun. Surv. Tutor.* **20**, 1492–1525 (2017).

ACKNOWLEDGEMENTS

We acknowledge helpful discussions with Hassan Shapourian and Glen Miller.

AUTHOR CONTRIBUTIONS

S.D., B.Q., and A.S. proposed the study of QKD in a packet-switched setting. R.M., S.D., and B.Q. designed the study and models. R.M. and B.Q. devised the key rate analysis. R.M. and S.D. created the simulation software. S.D. ran the simulations. R.M. wrote the manuscript with input from all other authors.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41534-023-00757-x.

Correspondence and requests for materials should be addressed to Stephen DiAdamo.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.