IEEE Transactions on Quantum Engineering

Received 13 May 2024; revised 20 September 2024; accepted 26 September 2024; date of publication 16 October 2024; date of current version 18 November 2024.

Digital Object Identifier 10.1109/TQE.2024.3476009

Quantum Switches for Gottesman–Kitaev–Preskill Qubit-Based All-Photonic Quantum Networks

MOHADESEH AZARI[1] (Graduate Student Member, IEEE), PAUL POLAKOS[2], AND KAUSHIK P. SESHADREESAN[1] (Member, IEEE)

[1]Department of Informatics and Networked Systems, University of Pittsburgh, Pittsburgh, PA 15260 USA
[2]Cisco Systems, New York, NY 10119 USA

Corresponding author: Mohadeseh Azari (e-mail: moa125@pitt.edu).

This work was supported in part by the National Science Foundation under Grant 2204985 and in part by Cisco Systems. An earlier version of this paper was presented in part at the IEEE International Conference on Communications (ICC 2024) [DOI: 10.1109/ICC51166.2024.10622457].

ABSTRACT The Gottesman–Kitaev–Preskill (GKP) code, being information theoretically near optimal for quantum communication over Gaussian thermal-loss optical channels, is likely to be the encoding of choice for advanced quantum networks of the future. Quantum repeaters based on GKP-encoded light have been shown to support high end-to-end entanglement rates across large distances despite realistic finite squeezing in GKP code preparation and homodyne detection inefficiencies. Here, we introduce a quantum switch for GKP qubit-based quantum networks. Its architecture involves multiplexed GKP qubit-based entanglement link generation with clients and their all-photonic storage, enabled by GKP qubit graph state resources. The switch uses a multiclient generalization of a recently introduced entanglement-ranking-based link matching heuristic for bipartite entanglement distribution between clients via entanglement swapping. Since generating the GKP qubit graph state resource is hardware intensive, given a total resource budget and an arbitrary layout of clients, we address the question of their optimal allocation to the different client–pair connections served by the switch such that the switch's sum throughput is maximized while also being fair in terms of the individual entanglement rates. We illustrate our results for an exemplary data center network, where the data center is a client of a switch, and all of its other clients aim to connect to the data center alone—a scenario that also captures the general case of a gateway router connecting a local area network to a global network. Together with compatible quantum repeaters, our quantum switch provides a way to realize quantum networks of arbitrary topology.

INDEX TERMS Entanglement swapping, Gottesman–Kitaev–Preskill (GKP) code, quantum resource allocation, quantum switches.

I. INTRODUCTION

Quantum networking is at the heart of the ongoing second quantum revolution [2]. At small distance scales, modular architectures for quantum computers comprised of extensive collections of interconnected small, finite-sized quantum logic units provide a way to scale up quantum computing power [3] in qubit platforms, such as trapped ions [4], [5], superconducting circuits [6], [7], and color centers in diamond [8]. On the other hand, large distance-scale quantum networks are key to enabling distributed quantum information processing [9] with applications in quantum data center networks [10], secure delegated quantum computation in the cloud [11], quantum key distribution networks [12], and quantum sensor networks [13], [14], [15].

Quantum communication needed for networking distinct quantum nodes irrespective of the distance scale is ubiquitously implemented using light. The various degrees of freedom of single photons, such as the polarization, time-bin, or spatio–spectro–temporal mode, provide means to encode quantum information in light [16], [17]. Given the relative ease of generating single-photon quantum states, these form the primary focus of present-day quantum networking efforts. However, modes of the quantized light field themselves, on the whole, are quantum objects, also referred to

as qumodes that can be prepared in a myriad possible multiphoton states and be used to encode quantum information more efficiently than with single photons [18]. Among such possibilities, the Gottesman–Kitaev–Preskill (GKP) bosonic error correcting code is known to be resilient to photon loss [19]. Qubits based on the GKP encoding have been shown to nearly achieve the quantum communication capacity of Gaussian thermal-loss channels under mean photon number constraint [20], which model most common transmission media, such as optical fiber and free-space links. As a result, GKP codes, although difficult to generate, are widely viewed as the future of quantum communication.

Quantum networking with light is enabled by specialized helper nodes, namely, quantum repeaters [21] and quantum switches (also referred to as quantum routers) [22], [23], [24], which consist of quantum optical sources and detectors, quantum memories, and fast optical switches. They can forward quantum data reliably in the face of photon loss and thermal noise and do so efficiently at rates above direct transmission [25]. While the former are line elements connecting two clients (i.e., nodes directly attached to it, which could be end-users or other repeaters or switches), the latter can switch between, connect, or correlate multiple clients. Together, they can be used to realize quantum networks of arbitrary topology at different distance scales.

Quantum repeaters based on the GKP code have been proposed and analyzed recently [26], [27], [28], [29]. Most notable among these for the entanglement rates enabled is the repeater of [27]. Its architecture uses multiplexed copies of the physical–logical GKP qubit resource. The logical qubit part consists of a collection of qubits prepared in the GKP code concatenated with a qubit quantum error correcting code and is retained at the repeater, serving as an all-photonic quantum memory, whereas the physical qubit part is simply a single GKP encoded qubit that is transmitted toward a neighboring node for interfacing via physical–physical GKP qubit Bell state measurement (BSM). These BSMs generate logical–logical GKP qubit-based entanglement links between pairs of adjacent nodes in a repeater chain. While the logical qubit can, in general, be any quantum error correcting code suitably chosen to provide robustness against given channel noise, Rozpdek et al. [27] considered the logical qubit to be seven physical GKP qubits encoded in the [[7,1,3]] Steane code. The overall entangled resource state (physical–logical Bell state), in this case, is an eight-qubit graph state of cube topology (up to local Hadamard gates), whose symmetric nature makes it convenient to analyze the performance of the repeater.

On the multiplexed logical–logical entanglement links, the repeaters of [27] implement an entanglement ranking-based link matching protocol, a heuristic where the entanglement links on either side of each repeater are ranked based on the quality of the link then matched across each repeater node by rank to perform error-corrected entanglement swapping between the corresponding logical GKP qubits (all-photonic quantum memories). The rankings are decided based on the analog outcome values of the physical–physical GKP BSMs that indicate the quality of the GKP qubit entanglement links. A chain of equispaced repeaters of this type was shown to support end-to-end entanglement rates as high as 0.7 entangled bits (ebits)/mode at total distances as large as 700 km under realistic assumptions for GKP qubit quality expressed in terms of GKP squeezing [30] and coherent homodyne detector efficiencies.

This article introduces and analyzes a quantum switch for GKP qubit-based entanglement distribution networks, whose architecture is compatible with that in [27]. We focus on bipartite entanglement switching, i.e., where the switch facilitates entanglement distribution between pairs of its clients, which may most generally be at different distances. We consider a heuristic for the switch that is a multiclient generalization of the entanglement ranking-based link matching protocol of Rozpdek et al. [27] where elementary entanglement links shared by the switch with the different clients are globally ranked and matched with each other to connect clients, or in other words, generate end-to-end entanglement links between clients. For the same choice of physical–logical entangled resource states as those in [27], we analyze the protocol for an exemplary data center network, in which the data center features as a client of the switch, and all the other clients intend to connect to the data center alone and not with each other. Given that the entangled resource states form the most valuable commodities at the nodes of the network, we study the problem of optimal allocation of these resources toward the different client pairings, or connections, such that the said protocol achieves the maximum sum throughput of the switch (or the total switch rate), while also being fair in terms of the individual entanglement rates.

The main contributions of this work can be summarized as follows.

1) We present an architecture for a GKP qubit-based quantum switch with multiple clients, where the clients are most generally at different distances from the switch and share different numbers of multiplexed elementary entanglement links.
2) We generalize the entanglement-ranking-based link matching heuristic of Rozpdek et al. [27] for the switch to distribute bipartite entanglement between clients via entanglement swapping.
3) For the simplest instance of the proposed switch, namely, one that connects just one client pair (i.e., enables just one connection) with the two clients being most generally at different distances (essentially an asymmetric repeater), we derive end-to-end entanglement rates based on the so-called six-state protocol [31], which is an achievable rate of distilling two-qubit maximally entangled states from the end-to-end entanglement states distributed by the switch.
4) For the two-client (single connection) switch, given a total number of GKP qubit-based entangled resource states, we numerically determine the optimal resource

IEEE Transactions on Quantum Engineering

allocation across the connection, i.e., allocation among the two clients to maximize end-to-end entanglement rate when the entanglement-ranking-based matching heuristic is used. Given the total distance between two clients, we also numerically determine the optimal placement of a switch/repeater node between the two.

5) For a switch that enables multiple connections, we numerically determine the optimal allocation of the entangled resource states such that the sum throughput of the switch is maximized while the different connections also receive fair individual entanglement rates. We elucidate this with the help of an exemplary data center network.

The rest of this article is organized as follows. Section II describes our switch model, architecture, and the generalized entanglement-ranking-based link matching (GERM) protocol heuristic for entanglement swapping. In Section III, we begin by analyzing the most straightforward instance of the proposed quantum switch connecting two users, most generally at different distances. Here, we make key observations on optimally allocating the GKP qubit-based entangled resource states at the switch, given a fixed number of resources, and placing the switch nodes, given a fixed total distance between users, to achieve the best sum end-to-end entanglement rates. In Section IV, we describe and analyze an illustrative yet comprehensive example of a multiclient quantum switch, namely, a data-center network. Here, we also consider entanglement-rate fairness and determine the optimal allocation of resource states that yield the highest total switch rate while providing fairness between all the entanglement connections. Finally, Section V concludes this article, and give general guidelines for the proposed quantum switch.[1]

II. SWITCH MODEL, ARCHITECTURE, AND PROTOCOL

Model: Consider the general layout of a quantum switch, as depicted in Fig. 1. We model the switch as having multiple clients (n), which are at different distances ($l_1, \ldots, l_n$), each attempting to generate different numbers of multiplexed elementary entanglement links with the switch ($k_1, \ldots, k_n$), periodically at a set clock rate. The switch connects these elementary entanglement links to generate end-to-end entanglement links between different pairs of clients. For simplicity, we assume a global clock rate. *Architecture:* The architecture of the proposed switch is primarily based on the GKP code. In short, the d-dimensional GKP code is a bosonic quantum error-correcting code whose code space is a d-dimensional qudit subspace of the infinite-dimensional Hilbert space associated with a single mode of the light field that has intrinsic resilience against photon loss. It is defined by a couple of syndrome measurements, which, along with conditional correction operations, can be repeatedly performed to preserve

[1]The MATLAB simulations of our heuristics are available at github.com/mohadesehazari98/Quantum_Switch

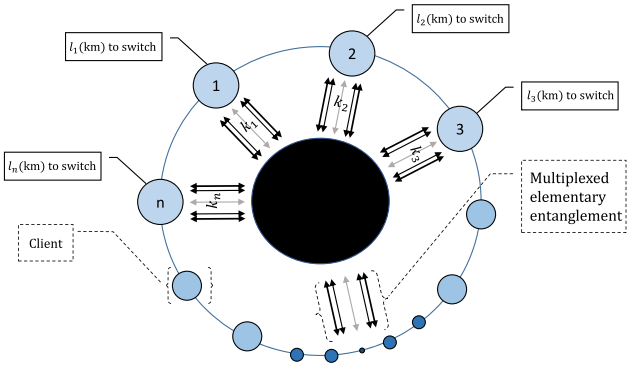


FIGURE 1. Generic quantum switch with n clients at different distances (l_i), and generating different numbers of multiplexed elementary entanglement links with the switch (k_i), where $i \in \{1, \ldots, n\}$.

a quantum state in the face of photon loss. Here, we focus on the $d = 2$ code, i.e., GKP qubits. For more details on the quantum physical description of realistic GKP qubits, associated noise characteristics (due to finite squeezing effects), and error correction properties, cf., [27].

The architecture involves elementary entanglement link generation between every client and the switch, and their storage. Toward this, all parties involved, i.e., the switch and each of the clients, use GKP qubit-based entangled resource states, which are physical–logical GKP qubit Bell states. The physical GKP qubit part of the entangled resource states, referred to as the outer leaf qubits, are transmitted toward one another. The outer leaf qubits meet halfway along the length of the transmission line to undergo GKP qubit BSM, resulting in a logical–logical GKP qubit entanglement link. The pair of logical GKP qubits, referred to as the inner leaf qubits, emulates quantum memories and thus acts as entanglement storage. In this work, the logical GKP qubit comprises the GKP qubit code concatenated with the [[7,1,3]] Steane code. The [[7,1,3]] Steane code is a qubit quantum error correcting code that encodes one logical qubit in seven physical qubits and can correct arbitrary errors on up to any one of the physical qubits. For more details on quantum error correcting codes, cf., [32]. In this case, the entangled resource state is equivalent to an eight-qubit graph state of cube topology up to Hadamard gates on four of the eight qubits. The entangled resource state can, e.g., be generated from a factory of individual realistic, finitely squeezed GKP qubits by a sequence of graph state fusion operations, cf., [27] for a detailed description of the scheme.

Another key feature of the architecture is multiplexing—more precisely, spatial multiplexing [33], where each client–switch pair attempts to generate multiple elementary entanglement links simultaneously over multiple independent channels. It is assumed that the switch and the clients are equipped sufficiently abundantly with physical GKP qubit resources required to prepare the necessary number of copies of physical–logical entangled resource states per clock cycle to support multiplexed elementary link generation near deterministically, as worked out in [27]. Moreover, the GKP

qubit BSM, which can be implemented as dual homodyne detection, i.e., a beam splitter followed by measurements of orthogonal quadratures on the two interfered modes, is inherently deterministic. As a result, all the multiplexed elementary link generation attempts are assumed to succeed deterministically. However, the quality of each link depends on the continuous real-valued BSM outcomes.

Finally, for end-to-end entanglement link generation between two clients, the architecture relies on logical–logical error-corrected entanglement swap operations between their inner leaf qubits at the switch that connect the corresponding elementary entanglement links. The hitherto described architecture of the switch is compatible with that in [27], which we thus refer the reader to for more details.

Protocol: Given multiple elementary entanglement links with each client, the proposed switch deploys a generalized multiclient version of the entanglement-ranking-based link matching protocol of Rozpdek et al. [27] to generate end-to-end entanglement links. To understand the algorithm behind the GERM protocol, consider the k_i multiplexed elementary entanglement links generated between the switch and the ith client in a given protocol round. Each of these links is of a different entanglement quality, quantified by the likelihood of no logical error on the outer leaf qubits involved in the BSM that generated the link, given by

$$P_{\text{no-error}} = (1 - P_p(p_0))(1 - P_q(q_0)). \tag{1}$$

Here, $P_p(p_0)$ and $P_q(q_0)$ are the likelihoods of incurring an error (incorrect detection of the GKP qubit state) when measuring in the p and q quadratures and observing real-valued outcomes p_0 and q_0, respectively. The aggregate collection of all the $\sum_{i=1}^{n} k_i$ elementary entanglement links is then *globally* ranked based on the value of their $P_{\text{no-error}}$ of (1), referred to as the outer leaf error, and sorted in descending order. The switch matches and connects pairs of links identified from top to bottom, as long as they are not with the same client and belong to the requested connections. The algorithm removes matched links to resume connecting subsequent pairs. In this work, we have enough links on the data center to leave no unmatched links. For brevity of analysis, we do not carry forward unused elementary entanglement links to subsequent protocol rounds; they are discarded. The generalized protocol is described below in terms of a pseudocode.

1: $n \leftarrow$ number of users
2: $k_i \leftarrow$ the number of multiplexed links of user i
3: $\sum_i k_i \leftarrow k_{\text{total}}$
4: ErrLik $\leftarrow$ Concatenate($P_{\text{no-error},i}$, $i \in \{1, \ldots, n\}$)
5: ErrLik $\leftarrow$ descending Sort(ErrLik)
6: **while** ErrLik $\neq 0$ **do**
7: count $\leftarrow 2$
8: **while TRUE do**
9: ErrLik(1) is $P_{\text{no-error},i}(j)$
10: ErrLik(count) is $P_{\text{no-error},i'}(j')$
11: **if** $(i \leftrightarrow i')$ is a requested connection **then**
12: **Connect** user i's j^{th} link to user i''s j'^{th} link
13: **Remove** ErrLik(1) and ErrLik(count)
14: **Break**
15: **else**
16: count $\leftarrow$ count $+ 1$
17: **end if**
18: **end while**
19: **end while**

Here we explain the notation used in the pseudocode. First, the total number of entangled resources limits the total number of elementary links between the different clients and the switch, as suggested by line 3. For example, if three users request to connect to the switch (each user should demand connection with at least one of the other users; otherwise, the switch will not dedicate resources to that user), then $k_1 + k_2 + k_3 = k_{\text{total}}$. How we determine k_1 through k_3 is related to the field optimization over the requested connections, which will be explained in detail in the following sections. In the fourth line, $P_{\text{no-error},i}$ is a list of size k_i, where k_i represents the number of multiplexed elementary links between the switch and user i. Each value of $P_{\text{no-error},i}$ indicates the probability of no logical error across k_i links connecting user i to the switch. When $P_{\text{no-error},i}$ of all the users are concatenated, the resulting list is called the error likelihood array (ErrLik) of size k_{total}. To enable the best quality links between the different users, the values of ErrLik need to be globally sorted in descending order from the best link to the weakest. One can, for e.g., use *numpy.concatenate* in Python and *vertcat* in MATLAB, followed by the *sort* command. To determine the list of all valid link matchings, lines 7 through 18 are repeated till the ErrLik becomes an empty array. The notations (i, i', j, j') of lines 9–12 are arbitrary and show possible users (i, i') and their corresponding links (j, j'). The command *Remove* in line 13 is equivalent to *numpy.delete(ErrLik, [1, count])* command in Python and *ErrLik(setdif(1:numel(ErrLik), [1, count]))* in MATLAB, which is to remove already matched links from the sorted list of links.

III. SWITCH CONNECTING TWO CLIENTS

Here, we analyze the most straightforward instance of the proposed switch, namely, one with just two clients that are most generally at different distances (l_1, l_2) and having different numbers of multiplexed elementary entanglement links with the switch (k_1, k_2), as depicted in Fig. 2. We will first derive an expression for the end-to-end entanglement rate in terms of ebits or maximally entangled qubit pairs (which can be distilled from the end-to-end entanglement links) generated per time step of the switch protocol. Subsequently, for any given k_{total}, i.e., the total number of multiplexed elementary entanglement links, that the switch

IEEE Transactions on Quantum Engineering

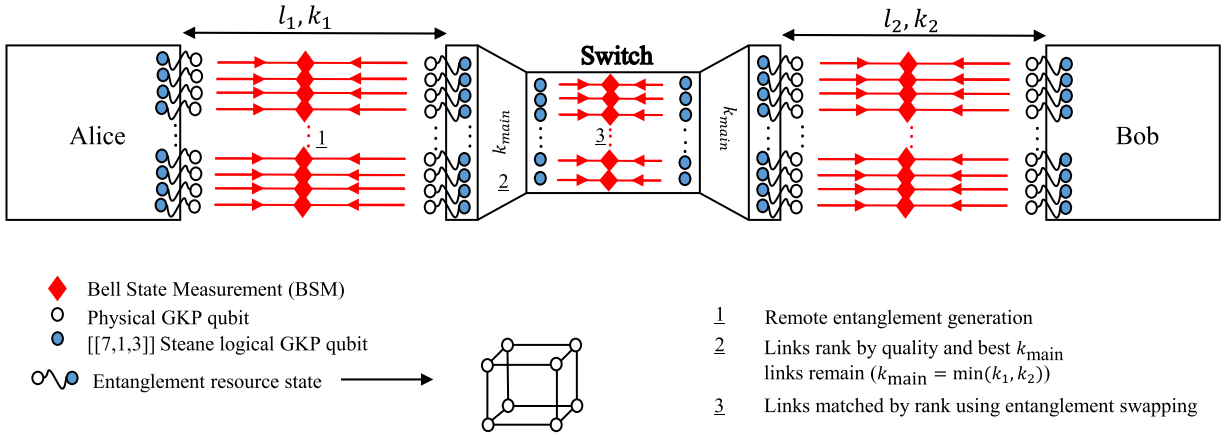


FIGURE 2. In its simplest form, the proposed multiplexed all-photonic quantum switch with GKP qubit resources encoded in the [[7,1,3]] Steane code consists of just two clients. The clients are located from the switch at known distances (l_1, l_2). The switch prepares $k_{\text{total}} = k_1\,(\text{left}) + k_2\,(\text{right})$ entangled resource states that correspond on the physical level to a cube graph state of eight GKP qubits. Remote entanglement generation is performed between the switch and each client by sending the bare physical GKP qubits (white/empty circles) toward each other for BSM. The GKP syndromes obtained from the real-valued BSM outcomes help rank the elementary links (red arrows) according to their estimated reliability. This ranking information, as well as logical BSM outcomes, is sent to the switch. The switch chooses the best $k_{\text{main}} = \min(k_1, k_2)$ links from each channel (left and right) to perform entanglement swapping on the concatenated-coded qubits (blue/filled circles).

can create for the two clients per time step, we will optimize the end-to-end entanglement rate over all possible allocations such that $k_1 + k_2 = k_{\text{total}}$.

A. END-TO-END ENTANGLEMENT RATES

First of all, we note that when $k_1 \neq k_2$, the overall number of end-to-end entanglement links that the switch can facilitate is $k_{\text{main}} = \min(k_1, k_2)$, where each client should use their best k_{main} links in quality. Second, in generating the multiplexed entanglement links between the switch and the two clients, the outer leaf qubits travel $l_1/2$ and $l_2/2$, respectively. Meanwhile, in local fiber spools, the corresponding inner leaf qubits travel twice the distance as the outer leaf qubits do, i.e., l_1 and l_2, respectively. This is due to the fact that the inner leaves serve as quantum memories for the time duration of elementary entanglement link generation, i.e., the total time incurred in the outer leaves undergoing the BSM and the analog outcomes from the measurement (containing information regarding the quality of the generated links) reaching the switch. When $l_1 \neq l_2$, every inner leaf qubit of elementary entanglement links travels an optical fiber of length given by $\max(l_1, l_2)$ to ensure that the switch has the ranking information of its entanglement links from both the clients.

The scheme for generating the GKP qubit-based entangled resource states from [27] involves a postselection test as part of graph state fusion operations at various intermediate points of the scheme (See Appendix A for a mathematical description of the test). The test involves a discard window size parameter ν, whose choice determines the quality of the qubits in the resource states eventually produced in terms of the probability of logical errors on the qubits, which has a role to play in the end-to-end rate calculation. The larger the discard window size used, the better the quality, i.e., lower the probability of errors on the qubits. However, the cost associated with the better quality is a lower probability of success of the postselection test, which implies higher resource overheads for the generation scheme. For their use in quantum repeaters and switches, it is sufficient for the GKP-qubit-based entangled resource states to be prepared at a quality, where the logical error probabilities on the qubits are commensurate to the probability of errors arising from the attenuation in optical fiber transmission of the inner leaf qubits in the local fiber spools during storage. In the present scenario of a switch with two clients, in the case of multiple nonidentical elementary links between the switch and each of the two clients, the required level of resource state error suppression in the resource state generation scheme via the choice of the discard window size thus depends on the maximum logical error probabilities of the inner leaf qubits.

In any given time step, the jth end-to-end entanglement link between two clients is created by entanglement swapping of the two clients' jth-ranked elementary entanglement links at the switch, where $j \in (1, \ldots, k_{\text{main}})$. The rate of generating perfect ebits from these end-to-end entanglement links is a function of the total inner and outer leaf logical error probabilities inclusive of errors from original entangled resource state preparation and attenuation in optical fiber transmission (internodal and local fiber spools), $Q_{X/Z,\text{outer},(i)}(j)$ and $Q_{X/Z,\text{inner},(i)}(j)$, respectively, of the parent elementary entanglement links associated with each of the two clients $i \in \{1, 2\}$. Here, $Q_{X/Z,\text{outer},(i)}$ is a k_i-dimensional vector, which holds the logical error probability on the outer leaf qubits of the elementary entanglement link between the ith client and the switch. On the other hand, $Q_{X/Z,\text{inner},(i)}$ is a $k_i \times 2$ matrix, which holds the logical error probability of the corresponding

inner leaf qubits, where the two columns track two different cases, as explained below.

Given that the inner leaf GKP qubits are further error-protected by an outer error correcting code, namely, the Steane code, the error syndromes of the latter can be factored in along with those of the GKP qubit syndromes to find the best end-to-end links. When the encoded inner leaf qubits are measured, we get a syndrome that is either zero ($s = 0$) or one ($s = 1$). A zero syndrome means that the p (or q) quadrature measurement did not detect any errors or that there is a three-qubit error, which would be a logical error at the level of the outer code that is less likely. On the other hand, a nonzero syndrome suggests a single- or two-qubit error, which with a high probability means incorrect identification of one or two physical qubits. As a result, the switch prefers an end-to-end link with $s = 0$ [27]. The first column of each $Q_{X/Z,\text{inner},(i)}$ represents the inner leaf logical error if no Steane code error syndrome ($s = 0$) is detected, and the second represents the case where a Steane code error syndrome ($s = 1$) is detected. The total error probability of the jth-ranked elementary entanglement link with the ith client using the Steane-code-level error syndrome s is given by

$$Q_{X/Z,(i)}(s, j) = Q_{X/Z,\text{inner},(i)}(s, j)\left(1 - Q_{X/Z,\text{outer},(i)}(j)\right) + \left(1 - Q_{X/Z,\text{inner},(i)}(s, j)\right) Q_{X/Z,\text{outer},(i)}(j). \quad (2)$$

The above equation follows from the fact that any elementary link experiences a logical error when either its inner qubits or outer qubits contain an error, but not both, as two X (or Z) errors over a link would nullify each other.

To classify end-to-end entanglement links based on the Steane-code-level error syndromes ($s = 1$) of their parent elementary entanglement links, we define the binary vectors $\vec{m}_{X/Z} = (\{m_{X/Z}(i) : i \in \{1, 2\}\})$. We note that

$$\|\vec{m}_{X/Z}\|_1 = c_{X/Z} \in \{0, 1, 2\}. \quad (3)$$

The total error probability associated with the jth-ranked end-to-end entanglement link and a given $\vec{m}_{X/Z}$ is then given by

$$Q_{X/Z,\text{end}}(\vec{m}_{X/Z}, j) = (1/2) \times \left(1 - \prod_{i=1}^{2} \left(1 - 2Q_{X/Z,(i)}(s = 1, j)\right)^{m_{X/Z}(i)} \times \left(1 - 2Q_{X/Z,(i)}(s = 0, j)\right)^{1-m_{X/Z}(i)}\right). \quad (4)$$

The total end-to-end entanglement rate of k_{main} multiplexed links when distilled separately from different j's and different $\vec{m}_{X/Z}$'s is given by

$$R_{\text{e2e}} = \sum_{j=1}^{k_{\text{main}}} \sum_{\vec{m}_X, \vec{m}_Z} p_X(\vec{m}_X) p_Z(\vec{m}_Z) \, r\left(Q_{X,\text{end}}(\vec{m}_X, j), Q_{Z,\text{end}}(\vec{m}_Z, j)\right). \quad (5)$$

Here, r is the secret-key fraction or a lower bound on distillable entanglement. The quantity $p_{X/Z}(\vec{m}_{X/Z})$ is the probability of the parent elementary entanglement links having error syndromes $\vec{m}_{X/Z}$ on their inner leaves, given by

$$p_{X/Z}(\vec{m}_{X/Z}) = \prod_{i=1}^{2} t_{X/Z,(i)}^{m_{X/Z}(i)} \left(1 - t_{X/Z,(i)}\right)^{1-m_{X/Z}(i)} \quad (6)$$

where $t_{X/Z,(i)}$ is the probability of an error syndrome ($s = 1$) on the inner leaves of the elementary link with the ith client. The quantities $t_{X/Z,(i)}$, $Q_{X/Z,\text{inner},(i)}(s = \{0, 1\})$, and $Q_{X/Z,\text{outer},(i)}(j = \{1, \ldots, k_i\})$ are all obtained through simulation.

Note that the above derivation can be easily generalized to a chain of quantum repeaters of asymmetric spacing with $i \in \{1, 2, \ldots, n_{\text{rep}} - 1\}$ in (4) and (6), where n_{rep} represents the total number of repeaters connecting an end-to-end connection.

B. OPTIMIZING RESOURCE ALLOCATION

For any given resource budget (at the switch), we aim to identify the optimal allocation that maximizes the proposed switch's end-to-end entanglement rate. In other words, given two clients at distances (l_1, l_2) and a total number of elementary entanglement links (k_{total}) that the switch can create cumulatively with the clients in a time step, we are interested in

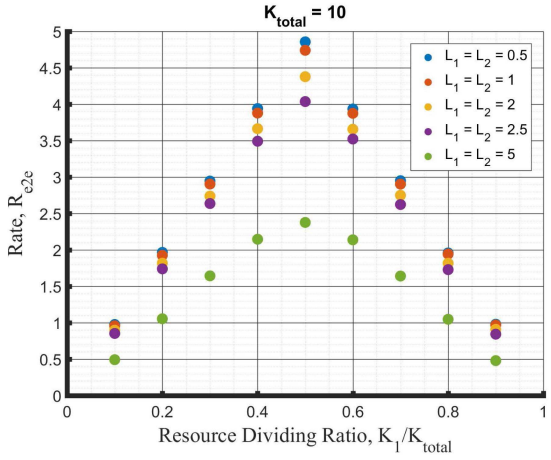
$$\max_{k_1, k_2} \; R_{\text{e2e}}(k_1, k_2; l_1, l_2) \quad \text{s.t. } k_1 + k_2 = k_{\text{total}} \quad k_1, k_2 \in \mathbb{Z}^+. \quad (7)$$

C. RESULTS

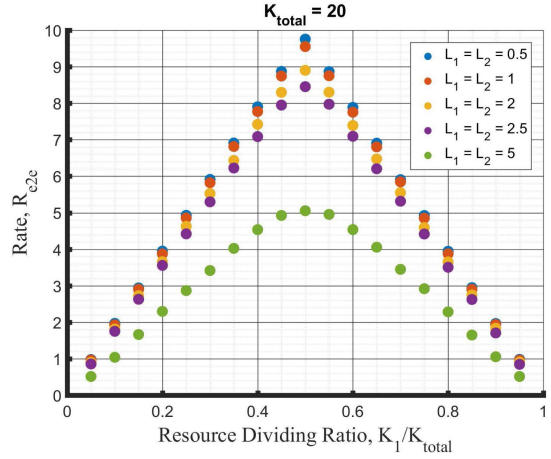
We ran numerical simulations to test the performance of the proposed quantum switch. The goal was to evaluate the total end-to-end entanglement rate between two clients at distances (l_1, l_2) away from the switch, where $l_i \in \{0.5, 1, 2, 2.5, 5\}$(km) $\forall i \in \{1, 2\}$. We simulated each set of distances, with a total number of entangled resource states being $k_{\text{total}} \in \{10, 20, 50\}$. Our metric for evaluation was the total number of ebits established between the clients. Fig. 3 shows the symmetric case of clients having the same distance from the switch ($l_1 = l_2 = l$) for different values of l and k_{total}. The plot shows the total switch rate as a function of resource allocation (tracked by the ratio k_1/k_{total}). It is found that an equal allocation of $k_1/k_{\text{total}} = 0.5$ maximizes the end-to-end entanglement rate, as expected. Also, comparing the results over different k_{total} reveals that the end-to-end rate increases as we increase the total resource budget but then quickly saturates. Note that as in (5), the total switch rate (R_{e2e}) is limited by the sum of per-mode rates, which is equal to $k_{\text{main}} = k_{\text{total}}/2$.

Fig. 4 shows the resource allocations that maximize the total end-to-end rate for more general settings beyond $l_1 = l_2$.

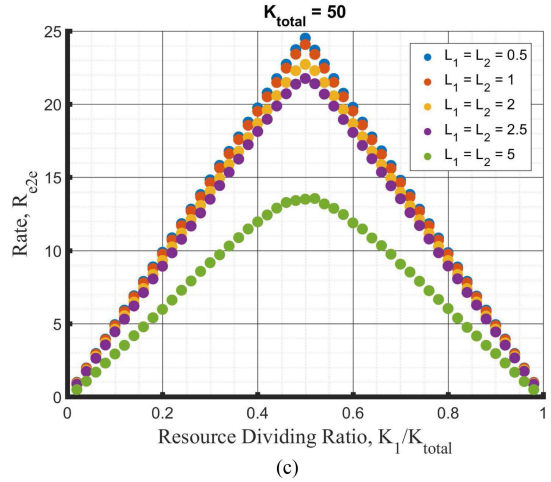
IEEE Transactions on Quantum Engineering



(a)

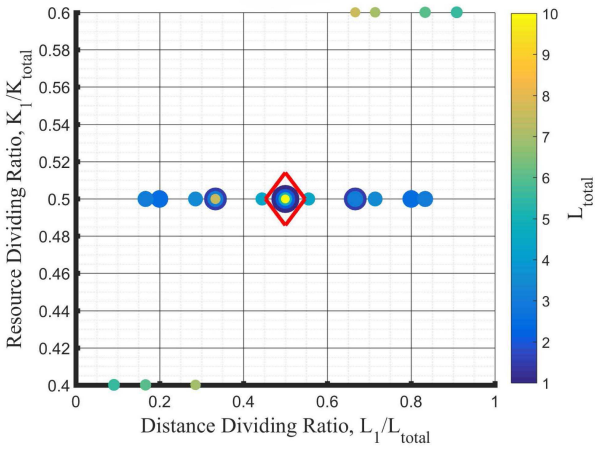


(b)

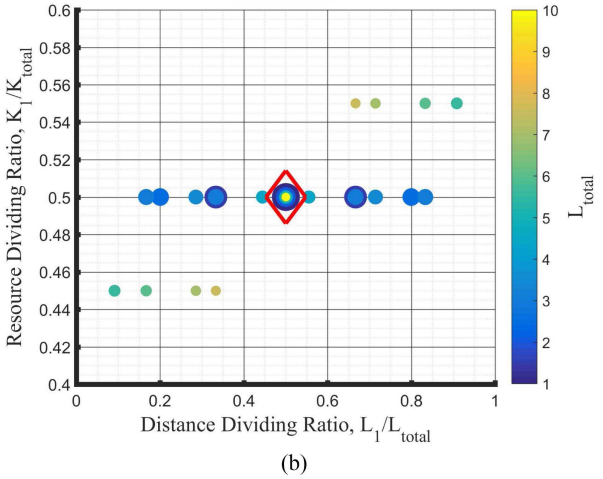


(c)

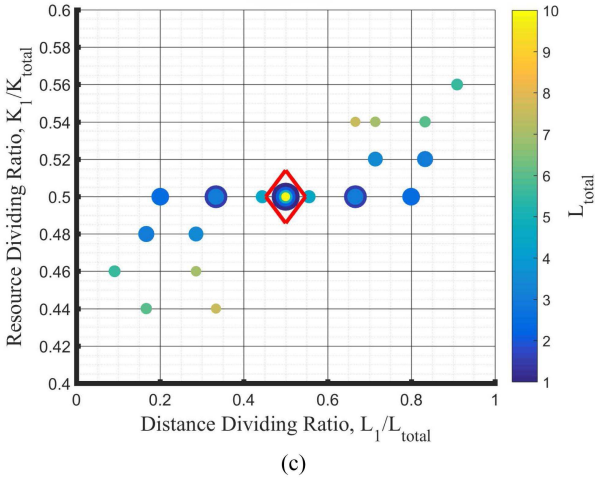
FIGURE 3. Total switch rate as a function of resource allocation. For a given setting (k_{total}, l_1, l_2), the maximum rate corresponds to the symmetric resource allocation ($k_1 = k_2$). The color-coding of the plot shows that users will experience a higher total switch rate the closer they get to the switch. (a) Total number of resource states (k_{total}) as 10. (b) Total number of resource states (k_{total}) as 20. (c) Total number of resource states (k_{total}) as 50.



(a)



(b)



(c)

FIGURE 4. General case of a two-client switch, where $l_1 + l_2 = l_{\text{total}}$ ($l_i \in \{0.5, 1, 2, 2.5, 5\}$km). For every setting ($l_1$, l_2, k_{total}), the resource dividing ratio of the optimum resource allocation is tracked by the ratio of elementary link distance over the total distance of the setting (l_1/l_{total}). The optimum allocation is found to be $k_1 = k_2 = k_{\text{total}}/2$. The red diamond shows the maximum total rate R_{e2e}, which belongs to $l_1 = l_2 = l_{\text{total}}/2$ and $k_1 = k_2 = k_{\text{total}}/2$. (a) Total number of resource states (k_{total}) as 10. (b) Total number of resource states (k_{total}) as 20. (c) Total number of resource states (k_{total}) as 50.

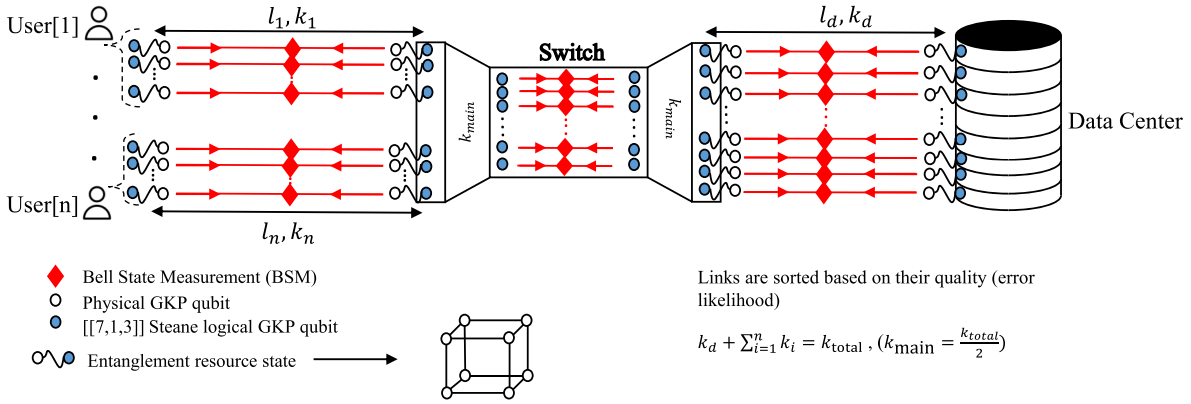


FIGURE 5. Architecture of the proposed multiplexed all-photonic quantum switch in a multiclient data-center network scenario. The data center-switch elementary entanglement links are ranked independently from the elementary links between the switch and the other clients. Links from these two ranked lists are matched and connected via entanglement swapping at the switch to generate end-to-end entanglement links.

Here, as the total distance between the two users (l_{total}) increases, the circle's color becomes brighter, and its diameter decreases. The reason for using both the color and size of the circles to represent the total distance is to distinguish between different $l_1 + l_2$ values with the same dividing ratios (l_1/l_{total}) when they coincide as is the case with the red diamond points in Fig. 4. We notice that with small deviations, the maximum rate belongs to symmetric allocation, i.e., $k_1 = k_2 = k_{\text{total}}/2$. These deviations, mostly in larger distances, are close enough to be considered as a symmetric allocation $(k_1 - k_2)/(2k_{\text{total}}) \ll \epsilon_{k_{\text{total}}}$ with $\epsilon_{10} = 0.1$ and $\epsilon_{50} = 0.06$. Not only does the symmetric allocation of resources turn out to be optimal, but also the symmetric disposition of clients about the switch for any fixed l_{total} yields the highest rate. To simplify the illustration, we have omitted the third dimension, which represents R_{e2e}. However, the optimal value of the latter quantity is marked as belonging to $\{l_1 = l_2, k_1 = k_2\}$. So, to operate the proposed quantum switch in its most straightforward instance with just two clients and with k_{total} entangled resource states, the setting that maximizes the end-to-end rate is to place the switch/repeater in the middle, i.e., $l_1 = l_2 = l_{\text{total}}/2$ and allocate resources equally, i.e., $k_1 = k_2 = k_{\text{total}}/2$.

We next study a case where client-1 has a distance of $l_1 = 0.5$ km from the switch, whereas client-2 has a larger distance of $l_2 = 5$ km. The benefit of increasing the number of second client's multiplexed links is that the error likelihood of its best elementary entanglement links would decrease, meaning client-2 will have more "good-quality" links in hand than before. This is useful since client-1 is already close to the switch, which guarantees that the error likelihood is small. However, when $k_1 \neq k_2$, the switch will have to throw away $\max(k_1, k_2) - \min(k_1, k_2)$ number of precious entangled resource states. Our simulation results indicate that the symmetric allocation $k_1 = k_2$ yields a higher total end-to-end rate than the asymmetric allocation $k_1 < k_2$. In other words, increasing k_2 has a downside that far exceeds its benefits because, as mentioned earlier, resource state generation is the most expensive part of a GKP switch architecture. Thus, employing all the entangled resource states, i.e., $k_1 = k_2$, results in the switch's best total end-to-end entanglement rate.

IV. SWITCH CONNECTING MULTIPLE CLIENTS

Having described the simplest two-client (i.e., single connection) switch and determined the optimal resource allocation for maximum end-to-end entanglement rate, we now move to multiple clients. To best elucidate the workings of the proposed switch in a multiclient scenario, we will focus on a particular example, namely, that of a data center network where one of the clients of the switch is a data center and all the other clients look to connect to the data center alone, as depicted in Fig. 5. The following describes the data center switch network setting, some nuances of switch operation, and our performance metrics. The latter includes the switch's sum throughput and a measure of entanglement rate fairness between the different clients. Given the switch's total entangled resource state budget, we will optimize the switch's sum throughput over all possible resource allocations under an entanglement fairness constraint.

A. DATA CENTER SWITCH NETWORK: OPERATIONS AND PERFORMANCE METRICS

Consider a switch network with a data center and n other clients, at distances $(l_d, l_1, \ldots, l_n)$, respectively, from the switch, as shown in Fig. 5. Let the number of elementary links between the switch and the data center be k_d, and between the switch and the other clients be $k_i \forall i \in \{1, \ldots, n\}$ such that $k_d + \sum_{i=1}^{n} k_i = k_{\text{total}}$, where k_{total} is the total resource budget at the switch. As all the connections facilitated by the switch include the data center, the GERM protocol (explained in Section II) takes a simplified form in this situation. The switch matches the ranked links from the data center with the ranked links of all other clients to generate

IEEE Transactions on Quantum Engineering

end-to-end entanglement links. The total number of end-to-end entanglement links that can be generated is given by $\min\{k_d, \sum_{i=1}^{n} k_i\}$.

We note that in operating the multiclient switch, to ensure that the switch receives every outer leaf ranking information required to implement the ranking-based matching protocol, all the inner leaf qubits at the switch should be stored for a time commensurate to light traveling a local fiber spool of length $\max\{l_1, l_2, \ldots, l_n, l_d\}$. Regarding the quality of GKP qubits, sufficient GKP squeezing is necessary to prepare resource states. As mentioned in Section III, the logical error probabilities resulting from finite GKP squeezing in individual GKP qubits are controlled by the discard window size ν. The error probability from the discard window does not need to be smaller than the logical error associated with the longest optical fiber transmission of the inner leaf qubits during local all-optical storage, expressed as $\max\{l_1, l_2, \ldots, l_n, l_d\}$.

In this scenario, we will assume that all users are at the end nodes of a local network trying to connect to the nearest data center. Therefore, it is reasonable to consider that the distances from the end-users to the switch are similar and nonzero. This positioning is the most optimal for users, as large distance gap between the users' outer leaves lead to limitations in resource allocation fairness and increase the length of the inner optical fiber spool. Placing the switch within the data center ($l_d = 0$) is not preferable as it introduces more asymmetry to the network. The best approach is to position the switch midway between the users' surrounding area and the data center.

The figure of merit here is the sum throughput of the switch, which is simply the sum of rates of all the individual end-to-end entanglement links that the switch enables. In the case of the data-center network example, it is given by

$$R_{\mathrm{s}} = \sum_{i=1}^{n} R_{\mathrm{e2e}}^{(i)}(k_d, k_i; l_d, l_i) \tag{8}$$

where the end-to-end entanglement rate of each individual connection $R_{\mathrm{e2e}}^{(i)} \forall i \in \{1, \ldots, n\}$ is defined and evaluated just as in (5).

Here, we also consider a fairness measure that we call Euclidean fairness, which shows the proximity of entanglement rates of different switch connections. It is defined by taking the Euclidean distance between the different clients' end-to-end rates ($R_i \equiv R_{e2e}^{(i)}$) scaled by their average

$$F(R_1, \ldots, R_n) = \frac{d(R_1, R_2, \ldots, R_n)}{\langle R_1, R_2, \ldots, R_n \rangle} \tag{9}$$

$$d^2(R_1, R_2, \ldots, R_n) = \frac{1}{2} \sum_{i,j=1}^{n} |R_i - R_j|^2. \tag{10}$$

The smaller the fairness measure F, the closer the client rates are to each other ($0 \leq d^2(R_1, \ldots, R_n) \leq n$). This metric ensures that, while maximizing total throughput, distant clients are not disadvantaged compared to nearby clients in terms of end-to-end entanglement rates with the data center. This guarantees that no clients are left unserved or underserved by the switch. An equally good alternative to our measure is the well-known Jain's fairness index [34].

B. OPTIMIZING RESOURCE ALLOCATION

Similarly to Section III, for a data-center switch network with the data center and the other clients at distances $(l_d, l_1, \ldots, l_n)$ and any given resource budget k_{total} at the switch, we now look to identify optimal allocation $(k_d, k_1, \ldots, k_n)$ that maximizes the sum throughput R_{s} of (8) for the proposed switch. We do so under the constraint that the fairness measure F of the client entanglement rates is bounded by a threshold F_{t}, whose value is chosen suitably. In other words, we are interested in

$$\begin{aligned} &\max_{k_d, k_1, \ldots, k_n} R_{\mathrm{s}}(k_d, k_1, \ldots, k_n; l_d, l_1, \ldots, l_n) \\ &\text{s.t. } k_d + \sum_{i=1}^{n} k_i = k_{\mathrm{total}} \\ &F(R_1, \ldots, R_n) < F_{\mathrm{t}} \\ &[k_d, k_1, \ldots, k_n] \in \mathbb{Z}^+. \end{aligned} \tag{11}$$

First of all, recall the observation from Section III, namely, that in any single connection between clients A and B through the proposed switch with a total resource state budget of k_{total}, regardless of (l_A, l_B), the setting that maximizes the rate R_{e2e} is $k_A = k_B = k_{\mathrm{total}}/2$. Based on this, we can conclude that for optimal operation, $k_d = k_{\mathrm{total}}/2$. This is because, the above observation implies $k_d = \sum_{i=1}^{n} k_i$, which along with $k_d + \sum_{i=1}^{n} k_i = k_{\mathrm{total}}$, we can derive the conclusion. Therefore, the optimization allocates the remaining half of the resources at the switch toward the other clients, namely, $k_i, i \in \{1, 2, \ldots, n\}$. Equivalently, the resource allocation can be thought of as allocating $2k_i \forall i \in \{1, \ldots, n\}$ resources toward each "client-i-data-center connection," and dividing them equally between client-i and the data center. This will ensure that the GERM protocol, when implemented at the switch, will leave no links unutilized. The connection-based resource allocation is more helpful when the switch has general "client-i-client-j connection".

C. RESULTS

We performed numerical simulations of the proposed quantum switch to identify optimal resource allocations in data center switch networks of different settings. Fig. 6 summarizes the analysis of a switch network of two end-user clients (client-1 and client-2) and a data center (client-3), where client-1 and client-2 request to connect to the data center. First, as the previous section has indicated, equal resource allocation across a single connection leads to the maximum end-to-end rate. This approach could lead to a more efficient use of resources and a better overall experience for all users involved. Accordingly, in scenarios where clients are symmetric, i.e., $l_1 = l_2$, and therefore the error loads (inversely proportional to the length of the link) are equal, an equal

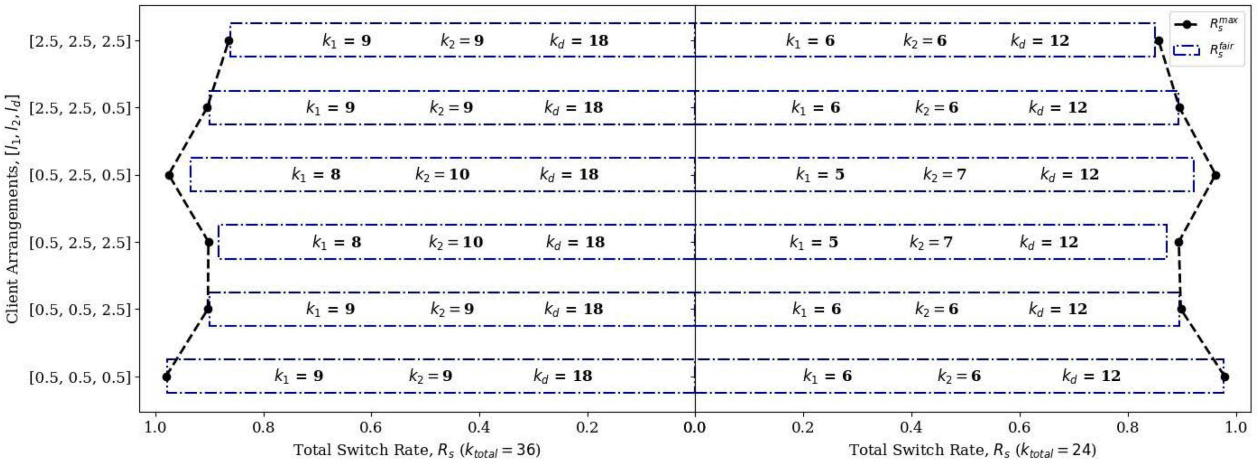


FIGURE 6. Switch rate analysis for a two-client, one-data center network with varying spacing $[l_1, l_2, l_d]$. The switching rate corresponding to fair resource allocation is represented by the width of the bars along the x-axis for different client arrangements and two different values of total resource state budget k_{total}. When fairness is ignored, the black dots connected by dashed lines indicate the maximum possible switch rate for the given client arrangement and total resource state budget.

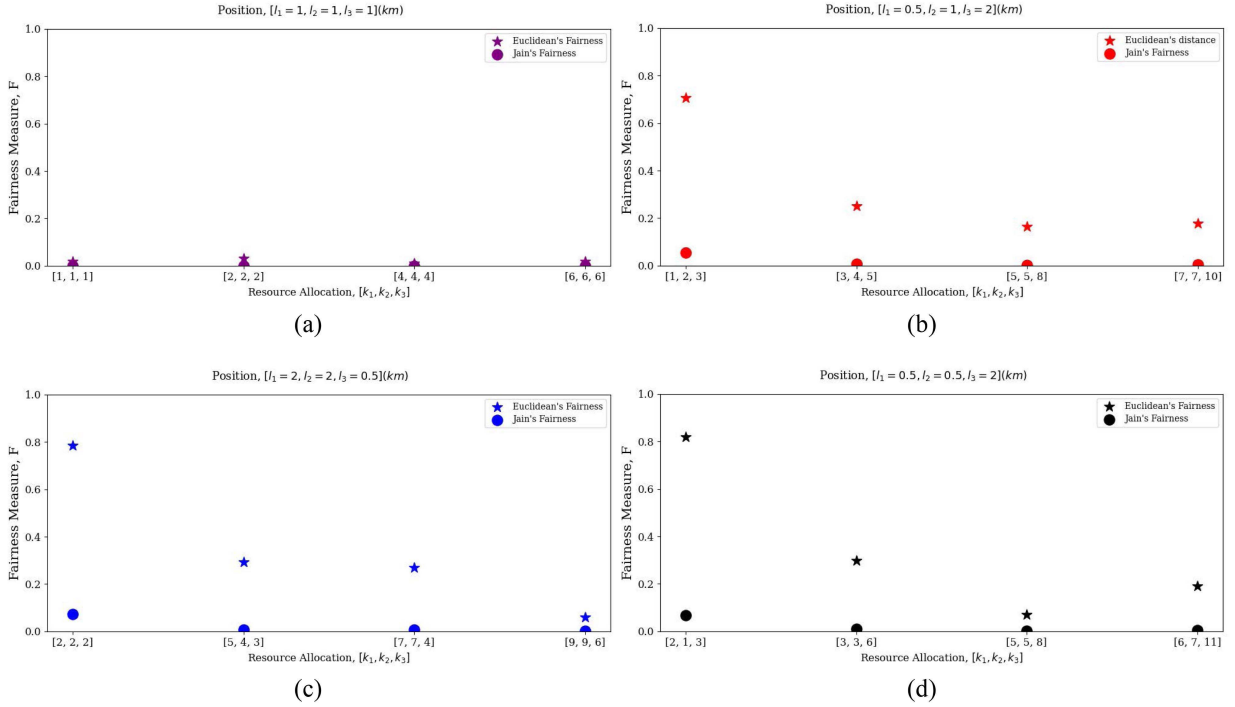


FIGURE 7. The fairness measure (F) of the fairest resource allocation (k_1, k_2, k_3) among the three clients that wish to connect to a data center at a distance l_d from the switch ($l_d = 5$(km)). We plot the measure for a given clients' setting (l_1, l_2, l_3) and a limited resource budget of k_{total} (where $k_1 + k_2 + k_3 = k_{\text{total}}/2$). The fairness measures are tailored (Euclidean and Jain's fairness) so that the higher the value, the less fair the allocation. We have $k_{\text{total}} \in \{6, 12, 24, 36, 48\}$ and we choose $[l_1, l_2, l_3]$ so that it includes (a) a symmetric case $[l_1 = 1, l_2 = 1, l_3 = 1]$(km), (b) an asymmetric case $[l_1 = 0.5, l_2 = 1, l_3 = 2]$(km), (c) a case where only one client is relatively close $[l_1 = 2, l_2 = 2, l_3 = 0.5]$(km), and (d) a case where only one client is relatively far away $[l_1 = 0.5, l_2 = 0.5, l_3 = 2]$(km).

distribution of resources $k_1 = k_2$, as anticipated, results in the highest possible total switch rate while also resulting in completely fair individual rates. This is observed in Fig. 6 in the form of the black dots overlapping with the blue bars. Note that here, the width of the blue line indicates the total switch rate corresponding to the resource allocation that also maximizes fairness between individual client rates, with the corresponding allocation depicted on the bars. On the other hand, the black line indicates the maximum overall switch rates, which does not always result in a fair distribution of client rates. Second, as noted in [27], the end-to-end rate eventually saturates as the number of multiplexed links in a connection increases. This is observed in Fig. 6 as the total switch rates corresponding to the left-hand side ($k_{\text{total}} =$

IEEE Transactions on Quantum Engineering

36) are only marginally more than those on the right-hand side (k_{total}=24). Third, allocating multiplexed links to lower error-load connections in a switch with limited resources is better since the poor-quality links might consume almost all the resources to significantly increase their individual rates. Therefore, when clients are asymmetrically distributed along the switch ($l_1 \neq l_2$), low error-load connections should have more resources if the maximum total switch rate is the target. However, on the other hand, if the aim is to attain a desired level of individual rate fairness, increasing the number of multiplexed links helps high error-load end-to-end connections to catch up. This is observed in Fig. 6 in the form of the gaps between the blue bars and the black dots, for example in the case corresponding to $l_1 = 0.5$ and $l_2 = 2.5$. Here, the black dot represents a rate distribution that favors the second client over the first.

The relationship between the different $R_{\text{e2e}}^{(i)}$s and R_{s} in (8) and the resource allocations k_i is quite subjective and depends on factors, such as the relative distances between the switch and the clients (l_i). To shed more light on this, we next study a dominant data center example, where the total switch rate is independent of the clients' resource allocation. This should allow us to explore improving fairness without compromising the total switch rate. Such a scenario is encountered, e.g., when the data center is much farther away from the switch than the other clients. This results in the order of magnitude of the total logical error probabilities being dictated by the data center distance, and in turn, the total switch rate R_{s} becoming independent of the other clients' resource budget allocations. For a switch that has three clients situated at distances $\{l_1, l_2, l_3\}$, $l_i \ll 5$ km, and a data center situated at a distance $l_d = 5$ km, Fig. 7 illustrates the fairness of the most equitable distribution of resources. This allocation is determined using both the Euclidean fairness measure as mentioned in (9) as well as the Jain's fairness index (used as $1 - F_{\text{Jain}}$), and we observe that the optimal allocations coincide. Other observations from the figure include the following. With the allocation approaching a more equitable distribution of resources, the fairness measures decrease. The rate fairness improves as the total resource state budget k_{total} increases. The clients farther away from the switch consume more resource states to enhance their end-to-end rates, pushing optimization of (11) toward a more equitable distribution of rates. End-to-end rates are distributed more fairly when clients are at similar distances from the switch. Finally, fair resource allocation in symmetric settings results in an equal end-to-end rate for all clients. For another example and a more detailed analysis of this interesting scenario of dominant clients of the switch, see Appendix B.

V. CONCLUSION AND OUTLOOK

We presented architectural and protocol designs for a GKP-qubit-based quantum switch for entanglement distribution networks. For the proposed switch design, we analyzed the optimal allocation of resources among any number of clients for different client distance arrangements. For a switch connecting two clients, we determined both the optimum positioning of the switch relative to the clients and the optimal allocation of resources such that the switch rate is maximized. For a switch connecting multiple clients, we investigated optimal resource allocation between the different clients for two different goals: 1) maximizing the total switch rate alone and 2) maximizing the total switch rate while also satisfying a constraint on fairness among all the individual rates enabled by the switch. We explained our findings by analysing a data center switch network, where the switch has many users and a data center as its client, and the users want to connect to the data center alone.

As part of our analysis of the data center switch network, we elucidated the scenario of a dominant data center, where the data center is much farther from the switch than clients are from the switch, such that the error rates of the data center elementary links govern the overall switch throughput regardless of the quality of the elementary links associated with the other clients. In this scenario, we observed that entangled resource state allocation between the clients can be tweaked to increase rate fairness without compromising the total switch rate. It is also interesting to consider the flip case of the above scenario, where the data center is close to the switch and no longer the bottleneck, but all the other clients are far away. Because half of the resources are still dedicated to poor-quality elementary links, which now exist with the clients instead of the data center, the total switch rate would remain invariant regardless of the resource allocations of individual clients.

Based on our findings, following is a summary of guidelines to optimally locate and allocate resources of the quantum network helper nodes, i.e., repeaters and switches, to enhance the overall switch rate and also the fairness of rates among clients.

1) To maximize the generation rate of end-to-end entangled links between two clients connected via a single switch, the relative location of the switch should be in the middle of the connection, and the resources should be shared equally. When it comes to an end-to-end connection, symmetric resource allocation always results in a better end-to-end rate, even when clients are at different distances from the switch. This applies to single switch connections and chains of switches (or repeaters) that connect two end-users.
2) The GERM protocol can optimize the switch's entanglement generation throughput across multiple clients and connections. The protocol ensures the highest overall rate by favoring higher quality connections. It achieves this by minimizing the logical error over its top-quality connections. Implementing this protocol can markedly improve the efficiency of entanglement generation, especially in critical settings where high-speed data transfer is essential.

3) To ensure fairness among clients, the switch should prioritize lower quality connections by giving them more entangled resource states to account for their relatively low per-mode rate. This approach will help ensure that individual end-to-end rates are comparable, regardless of connection quality.

We suggest a few possible directions to further this work. First, in describing the optimal allocation in the data center switch network, we mentioned how allocations can be identified toward connections instead of clients, and they turn out to be the same in this case as the clients identify connections. In general multiclient scenarios, connection-based allocation is crucial to ensure that all connections are served. Here, as opposed to the GERM protocol, which is based on global ranking-based matching, a different link-matching protocol heuristic for the switch could be considered, where after assigning the required number of resources to each connection, links are ranked at clients locally within each connection that they are a part of. Such a local-matching protocol would favor fairness of rates across active connections, but each individual end-to-end rate and, naturally, the overall switch rate would be decreased compared to those achieved by the GERM protocol. The local matching heuristic may be studied thoroughly in future works and be of interest to networks where fairness is the top priority. Second, the optimal placement of the switch within an area of an M-node, V-edge graph of network nodes. To start this investigation, we need an easy-to-compute, close expression for the optimum discard window size for arbitrary internodal spacings, which can be established empirically.

Third, while the GERM protocol indeed delivers the maximum rate throughput for a single switch as illustrated by our results, it is an interesting open question to investigate if, in a network of switches, implementing the GERM protocol at each switch also amounts to maximizing the throughput of the switch network, or if there exist other network-wide protocols that can outperform switchwise GERM protocol. Fourth, to address the rate disparity between various quality connections, increasing the number of hard-to-prepare entangled resource states in high-error-load connections only gradually enhances fairness. It ultimately leads to diminishing returns in end-user rates. Exploring a time-shared quantum switch should be interesting, where we group similar quality connections and have the switch cater to the different connection groups at different time steps.

Finally, we note that our work in this article concerns maximizing the sum throughput of the switch network with or without fairness (above a prespecified threshold) between individual rates of the various connections enabled by the switch. The switch does so regardless of the requests (demands) for entanglement across all the other connections the switch serves. It is more similar to connection-oriented, time-, or frequency-division multiplexing-based circuit switching in classical networks, where resources are reserved for the different connections. It would also be interesting to study the dynamic allocation of the entangled state resources driven by entanglement requests to allow for more effective resource-sharing between connections to cater to real demands for entanglement, more similar to packet switching in classical networks. In this regard, the service rate capacity region of a switch that dynamically serves the demands of the different connections for mixed-partite entanglement (bipartite and multipartite entanglement) has been established, albeit for the case of ideal elementary entanglement links that are Bell states [35], [36], [37]. It would be interesting to determine analogous capacity regions for the multiplexed GKP qubit-based elementary links considered in this work.

APPENDIX A
POSTSELECTION TEST IN RESOURCE STATE GENERATION

In a quantum network enabled by GKP qubit entanglement, the quantum helper nodes (quantum repeaters and quantum switches) would connect to one another by generating elementary entanglement links using entangled resource states and then swapping entanglement. This article follows a multiplexed architecture, where overall k_{total} entangled resource states are generated at each helper node, e.g., see Fig. 2. Each resource state is a Bell pair between one physical GKP state and one logical GKP state encoded in a [[7,1,3]] Steane code. In total, each resource state consists of eight physical GKP states. Each helper node station keeps the [[7,1,3]] logical GKP qubit part of the Bell state, referred to as inner leaf qubits, and sends the remaining physical GKP qubit, which is the other part of the Bell state, toward the other party as outer leaves. This choice is because the inner leaf qubits go through a fiber two times longer than the outer leaf qubits. So, an additional error-correcting layer (GKP + Steane code) better protects stored quantum information from noise and photon loss. We will not go through the procedure of resource state generation; one can examine [27] for a detailed discussion on the entangled eight GKP qubit resource state cube and its generation. However, below, we briefly discuss the role of a postselection test and associated discard window in the procedure.

The ideal GKP qubit computational basis states have support in the q-quadrature basis at integer multiples of $\sqrt{\pi}$ alone (odd multiples alone for the $|1\rangle$ state and even multiples alone for the $|0\rangle$ state). Their realistic finite squeezed versions include a support that is spread out around the same quadrature values. When a general superposition state of the GKP qubit is measured in the computational basis, i.e., using homodyne detection along the q quadrature, it is thus possible to observe an outcome that is from anywhere along the superposed supports of the $|0\rangle$ and $|1\rangle$ bases states in the q-quadrature basis, i.e., around all integer multiples of $\sqrt{\pi}$. Values in the superposed support close to odd integer multiples of $\sqrt{\pi}/2$ create an ambiguity in inferring the logical outcome of the measurement as they may have arisen from either of their neighboring multiples of $\sqrt{\pi}$, i.e., the supports of the $|0\rangle$ and $|1\rangle$ bases states, leading to a possible error in inference. The probability of the inference error can be

$K_{total} = 12$	$Switch\ Rate\ (L_1, L_2, L_3, L_4)$				
[K(1 - 4), K(2 - 4), K(3 - 4)]	R_s(0.5,0.5,0.5,5)	R_s(1,1,1,5)	R_s(0.5,0.5,2,5)	R_s(2,2,0.5,5)	R_s(0.5,1,2,5)
1. [8, 2, 2]	0.582250398	0.585796779	0.58122263	0.575605979	0.57457593
2. [6, 4, 2]	0.578506658	0.579519569	0.577277264	0.568637853	0.579762239
3. [4, 6, 2]	0.585683684	0.581139698	0.584953492	0.570669917	0.580171196
4. [2, 8, 2]	0.574250283	0.576180082	0.573233996	0.566279299	0.575103146
5. [6, 2, 4]	0.585791643	0.574597787	0.582933276	0.566318072	0.573108314
6. [4, 4, 4]	0.583595629	0.58124416	0.580705056	0.573341417	0.570137304
7. [2, 6, 4]	0.586137438	0.581336977	0.583152413	0.57396327	0.581496064
8. [4, 2, 6]	0.57799734	0.570113201	0.573480705	0.564128132	0.573352431
9. [2, 4, 6]	0.577624383	0.575699416	0.572558423	0.570345596	0.577952207
10. [2, 2, 8]	0.584622665	0.577827255	0.577044077	0.574269469	0.574829168
Mean	0.581646012	0.578345492	0.578656133	0.5703559	0.5760488
Standard Deviation	0.004225943	0.004420083	0.004556057	0.003937907	0.003643765
Maximum	0.586137438	0.585796779	0.584953492	0.575605979	0.581496064
Minimum	0.574250283	0.570113201	0.572558423	0.564128132	0.570137304

$K_{total} = 24$	$Switch\ Rate\ (L_1, L_2, L_3, L_4)$				
[K(1 - 4), K(2 - 4), K(3 - 4)]	R_s(0.5,0.5,0.5,5)	R_s(1,1,1,5)	R_s(0.5,0.5,2,5)	R_s(2,2,0.5,5)	R_s(0.5,1,2,5)
1. [20, 2, 2]	0.598664876	0.627633674	0.600134769	0.619892581	0.604446843
2. [18, 4, 2]	0.60023809	0.626938677	0.601725892	0.619507244	0.604479366
3. [16, 6, 2]	0.604842631	0.630013468	0.606262353	0.622331015	0.602403782
4. [14, 8, 2]	0.598387585	0.630232165	0.599821853	0.622570649	0.60301265
5. [12, 10, 2]	0.599900515	0.629922893	0.601355748	0.622221152	0.602825801
6. [10, 12, 2]	0.605432052	0.626958063	0.606919969	0.619618921	0.601055395
...			**...**		
53. [4, 2, 18]	0.599804238	0.630716928	0.59880028	0.626459829	0.59836367
54. [2, 4, 18]	0.601293263	0.629008164	0.600371843	0.624681802	0.596486703
55. [2, 2, 20]	0.600680989	0.630316969	0.599485622	0.626412436	0.598812452
Mean	0.601819837	0.627954998	0.602540478	0.621893163	0.600377585
Standard Deviation	0.001988607	0.002307038	0.002221231	0.00248986	0.002407584
Maximum	0.606054404	0.631944683	0.606919969	0.627017378	0.606021269
Minimum	0.598069181	0.622677192	0.597586135	0.616091987	0.595955352

$K_{total} = 36$	$Switch\ Rate\ (L_1, L_2, L_3, L_4)$				
[K(1 - 4), K(2 - 4), K(3 - 4)]	R_s(0.5,0.5,0.5,5)	R_s(1,1,1,5)	R_s(0.5,0.5,2,5)	R_s(2,2,0.5,5)	R_s(0.5,1,2,5)
1. [32, 2, 2]	0.607029829	0.622974221	0.611531524	0.617563363	0.613674825
2. [30, 4, 2]	0.606343592	0.624450494	0.610800689	0.619488309	0.615158231
3. [28, 6, 2]	0.605188662	0.626669018	0.609654829	0.621151688	0.614369943
4. [26, 8, 2]	0.60849695	0.626093739	0.612944095	0.620482263	0.611222453
5. [24, 10, 2]	0.606131865	0.629167885	0.610499588	0.623589682	0.611088836
6. [22, 12, 2]	0.608935711	0.628824473	0.613459557	0.62341544	0.61303428
...			**...**		
134. [4, 2, 30]	0.608067227	0.627889488	0.609246736	0.626513599	0.610084138
135. [2, 4, 30]	0.60789435	0.627469956	0.609153512	0.625981773	0.609106193
136. [2, 2, 32]	0.605059879	0.625593389	0.605721193	0.624186077	0.608070011
Mean	0.607243792	0.626418211	0.610867867	0.622754429	0.61139839
Standard Deviation	0.001709887	0.0019567	0.00190737	0.002299209	0.001765803
Maximum	0.612293236	0.630590013	0.616566908	0.628244333	0.616327028
Minimum	0.603048711	0.621557112	0.604698089	0.617380554	0.606830002

FIGURE 8 **Comparison of total switch rates R_s across all the possible resource allocations [$k(1 \leftrightarrow 4), k(2 \leftrightarrow 4), k(3 \leftrightarrow 4)$] between different connections for a switch network with a data center and three other clients, with a total number of resource states $k_{\text{total}} = \sum_{i=1}^{3} k(i \leftrightarrow 4)$ available at the switch. Each table considered a variety of clients' settings (l_1, l_2, l_3, l_4). As the total number of resource states (k_{total}) increases, the overall switch rate (R_s) grows as well. The identical feature between these tables is the R_s independence from clients' allocation. The reason is behind how we defined a data center or a user ($l_i \ll l_n$, $l_i < 3$km) so that the data center controls the overall switch logical error.**

reduced by a postselection test, where outcomes in a discard window around the odd integer multiples of $\sqrt{\pi}/2$ are simply discarded. For a discard window of size ν symmetrically placed about the odd multiples of $\sqrt{\pi}/2$, the probability of successfully passing the postselection test without incurring a logical error is given by

$$E_0(\nu, \sigma^2) = \sum_{m\in\mathbb{Z}} \int_{\frac{4m-1}{2}\sqrt{\pi}+\nu}^{\frac{4m+1}{2}\sqrt{\pi}-\nu} \frac{e^{-x^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}}\, dx. \qquad (12)$$

Increasing the discard window size ν suppresses the errors at the expense of increased overheads in terms of the primer resources required to generate the necessary number of the entangled resource states. We need to increase the discard window so that the errors from resource state generation are at the order of magnitude of the inner leaf qubit storage and communication channel errors. The error from resource generation being any better does not help anyway. The most suitable value for the discard window is thus related to the internodal spacing L across which the entangled resource state is used for elementary entanglement generation. For a large (small) internodal spacing, we face a large (small) communication channel error, and so we need the inner leaf qubit storage errors to be at large (small). This means the discard window size should be small (large). For values of $L \in \{0.5, 1, 2, 2.5, 5\}$ km, reasonable choices for the corresponding discard window sizes ν were numerically found in [27] to be $\{7, 6, 5, 4, 3\} \times \sqrt{\pi}/20$, respectively. Optimal values of ν can be similarly determined for other values of L.

APPENDIX B
RESOURCE ALLOCATION AND SWITCH RATE

In this appendix, we discuss the case where one of the clients of a switch is dominant, and how in such a case, changing the other clients' resource allocation scarcely affects the overall switch rate R_s. We numerically simulate a switch with a total number of entangled resource states k_{total}, connecting $n$clients to 1 data center in each time step. Acceptable connections are from any clients to the data center. Accordingly, half of the resource states are assigned to the data center ($k_{\text{total}}/2$). The question is, how will the total switch rate (R_s) be affected by changing the allocation of the remaining resource states ($k_{\text{total}}/2$) shared between the clients? Fig. 8 tries to address this question over three storylines.

The table in Fig. 8 gives the total switch rate R_s connecting three clients indexed as $\{1, 2, 3\}$ and a data center $\{4\}$ over a variety of distances (distance values and k_{total} are assumed to be known beforehand). The allowable connections are (client-1, data center), (client-2, data center), and (client-3, data center). The first three columns of the table represent every possible resource allocation, where allocations are shown per connection instead of per client. For example, $k(i \leftrightarrow 4)$ is the total number of entangled resource states allocated to the connection of client-i and the data center such that $\sum_{i=1}^{3} k(i \leftrightarrow 4) = k_{\text{total}}$. Upon examining the table, we find that for a choice of (l_1, l_2, l_3, l_4) across the column, through all the possible allocations of the k_{total} resources (sharing between the connections), the values of R_s are concentrated around the mean. Standard deviations are found to be $\ll 0.01$, with the maximum and minimum being close to the average. Since the data center is the dominant client and $l_i \in \{0.5, 1, 2\}$ km, for $i \in \{1, 2, 3\}$, the switch sees all the other clients as a single client, requesting to connect to the data center.

Not all switches have a dominant client (clients can be in the same order), but if they do, from the switch perspective, the dominant client governs the error. Let us study this from a simpler view. Two clients, client-1 and client-2, with resource states shared equally between them ($k_1 = k_2 = 10$), are connecting via the proposed repeater in Fig. 2. We simulate two sets of distances, namely, $\{(l_1 = 5, l_2 = 5)$ km and $(l_1 = 0.5, l_2 = 5)\}$ km. The results are: $R_s(l_1 = 5, l_2 = 5)$ km $\approx R_s(l_1 = 0.5, l_2 = 5)$ km. This means that when one of the two clients controls errors, there is no benefit in improving the other client. That is why assigning most resources to the client-i-data center connection where client-i is the closest client to the switch will not favorably increase the switch rate. In our analysis, we assign each connection at least two links, so assigning all the resources k_{total} to a single connection is prohibited.

REFERENCES

[1] M. Azari, P. Polakos, and K. P. Seshadreesan, "A GKP qubit-based all-photonic quantum switch," in *Proc. 2024-IEEE Int. Conf. Commun.*, 2024, pp. 503–508, doi: 10.1109/ICC51166.2024.10622457.

[2] J. P. Dowling and G. J. Milburn, "Quantum technology: The second quantum revolution," *Philos. Trans. A. Math. Phys. Eng. Sci.*, vol. 361, no. 1809, pp. 1655–1674, Aug. 2003, doi: 10.1098/rsta.2003.1227.

[3] A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, and J. M. Gambetta, "Validating quantum computers using randomized model circuits," *Phys. Rev. A*, vol. 100, Sep. 2019, Art. no. 032328, doi: 10.1103/PhysRevA.100.032328.

[4] K. Zhang et al., "Modular quantum computation in a trapped ion system," *Nat. Commun.*, vol. 10, no. 1, Oct. 2019, Art. no. 4692, doi: 10.48550/arXiv.1907.12171.

[5] C. Monroe et al., "Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects," *Phys. Rev. A*, vol. 89, no. 2, Feb. 2014, Art. no. 022317, doi: 10.1103/PhysRevA.89.022317.

[6] S. Stein et al., "HetArch: Heterogeneous microarchitectures for superconducting quantum systems," in *Proc. 56th IEEE/ACM Int. Symp. Microarchitecture*, 2023, pp. 539–554. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10411374

[7] C. Zhou et al., "A modular quantum computer based on a quantum state router," 2022, doi: 10.21203/rs.3.rs-1547284/v1.

[8] K. Nemoto et al., "Photonic quantum networks formed from NV-centers," *Sci. Rep.*, vol. 6, no. 1, May 2016, Art. no. 26284, doi: 10.1038/srep26284.

[9] J. C. Boschero, N. M. Neumann, W. van der Schoot, T. Sijpesteijn, and R. Wezeman, "Distributed quantum computing: Applications and challenges," 2024, doi: 10.48550/arXiv.2410.00609.

[10] J. Liu, C. T. Hann, and L. Jiang, "Data centers with quantum random access memory and quantum networks," *Phys. Rev. A*, vol. 108, no. 3, Sep. 2023, Art. no. 032610, doi: 10.1103/PhysRevA.108.032610.

[11] J. F. Fitzsimons, "Private quantum computation: An introduction to blind quantum computing and related protocols," *NPJ Quantum Inf.*, vol. 3, no. 1, pp. 1–11, Jun. 2017, doi: 10.1038/s41534-017-0025-3.

[12] M. Mehic et al., "Quantum key distribution: A networking perspective," *ACM Comput. Surv.*, vol. 53, no. 5, pp. 1–41, Sep. 2020, doi: 10.1145/3402192.

[13] B. K. Malia, Y. Wu, J. Martínez-Rincón, and M. A. Kasevich, "Distributed quantum sensing with mode-entangled spin-squeezed atomic states," *Nature*, vol. 612, no. 7941, pp. 661–665, Dec. 2022, doi: 10.1038/s41586-022-05363-z.

[14] Z. Zhang and Q. Zhuang, "Distributed quantum sensing," *Quantum Sci. Technol.*, vol. 6, no. 4, Jul. 2021, Art. no. 043001, doi: 10.1088/2058-9565/abd4c3.

[15] X. Guo et al., "Distributed quantum sensing in a continuous-variable entangled network," *Nat. Phys.*, vol. 16, no. 3, pp. 281–284, Dec. 2019, doi: 10.1038/s41567-019-0743-x.

[16] S. Slussarenko and G. J. Pryde, "Photonic quantum information processing: A concise review," *Appl. Phys. Rev.*, vol. 6, no. 4, 2019, Art. no. 041303, doi: 10.1063/1.5115814.

[17] F. Flamini, N. Spagnolo, and F. Sciarrino, "Photonic quantum information processing: A review," *Rep. Prog. Phys.*, vol. 82, no. 1, Jan. 2019, Art. no. 016001, doi: 10.1088/1361-6633/aad5b2.

[18] C. Weedbrook et al., "Gaussian quantum information," *Rev. Mod. Phys.*, vol. 84, no. 2, pp. 621–669, May 2012, doi: 10.48550/arXiv.1110.3234.

[19] D. Gottesman, A. Kitaev, and J. Preskill, "Encoding a qubit in an oscillator," *Phys. Rev. A*, vol. 64, no. 1, Jun. 2001, Art. no. 012310, doi: 10.1103/PhysRevA.64.012310.

[20] K. Noh, V. V. Albert, and L. Jiang, "Quantum capacity bounds of Gaussian thermal loss channels and achievable rates with Gottesman-Kitaev-Preskill codes," *IEEE Trans. Inf. Theory*, vol. 65, no. 4, pp. 2563–2582, Apr. 2019, doi: 10.1109/TIT.2018.2873764.

[21] W. J. Munro, K. Azuma, K. Tamaki, and K. Nemoto, "Inside quantum repeaters," *IEEE J. Sel. Top. Quantum Electron.*, vol. 21, no. 3, pp. 78–90, May/Jun. 2015, doi: 10.1109/JSTQE.2015.2392076.

[22] Y. Lee, E. Bersin, A. Dahlberg, S. Wehner, and D. Englund, "A quantum router architecture for high-fidelity entanglement flows in quantum networks," *NPJ Quantum Inf.*, vol. 8, no. 1, 2022, Art. no. 75, doi: 10.48550/arXiv.2005.01852.

[23] G. Vardoyan, P. Nain, S. Guha, and D. Towsley, "On the capacity region of bipartite and tripartite entanglement switching," *ACM Trans. Model. Perform. Eval. Comput. Syst.*, vol. 8, no. 1-2, pp. 1–18, 2023, doi: 10.1145/357180.

[24] I. Tillman, T. Vasantam, and K. P. Seshadreesan, "A continuous variable quantum switch," in *Proc. 2022 IEEE Int. Conf. Quantum Comput. Eng.*, 2022, pp. 365–371, doi: 10.1109/QCE53715.2022.00057.

[25] S. Pirandola, R. Laurenza, C. Ottaviani, and L. Banchi, "Fundamental limits of repeaterless quantum communications," *Nat. Commun.*, vol. 8, Apr. 2017, Art. no. 15043, doi: 10.1038/ncomms15043.

[26] F. Schmidt, D. Miller, and P. van Loock, "Error-corrected quantum repeaters with GKP qudits," *Phys. Rev. A*, vol. 109, 2023, Art. no. 042427, doi: 10.1103/PhysRevA.109.042427.

[27] F. Rozpdek, K. P. Seshadreesan, P. Polakos, L. Jiang, and S. Guha, "All-photonic Gottesman-Kitaev-Preskill–qubit repeater using analog-information-assisted multiplexed entanglement ranking," *Phys. Rev. Res.*, vol. 5, Oct. 2023, Art. no. 043056, doi: 10.1103/PhysRevResearch.5.043056.

[28] K. Fukui, R. N. Alexander, and P. van Loock, "All-optical long-distance quantum communication with Gottesman-Kitaev-Preskill qubits," *Phys. Rev. Res.*, vol. 3, no. 3, Aug. 2021, Art. no. 033118, doi: 10.1103/PhysRevResearch.3.033118.

[29] F. Rozpedek, K. Noh, Q. Xu, S. Guha, and L. Jiang, "Quantum repeaters based on concatenated bosonic and discrete-variable quantum codes," *NPJ Quantum Inf.*, vol. 7, no. 1, pp. 1–12, Jun. 2021, doi: 10.48550/arXiv.2011.15076.

[30] I. Tzitrin, J. E. Bourassa, N. C. Menicucci, and K. K. Sabapathy, "Progress towards practical qubit computation using approximate Gottesman-Kitaev-Preskill codes," *Phys. Rev. A*, vol. 101, no. 3, Mar. 2020, Art. no. 032315, doi: 10.1103/PhysRevA.101.032315.

[31] D. Bruß, "Optimal eavesdropping in quantum cryptography with six states," *Phys. Rev. Lett.*, vol. 81, pp. 3018–3021, Oct. 1998, doi: 10.1103/PhysRevLett.81.3018.

[32] S. J. Devitt, W. J. Munro, and K. Nemoto, "Quantum error correction for beginners," *Rep. Prog. Phys.*, vol. 76, no. 7, Jul. 2013, Art. no. 076001, doi: 10.1088/0034-4885/76/7/076001.

[33] P. Dhara, N. M. Linke, E. Waks, S. Guha, and K. P. Seshadreesan, "Multiplexed quantum repeaters based on dual-species trapped-ion systems," *Phys. Rev. A*, vol. 105, Feb. 2022, Art. no. 022623, doi: 10.1103/PhysRevA.105.022623.

[34] R. K. Jain et al., "A quantitative measure of fairness and discrimination," *Eastern Res. Lab. Digit. Equip. Corporation*, Hudson, MA, USA, vol. 21, 1984, Art. no. 1, doi: 10.48550/arXiv.cs/9809099.

[35] I. Tillman, T. Vasantam, D. Towsley, and K. P. Seshadreesan, "Calculating the capacity region of a quantum switch," 2024, *arXiv:2404.18818*, doi: 10.48550/arXiv.2404.18818.

[36] I. Tillman, T. Vasantam, and K. P. Seshadreesan, "A continuous variable quantum switch," in *Proc. 2022 IEEE Int. Conf. Quantum Comput. Eng.*, 2022, pp. 365–371, doi: 10.1109/QCE53715.2022.00057.

[37] T. Vasantam and D. Towsley, "Stability analysis of a quantum network with max-weight scheduling," 2021, *arXiv:2106.00831*, doi: 10.48550/arXiv.2106.00831.