Quantum Circuit Switching with One-Way Repeaters in Star Networks

Iñesta, Alvaro G.; Choi, Hyeongrak; Englund, Dirk; Wehner, Stephanie

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Quantum Circuit Switching
with One-Way Repeaters in Star Networks

Álvaro G. Iñesta*[†][‡], Hyeongrak Choi[§], Dirk Englund[§], and Stephanie Wehner*[†][‡]

*QuTech, Delft University of Technology, Lorentzweg 1, 2628 CJ Delft, The Netherlands
[†]Quantum Computer Science, EEMCS, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
[‡]Kavli Institute of Nanoscience, Delft University of Technology, Lorentzweg 1, 2628 CJ Delft, The Netherlands
[§]Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA

Abstract—Distributing quantum states reliably among distant locations is a key challenge in the field of quantum networks. One-way quantum networks address this by using one-way communication and quantum error correction. Here, we analyze quantum circuit switching as a protocol to distribute quantum states in one-way quantum networks. In quantum circuit switching, pairs of users can request the delivery of multiple quantum states from one user to the other. After waiting for approval from the network, the states can be distributed either sequentially, forwarding one at a time along a path of quantum repeaters, or in parallel, sending batches of quantum states from repeater to repeater. Since repeaters can only forward a finite number of quantum states at a time, a pivotal question arises: is it advantageous to send them sequentially (allowing for multiple requests simultaneously) or in parallel (reducing processing time but handling only one request at a time)? We compare both approaches in a quantum network with a star topology. Using tools from queuing theory, we show that requests are met at a higher rate when packets are distributed in parallel, although sequential distribution can generally provide service to a larger number of users simultaneously. We also show that using a large number of quantum repeaters to combat channel losses limits the maximum distance between users, as each repeater introduces additional processing delays. These findings provide insight into the design of protocols for distributing quantum states in one-way quantum networks.

Index Terms—circuit switching, resource allocation, one-way, quantum network, quantum repeaters

I. INTRODUCTION

In the field of quantum networking, the efficient delivery of quantum information between distant users remains a central challenge [1]. A common strategy is to employ quantum repeaters to facilitate quantum communication among remote network nodes [2], [3]. Many repeater architectures require two-way communication and long-lived quantum memories, as they rely on the distribution of entanglement via heralded generation [4], [5] and entanglement swaps [6]–[8]. Conversely, the so-called *third-generation quantum repeaters* use quantum error correction to send quantum data using only one-way communication [2], [9]–[11]. Here, we focus on one-way

quantum networks, which employ third-generation repeaters to distribute quantum states.

In one-way quantum networks, each quantum state that must be delivered to a remote location (e.g., a data qubit or half of an entangled pair) is encoded and sent over the network as a *quantum data packet*, which consists of the encoded state and some additional metadata (such as the packet destination) [12]. In classical networks, data packets are commonly distributed across the network according to a packet switching or a circuit switching protocol [13]. In packet switching [14], [15], users can send data packets at will, which are then forwarded from one router to the next one whenever possible. Routers may store the packets until they are able to forward them. In circuit switching (see, e.g., [16]), users first request service to the network. Later, the network reserves one or multiple paths (also called circuits) where routers reserve dedicated resources to meet that request, and there is in principle no need to store data in intermediate routers. In classical networks, packet switching has found more success than circuit switching since it makes better use of the resources in large networks with conflicting routes and high traffic loads. In quantum networks, quantum data packets can also be distributed according to a packet switching [12] or a circuit switching protocol. Quantum circuit switching has been studied in the context of two-way quantum repeaters [17], but not in the context of one-way quantum networks, to the best of our knowledge.

The main challenge when designing a one-way protocol to distribute quantum states is decoherence: quantum states have a limited lifetime and cannot be stored indefinitely in intermediate quantum repeaters. This means that the protocol must ensure rapid delivery of the quantum data packets. Additionally, some multi-party applications may need to consume multiple quantum states simultaneously (e.g., to implement verifiable blind quantum computations, multiple qubits must be sent from a client to a server, where they must coexist while some operations are performed [18], [19]). In those cases, the quantum states must be distributed within some time window – otherwise, by the time the last state is distributed, the quality of the first state would have decayed too much due to decoherence. Consequently, one-way protocols must prevent large variations in the time between the delivery of successive quantum data packets to ensure all of them are delivered
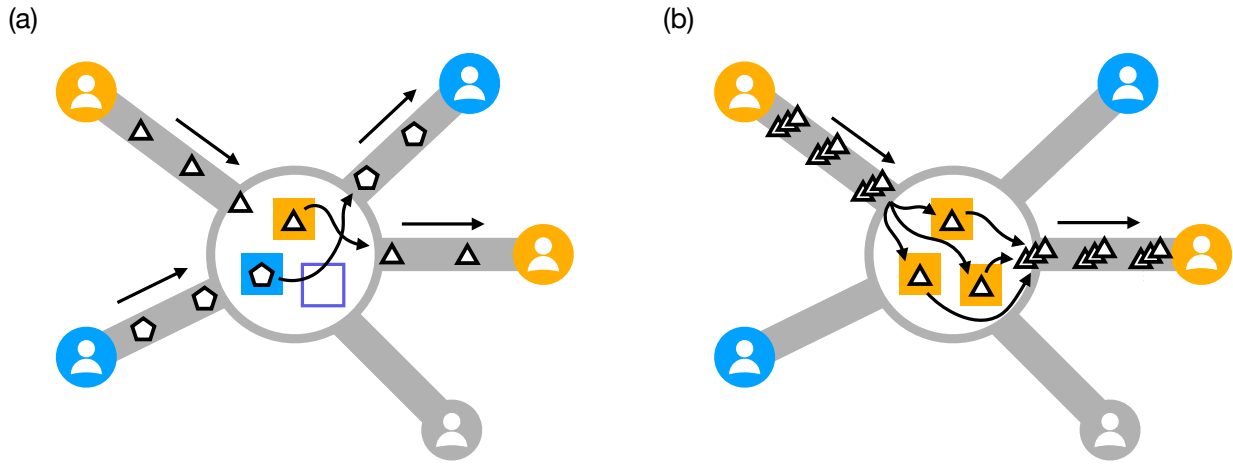
Fig. 1. **In QCS, packets can be distributed (a) sequentially or (b) in parallel.** Illustration of a QCS scheme with (a) sequential and (b) parallel distribution of packets, in a star network with five users and a single central repeater (big circle). The repeater contains $k = 3$ forwarding stations (squares), therefore it can forward at most $k = 3$ quantum data packets (triangles and pentagons) at a time. In sequential distribution of packets, a single forwarding station is reserved to meet each of the requests submitted by the pairs of users in orange and blue. In parallel distribution, all three stations are used in parallel to meet a single request at a time.

within a specific time window. We expect the pre-allocation of resources in circuit switching to enable the protocol to meet these timing constraints required to successfully distribute quantum states over a network, regardless of network traffic. In this work, we introduce *quantum circuit switching* (QCS) for one-way quantum networks, and we investigate how to allocate resources for each request to distribute quantum states.

Each quantum repeater has limited capabilities, namely, it can only forward a finite number of packets, k, at a time (see Figure 1). We then say that the repeater has k *forwarding stations*, which operate independently and can forward one packet at a time. When multiple requests need to use the same repeater, the circuit switching protocol can either (i) reserve one forwarding station per request and process up to k requests simultaneously (Figure 1a), or (ii) reserve all k forwarding stations to meet one request at a time (Figure 1b). For simplicity, we omit the intermediate case where more than one but less than k stations are reserved for each request. In the first approach, quantum data packets are distributed *sequentially*, while in the second approach packets are distributed in *parallel* (requiring some form of frequency multiplexing).

The performance of both approaches can be measured using the *mean sojourn time*, which is the time passed since the request is submitted until it is met – this includes some *waiting time* before the request is processed and the *service time* needed to successfully distribute all the quantum packets. Packets can be lost while traveling from node to node, so the total number of packets that need to be sent to successfully deliver a given number of packets is a priori unknown. As a consequence, the sojourn, waiting, and service times are random variables. We expect shorter service times when packets are distributed in parallel rather than sequentially. However, parallel distribution can only serve one request at a time, and this might entail large waiting times. A natural question arises:

in what situations is it actually advantageous to distribute quantum packets in parallel instead of sequentially? That is, *when does parallel distribution provide shorter mean sojourn times?* Addressing this resource-allocation question early on is crucial for the design of efficient protocols for distributing quantum states within one-way quantum networks. Here, we answer the question assuming a star network: a simplified topology where all users are connected via a number of quantum repeaters to a central repeater (similar to a quantum switch architecture [20], [21]).

Our main contributions are the following:

- We formalize the concept of quantum circuit switching in the context of one-way quantum networks.
- We provide analytical tools to compute the mean sojourn time of a QCS scheme, using techniques from queuing theory.
- We compare QCS schemes with sequential and parallel distribution of packets in a star network, and provide heuristics for the design of QCS protocols in more complex networks.

Our main findings are the following:

- In star networks with sequential distribution of packets, we can increase the number of forwarding stations k to increase the number of users supported by the network (i.e., the maximum number of users that allow requests to be met within a finite amount of time), which scales as $\sim \sqrt{k}$. This is not possible in parallel distribution: the number of users is capped and cannot be increased indefinitely by adding more forwarding stations per repeater.
- Parallel distribution generally provides smaller mean sojourn times than sequential distribution.
- There exists a trade-off between the number of users and the physical distances between them: too many users and too long distances overload the system and yield infinite

1858

Physical topology (star network)	
u	Number of users
L	Distance between each user and the central repeater
N	Number of repeaters between each user and the central repeater

Hardware	
k	Number of forwarding stations per repeater
t_{fwd}	Forwarding time per forwarding station and quantum data packet
p	Probability of successful packet delivery from user to user
c	Speed of light in the physical channels

Requests	
n	Number of quantum data packets per request
w	Request time window
λ_0	Request submission rate per pair of users

Quantum Circuit Switching	
m	Number of packets of the same request distributed concurrently ($m = 1$ for sequential distribution and $m = k$ for parallel distribution)

waiting times.
- When there is a large number of users, adding more intermediate repeaters to combat channel losses and minimize the probability of packet loss does not allow the users to be located further away. In fact, adding more repeaters limits the size of the network to smaller scales, since each repeater introduces additional delays.

In Section II, we explain the problem setup: the quantum network and quantum data packets model, the requests model, and the quantum circuit switching scheme. In III, we present our main results: we analyze the operating regimes of both packet distribution strategies and compare their performance. We do this in a variety of scenarios, considering different quantum repeater architectures based on the financial budget available. Lastly, in IV, we discuss the limitations and the implications of our work.

II. PROBLEM SETUP

In this Section, we present the problem setup. In II-A, we discuss our model of quantum network and quantum data packets. In II-B, we explain how the nodes submit requests and how to measure the performance of the network in meeting those requests. Lastly, we formalize the concept of quantum circuit switching in II-C. We provide a summary of all the parameters introduced in this Section in Table I.

A. Quantum networks and quantum data packets

We consider a quantum network with *third-generation quantum repeaters* [2], [10], [11], [22]. These type of repeaters use quantum error-correction to distribute quantum information using one-way communication, as opposed to first- and second-generation repeaters, which use heralded entanglement generation and two-way communication [22]. To forward quantum data, a repeater must first decode an incoming logical state (generally shared as a collection of photons) and correct any errors, then reencode the state again into multiple physical qubits, and finally send the encoded state to the next repeater.

We make the following assumptions about the quantum network:

- We consider a quantum network with a *star topology*, where u users are connected to a central repeater (see Figure 2). This is the simplest case of network where paths between users intersect, leading to shared resources and routing conflicts.
- For simplicity, we assume that all users are at the same physical distance L from the central repeater (see Figure 2). Moreover, there are N repeaters between each user and the central repeater. All repeaters are placed at distance $L_0 = L/(N + 1)$ from each other.
- Users can encode and decode quantum data packets, according to some quantum code. A *quantum data packet* consists of a logical qubit and a classical header containing some metadata, as proposed in ref. [12]. The logical qubit consists of multiple physical qubits, generally in the form of photons. The classical header provides relevant information about the routing of the data packet (e.g., the destination of the packet).
- Each repeater has k *forwarding stations* that can decode and encode according to the same quantum code. Each station can receive/send physical qubits from/to any node. Moreover, each station can only forward (i.e., decode, reencode, and send) one quantum data packet at a time, which takes time t_{fwd}. After forwarding a packet, the station can be immediately used to forward another packet from any source node to any destination node (i.e., there is no downtime).
- Physical qubits can suffer from noise in the encoding and decoding circuits and in the physical channels. We assume that each packet can be successfully decoded at destination with probability p (this includes the case in which the encoded qubits suffered no errors and also the case in which they suffered a correctable number of errors). With probability $1-p$, a number of uncorrectable errors will be detected at destination and the packet will be discarded. We consider the probability of a logical error being unnoticed to be negligible, since we assume the main source of errors to be photon loss in the channels, which can be detected.

B. Requests and performance measures

In many quantum network applications, multiple copies of a quantum state need to be distributed over a short interval of time. For example, verifiable blind quantum computing requires multiple qubits to be sent from a client to a server, where they must coexist while some operations are performed [18], [19]. Another example is verifiable quantum secret sharing [23], where the parties involved want to verify that the dealer successfully distributed a quantum secret among them. This verification requires the dealer to distribute multiple ancillary quantum states to each party that will be consumed
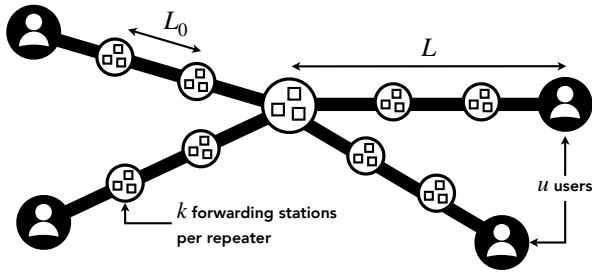
Fig. 2. **Illustration of a star quantum network.** u users are at distance L from a central repeater. There are N repeaters between each user and the central repeater ($N = 2$ in the figure), and each of them has k forwarding stations (squares). The spacing between adjacent nodes is $L_0 = L/(N+1)$.

simultaneously (note that other proposals perform the verification stage sequentially [24]). In these examples, a number of quantum states must be sent from one network node to another over a short interval of time due to decoherence – otherwise, when the last state is distributed, the quality of the state that was distributed first would have decayed too much and would not be useful anymore.

With this type of general application in mind, we define an (n, w)-*request* as request for n quantum data packets to be distributed among two users within time w. Each packet contains a copy of the same quantum state.

We make the following additional assumptions about the requests:

- To simplify the analysis, we assume that all pairs of users make requests with the same parameters n and w.
- Each pair of nodes submits requests following a Poisson process with rate λ_0. The total *request submission rate* is then

$$\lambda = \binom{u}{2}\lambda_0 = \frac{u(u-1)}{2}\lambda_0, \tag{1}$$

where u is the number of users.

The performance of protocols for quantum state distribution is typically measured with the quality of the distributed states and/or the distribution rate (see, e.g., [25]–[28]). In our model, quantum data packets are either successfully distributed or a failure flag is raised, and therefore there is no need to study the quality of the state. The main quantity of interest in our problem is therefore the rate at which (n, w)-requests are met. In particular, to measure the performance of the protocol, we propose the *mean sojourn time*: the average time passed since a request is submitted until it is met (i.e., until the n-th quantum packet is successfully delivered within the time window w). The sojourn time of a request can be computed as the sum of two main contributions:

- *Waiting time* (T_{wait}): time since the request is submitted until some forwarding stations are available for the request. The waiting time depends on the number of requests in the queue at a given time and is a random variable.

- *Service time* (T_{service}): time since the stations are available for the request until the request is met. Since packets can be lost, the service time is a random variable.

In particular, we are interested in the mean sojourn time, which can be computed as

$$\mathbb{E}\left[T_{\text{sojourn}}\right] = \mathbb{E}\left[T_{\text{wait}}\right] + \mathbb{E}\left[T_{\text{service}}\right]. \tag{2}$$

In this work, we implicitly refer to the steady-state mean values when we discuss the mean sojourn time, the mean waiting time, and the mean service time. The steady state corresponds to the status of the system where mean values have reached an equilibrium following a transient period. In the calculation of the sojourn time, one could also add some control time, which would include the time required to submit a request. We can assume this control time is independent of the choice of protocol for quantum state distribution, and therefore we leave it outside of our analysis (if we assume the control time is constant, our results would just shift towards larger sojourn times).

We now summarize how to compute each term from (2):

- To compute the mean waiting time, we model the system as an M/G/s queue – this is a queue model where request arrivals follow a Poisson distribution and therefore are Markovian (M), the service time follows a general distribution (G), and a maximum of s requests can be processed simultaneously. This formulation allows us to derive analytical solutions and approximations for the mean waiting time in a variety of situations, although in some cases Monte Carlo methods are required. See the Appendix for further details.

- The service time distribution is hard to compute analytically. The reason is that calculating the number of packets that need to be sent until n are successfully received over a time window w is a challenging task [19]. When packets are sent in batches of m and each packet can be lost with probability p, the number of batches until success is denoted by the random variable $B_{n,w,p,m}$. The mean service time can be computed in terms of the mean value of $B_{n,w,p,m}$:

$$\mathbb{E}\left[T_{\text{service}}\right] = \frac{2L}{c} + t_{\text{fwd}}(2N+1)\mathbb{E}\left[B_{n,w,p,m}\right], \tag{3}$$

where c is the speed of light in the communication channels. The first term corresponds to the time required for a packet (or a batch of packets) to travel from one user to another. The second term corresponds to the delay introduced by the forwarding stations: each station takes time t_{fwd} to forward a packet, each packet is forwarded at $2N+1$ intermediate repeaters, and a total of $\mathbb{E}\left[B_{n,w,p,m}\right]$ batches of packets must be sent on average to meet the request. In the Appendix, we explain how to compute the probability distribution of $B_{n,w,p,m}$ (in some cases, numerical approximations are required). Note that $m \leq k$, since repeaters can only forward at most k packets simultaneously.

1860

Another relevant quantity is the *load* of the system, which is the ratio between the request submission rate, $\lambda_0 u(u - 1)/2$, and the total rate at which requests are serviced, $k/(m\mathbb{E}[T_{\text{service}}])$, where $1/\mathbb{E}[T_{\text{service}}]$ is the rate at which one request is serviced and k/m is the number of requests that can be processed simultaneously. The load can be written as

$$\rho = \frac{\lambda_0 u(u - 1)m}{2k}\mathbb{E}[T_{\text{service}}]. \tag{4}$$

It can be shown that, if and only if $\rho > 1$, the system is overloaded and the sojourn times go to infinity (see Chapter 14 from [29]). This happens when requests are submitted at a higher rate than they are serviced. For example, if the number of users is too large, the number of requests waiting to be serviced will grow indefinitely and the sojourn times will go to infinity over time. Therefore, service will only be possible if the load remains below one.

C. Quantum Circuit Switching

Quantum circuit switching (QCS) protocols are a particular type of protocol for on-demand distribution of quantum states. In QCS, each pair of users can submit (n, w)-requests to the network controller. Incoming requests are placed in a first-in-first-out queue, where older requests have priority over more recent ones. Once enough resources are available, a request can leave the queue and some network resources are reserved to meet that request. In our model, the resources reserved are forwarding stations at quantum repeaters over a path that connects both users. After this, one of the users can start sending quantum data packets, which are forwarded from repeater to repeater until arriving at the other user. This process continues until n packets are successfully delivered over a sliding time window w (recall that each packet has a probability p of being flagged with an uncorrectable error). When the process is completed, the forwarding stations become available and can be assigned to the first request in the queue. Note that we consider a queue with infinite capacity, as opposed to classical circuit switching, where the queue has a maximum size and additional requests are rejected (e.g., new calls were rejected when all lines were occupied in early analog telephone networks) – we do this because we are interested in understanding the behavior of the system without such a constraint. Moreover, each pair of users is allowed to submit new requests even when some of their previous requests are still waiting in the queue.

The main degree of freedom in our QCS proposal is the number of forwarding stations per repeater that are reserved for each request:

1) If we reserve only one forwarding station per request, packets must be sent sequentially, i.e., $m = 1$, since each station can only forward one packet at a time (Figure 1a). We call this strategy *sequential distribution* of packets.
2) An alternative is to reserve multiple forwarding stations, such that multiple packets can be routed in parallel. We call this approach *parallel distribution* of packets. For simplicity, we assume that QCS with parallel distribution

reserves all k stations for each request (Figure 1b), i.e., $m = k$. This assumption allows us to compare both extreme cases: one in which only one station per repeater is reserved for each request, and the other in which all k stations are reserved for each request. In practice, parallel distribution of quantum packets can be implemented with some form of frequency multiplexing [30].

In sequential distribution, the service time T_{service} is larger, but many requests can be processed simultaneously (as long as there are $k > 1$ forwarding stations), which might yield a shorter waiting time T_{wait}. Conversely, parallel distribution provides a faster service (i.e., smaller T_{service}) at the expense of processing one request at a time, which could entail a longer waiting time T_{wait}. It is therefore nontrivial to determine a priori whether packets should be distributed sequentially or in parallel. The first step in the design of efficient QCS protocols is to answer the following question: *should quantum data packets be distributed sequentially or in parallel?* This is the main question we address in this work.

III. SEQUENTIAL VS PARALLEL DISTRIBUTION OF QUANTUM DATA PACKETS

In this Section, we explore the situations in which sequential distribution of packets may be advantageous over parallel distribution and vice versa. We consider two use cases:

1) *Small budget.* We consider the more affordable one-way quantum repeater architecture from ref. [11], which only requires two matter qubits and a single-photon emitter per forwarding station. The encoding used by this repeater is not fault-tolerant and therefore can only be used over short inter-repeater distances ($L_0 \sim 1$ km). Over such a short distance, the scheme is near-deterministic, i.e., we can assume packets are successfully delivered with probability $p \approx 1$. Since we consider a tight financial budget, we also assume that there is a single central repeater that is directly connected to the users (i.e., $N = 0$), which means that the size of the network cannot be larger than $L \sim 1$ km.
2) *Large budget.* In this case, we employ a more expensive type of forwarding station: the all-photonic proposal from ref. [31]. This architecture can encode quantum states using large distance codes, which allows for distribution of packets over longer inter-repeater distances ($L_0 > 1$ km). Since the financial budget is larger than before, we can place N repeaters in between each user and the central repeater. The probability of successful delivery depends on the number of repeaters, N, and the size of the network, L:

$$p(L, N) = 10^{-\frac{\alpha_{\text{eff}}(L_0)}{10}2L}, \tag{5}$$

where $L_0 = L/(N + 1)$ is the inter-repeater distance, and $\alpha_{\text{eff}}(L_0)$ is an effective attenuation coefficient that depends on the type of encoding used and on the repeater efficiency. Here, we assume $\alpha_{\text{eff}}(L_0) \approx 10^{-6}(277L_0^2 + 29L_0^4)$ dB/km, which corresponds to forwarding stations

that employ the [[48, 6, 8]] generalized bicycle code [32] and have a 90% efficiency (which incorporates photon-source and detector efficiencies, on-chip loss, and coupling losses into a single parameter). This encoding and efficiency was also used as an example in ref. [31]. To make good use of the budget available, we choose N following the strategy from ref. [31], where the authors propose maximizing the number of repeaters per kilometer divided by the probability p, i.e., they optimize the cost function $(2N+1)/(Lp)$. For the $[[48, 6, 8]]$ code and a fixed distance L, the value of N that minimizes the cost is the one that yields $p \approx 0.7$ (e.g., the optimal solutions for $L = 7.5, 13, 18, 30$ km are $N = 0, 1, 2, 5$).

The main motivation behind these two use cases is that they correspond to a scenario with deterministic delivery of packets over short distances ($p = 1$, use case 1) and a scenario with probabilistic delivery over longer distances ($p < 1$, use case 2).

In the following subsections, we analyze the performance of a QCS protocol in the previous use cases. In Subsection III-A we analyze the maximum number of users that the system can support before sojourn times go to infinity. Then, in III-B, we compare the performance of sequential and parallel distribution, in terms of the mean sojourn time, when the budget is small (use case 1) and when it is large (use case 2). Lastly, in III-C, we focus on use case 2 and study the interplay between the number of users and the distances between them, considering a fixed number of repeaters N. In the extended version of this paper [33], we provide additional details and motivation for the parameter values chosen in our examples (including our choices of all-photonic quantum repeaters) and present additional examples.

A. Critical number of users

Before looking at the performance of sequential and parallel distribution of packets, the first question we ask is: *how many users can be supported by the quantum network?* That is, we want to know the maximum number of users that can be serviced before the sojourn times go to infinity – what we call the *critical number of users*, u_{crit}. We find u_{crit} by ensuring that the load of the system (4) is below 1, which yields

$$u_{\mathrm{crit}} = \left\lfloor \frac{1}{2} + \frac{1}{2}\sqrt{1 + \frac{8k}{\lambda_0 m \mathbb{E}\left[T_{\mathrm{service}}\right]}} \right\rfloor, \qquad (6)$$

where $m = 1$ for sequential distribution and $m = k$ for parallel distribution.

In the small-budget situation (use case 1), $p = 1$ and $N = 0$. Then, the mean service time from (3) takes a closed form:

$$\mathbb{E}\left[T_{\mathrm{service}}\right]_{p=1} = \frac{2L}{c} + t_{\mathrm{fwd}} \left\lceil \frac{n}{m} \right\rceil. \qquad (7)$$

This expression can be derived using (3) and the fact that $\mathbb{E}[B_{n,w,1,m}] = \lceil n/m \rceil$ (for further details on the latter, see the Appendix). From (6) and (7), one can show that, for a small budget, the critical number of users is larger when packets are distributed sequentially rather than in parallel. This can
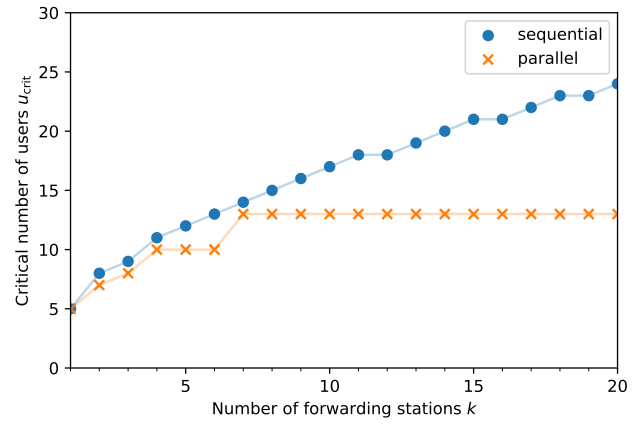

Fig. 3. **A network with more forwarding stations k can support more users, but only when distribution of packets is sequential.** Critical number of users, u_{crit}, vs number of forwarding stations, k, for QCS with sequential (blue dots) and parallel (orange crosses) distribution of packets, in the small budget use case (see (6) and (7)). The critical number of users is the maximum number of users the system can support before the sojourn time goes to infinity. In sequential distribution, $u_{\mathrm{crit}}(k)$ scales as $\sqrt{k}$. In parallel distribution, $u_{\mathrm{crit}}(k) = \mathrm{constant}, \forall k > n$. Parameters used in this figure: $N = 0$, $p = 1$, $n = 7$, $w = 10$, $\lambda_0 = 10^{-4}$ $\mu\mathrm{s}^{-1}$, $c = 0.2$ km/μs, $t_{\mathrm{fwd}} = 100$ μs.

be seen in Figure 3, which shows u_{crit} for increasing number of forwarding stations. It is possible to provide service to a larger number of users (i.e., increase u_{crit}) by increasing the number of forwarding stations k, but only when packets are distributed sequentially. In particular, the critical number of users scales as $\sqrt{k}$. When packets are distributed in parallel, increasing the number of stations beyond $k = n$ does not allow for an increased number of users. This is a consequence of the parallel distribution model: we assume that all the forwarding stations are used simultaneously for a single request. When packet distribution is deterministic ($p = 1$), only n forwarding stations are needed to distribute n packets in parallel, and increasing k beyond this number will not provide any added benefit – those extra forwarding stations will remain unused.

When the financial budget is large (use case 2), the service time follows a nontrivial distribution and the mean value cannot be computed with (7) anymore. Depending on the values of p and w, we were able to compute the mean service time analytically or required Monte Carlo sampling (see the Appendix for further details). We observed that, for most parameter regimes, sequential distribution can still provide service to a larger number of users than parallel distribution. However, this is not true in the parameter regime where resources are scarce, namely, when: (i) packet distribution is not deterministic ($p < 1$; specifically in our use case, $p \approx 0.7$), (ii) applications require many states to be distributed over a short period of time or users have short-lived quantum memories (small w compared to n), and (iii) there are only a few forwarding stations per repeater (small k). In a situation that meets these three conditions, many states must be successfully delivered over a short time window. Since each packet has a nonzero probability of failure, this process can take a very long

time when packets are delivered one by one, as successful packets older than the time window are discarded. Hence, parallel distribution can complete requests much faster and therefore provide service to a larger number of users. This reasoning only holds if k is small: if there is a large number of forwarding stations, more users can be served simultaneously with sequential distribution.

In the large-budget scenario, the asymptotic behavior in k is similar to the small-budget case. The mean service time when $p < 1$ is lower bounded by (7), and therefore the critical number of users for $p = 1$ (small budget) is an upper bound for $p < 1$ (large budget). In particular, this means that u_{crit} cannot scale faster than $\sqrt{k}$ with sequential distribution, and it converges to a constant as $k \to \infty$ with parallel distribution.

B. Mean sojourn time

Now we focus on the scenarios in which both sequential and parallel distribution can provide service within a finite mean sojourn time (MST). The MST can be computed according to (2), which requires calculating the mean waiting time and the mean service time in advance. In the Appendix we provide analytical formulas and numerical methods to compute them.

Figure 4 shows the relative difference in MST between sequential and parallel distribution, for the small-budget use case (Figure 4a) and the large-budget use case (Figure 4b). In both use cases, there is a region where both sequential and parallel distributions are possible. This region is shaded blue/red when sequential/parallel distribution is faster. When the number of users is too large, service is not possible. We indicate with an 's'/'p' those regions where service can only be provided by sequential/parallel distribution. In dark regions with an 'x', service is not possible at all. We draw the following main observations from the figure:

- In the small-budget use case (Figure 4a), increasing the number of users (for fixed k) leads to a region where only sequential distribution can provide service. If the number of users keeps increasing, no service is possible at all. In the large-budget case (Figure 4b), we observe a similar behavior, except that there is also a region where only parallel distribution (instead of sequential) is possible. This is the same phenomenon that was discussed in the previous section: when $p < 1$, w is close to n, and k is small, parallel distribution supports more users.

- For a fixed number of forwarding stations k, parallel distribution is faster than sequential distribution when the number of users is small. As we increase the number of users (for fixed k), the advantage of parallel over sequential distribution generally decreases. For some values of k, sequential distribution eventually becomes slightly better. This happens because the main feature of sequential distribution is that multiple requests can be processed simultaneously, and this is particularly beneficial when there is a large number of requests (in our model, a large number of users implies a large number of requests). In some other cases, parallel distribution is always better (e.g., in the small-budget case from Figure

4a with $k > 6$). When parallel distribution has a larger critical number of users (e.g., in the large-budget case from Fig 4b with $k = 2, 3$), the advantage in MST over sequential distribution can actually increase dramatically as the number of users increases (e.g., increasing u from 4 to 5 when $k = 2$, in Figure 4b). The reason is that the MST of sequential distribution diverges as we get close to a region where only parallel distribution is possible.

From this analysis, we conclude that parallel distribution generally fulfills user requests at a higher rate, although in some situations sequential distribution is preferable as it can provide service to a larger number of users.

C. Many users over long distances

Next, we investigate the effect of increasing the number of users and the distances between them. Here, we assume the same all-photonic forwarding stations from ref. [31] as in the large-budget use case. This means that the probability of successful packet delivery depends on the distance between users and on the number of repeaters according to (5). However, as opposed to the large-budget use case, we now consider a fixed but arbitrary number of repeaters per user, N, to be placed between the user and the central repeater – that is, we do not choose N according to any cost function optimization. Each repeater has a fixed number of forwarding stations k. The question we address here is: *how far apart can the users be to provide service to them?*

Specifically, we want to analyze the critical distance L_{crit} that overloads the system, i.e., the value of L that makes the load from (4) equal to 1. Combining (3) and (4), we find that the critical distance is given by

$$L_{\text{crit}} = \frac{ck}{\lambda_0 u(u-1)m} - \frac{ct_{\text{fwd}}}{2}(2N+1)\mathbb{E}\left[B_{\text{n,w,p}(L_{\text{crit}},N),m}\right].$$
(8)

Recall that m is the number of packets that are sent in each batch ($m = 1$ for sequential distribution and $m = k$ for parallel distribution). Note that (8) is a transcendental equation, since p is a function of L_{crit}, and it must be solved numerically.

The first observation from (8) is that L_{crit} decreases (and eventually drops to zero) for increasing u, regardless of the number of repeaters N. The second term is always negative, so the critical distance can be upper bounded as

$$L_{\text{crit}} \leq \frac{ck}{\lambda_0 u(u-1)m}.$$
(9)

The scaling with the number of users is therefore upper bounded by $\sim u^{-2}$. This unveils a tradeoff between the number of users and the distances between them. When service is provided to a large number of users, the number of incoming requests will increase and this will put a higher load on the system. If the distances between users are large, service times will increase, since packets will need to travel further away, which will also increase the load. Consequently, we must decrease u to increase L and vice versa. Figure 5 shows an example where we can observe this effect. The plot assumes
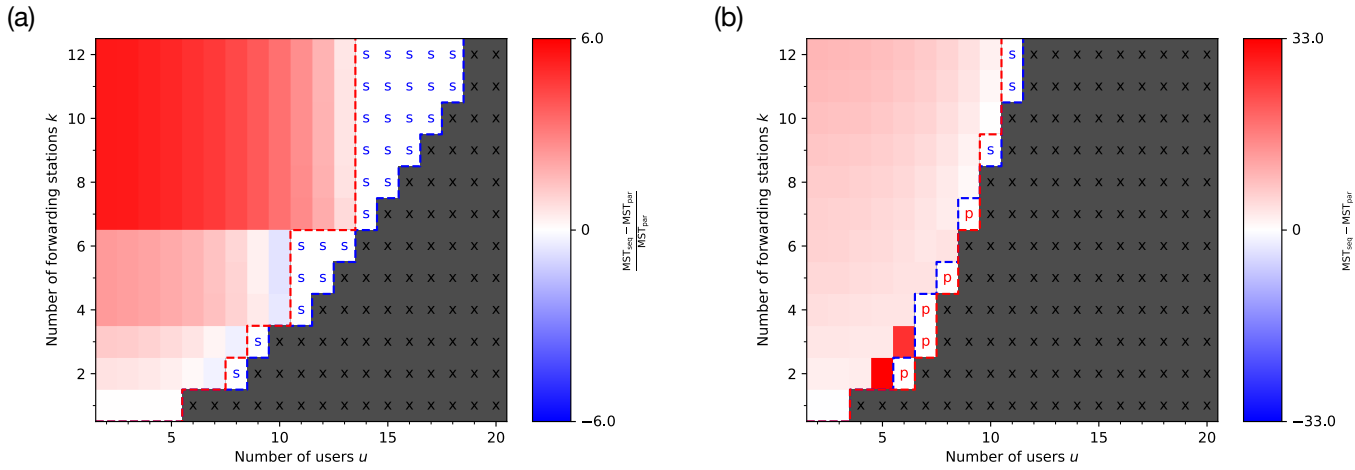
Fig. 4. **Parallel distribution of packets is generally faster.** Relative difference in mean sojourn time (MST) between sequential and parallel packet distribution, for different numbers of users u and forwarding stations k. **(a)** Small-budget use case ($p = 1$; $w \geq n$; $N = 0$, $L = 1$ km) and **(b)** large-budget use case ($p \approx 0.7$) with $N = 0$, $L = 7.5$ km, and $w = 8$. Sequential/parallel distribution provides lower MST in blue/red regions. In regions with an 's'/'p', only sequential/parallel distribution can provide service (i.e., yield finite MST). In dark regions with an 'x', no service is possible. Parameters used in this figure: $n = 7$, $\lambda_0 = 10^{-4}$ μs^{-1}, $c = 0.2$ km/μs, $t_{fwd} = 100$ μs. MSTs in (a) calculated with (7). MSTs in (b) calculated with Monte Carlo sampling with 10^7 samples (the standard error in the relative difference in MST was below 0.5 for every combination of parameters).

sequential distribution of packets and specific values of w and k, but we have observed the same behavior for parallel distribution and other parameter values.

We have shown that L and u cannot be scaled up simultaneously when N is fixed, but *what if we add more intermediate repeaters to boost the probability of successful packet delivery? Would this allow us to place many users further apart?* Intuitively, one may think that this is the case, since a larger number of repeaters N provides a larger success probability p (see (5)), which should decrease the service time. However, this intuition is not always correct, since each repeater introduces a delay t_{fwd} every time it forwards a quantum data packet. These delays accumulate and increase the service time, despite the gain from a higher p. If there are too many repeaters, these forwarding delays may overload the system, preventing requests from being met within a finite amount of time.

When the number of users is small, increasing the number of repeaters is actually beneficial: it allows us to increase the distances between users (i.e., larger N provides larger L_{crit}), as can be seen in Figure 5. When the number of users is large, service is only possible if the number of repeaters is small, that is, L_{crit} drops to zero quicker for larger N. As a consequence, for intermediate values of u, there is an finite value of N that maximizes the critical distance. For instance, when there are six users in the example from Figure 5, using $N = 2$ repeaters allows the users to be further apart than if we used $N = 5$ repeaters.

From this analysis we draw two main conclusions:

1) There exists a tradeoff in the implementation of a QCS protocol: the number of users and the distances between them cannot be scaled up simultaneously. This tradeoff happens for any number of repeaters N and any function
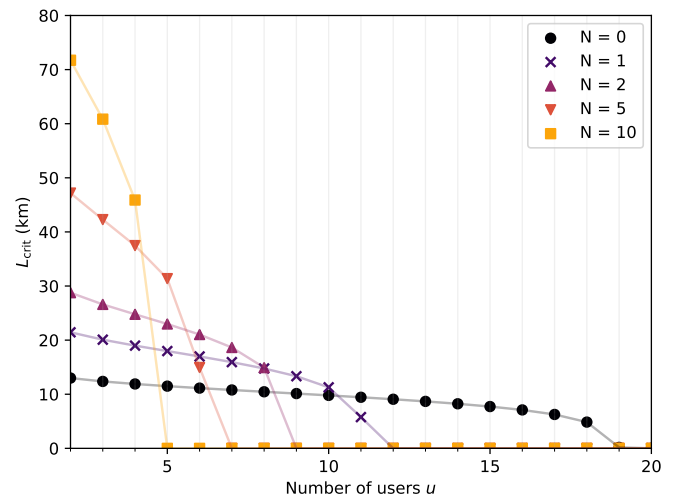

Fig. 5. **Networks with many users cannot cover long distances.** Critical distance L_{crit} vs number of users u, for different numbers of repeaters N. Parameters used in this figure: sequential distribution, $n = 7$, $w \to \infty$, $k = 12$, $\lambda_0 = 10^{-4}$ μs^{-1}, $c = 0.2$ km/μs, $t_{fwd} = 100$ μs.

$p(L, N)$ (see (9)), which means that the tradeoff will exist even with unlimited resources and ideal hardware.

2) Increasing the number of repeaters to boost the probability of successful packet delivery is not necessarily desirable, since we must also consider the forwarding delays.

IV. OUTLOOK

This paper lays the groundwork for further exploration and refinement of quantum circuit switching (QCS) protocols in the context of one-way quantum networks. We have explored two fundamental resource-allocation strategies: in the first one,

quantum data packets are distributed sequentially, and, in the second one, they are distributed in batches. We concluded that sequential distribution can generally provide service to a larger number of users, although parallel distribution generally meets requests at a higher rate. We also found a tradeoff between the number of users supported by the QCS protocol and the size of the network.

Here, we have considered the mean sojourn time as the main quantity to measure protocol performance. A more detailed characterization of the whole probability distribution of the sojourn time is left as future work.

Future research could also focus on extending the analysis to more complex network topologies beyond the star configuration considered here. Investigating the impact of connectivity patterns on the performance of QCS protocols would provide valuable insights for real-world quantum network deployment. Moreover, exploring adaptive QCS strategies that dynamically adjust resource allocation based on traffic conditions (i.e., finding an intermediate strategy between sequential and parallel distribution of packets) could enhance efficiency and scalability.

Different request models must be also investigated in future work. Here, we have assumed that each pair of users submits requests according to a Poisson process. However, some networks may experience different types of requests, e.g., peak demands may happen at specific times of the day. Moreover, each pair of users could request a different number of packets n to be distributed over a different time window w. Considering a different request model would potentially lead to different conclusions about the scalability of the network (e.g., we would expect a different scaling of the critical number of users with the number of forwarding stations, and a different relationship between critical distance and number of users).

We consider QCS and its integration into one-way quantum networks a promising avenue towards practical quantum information processing and quantum internet applications.

APPENDIX
ANALYTICAL CALCULATION OF THE MEAN WAITING TIME AND THE MEAN SERVICE TIME

In this Appendix, we show how to calculate the mean waiting time and the mean service time of a QCS protocol in a star network. For that, we model the system as an M/G/s queue. In this queue model, new requests arrive following a Poisson distribution, i.e., request arrivals are Markovian (M). Incoming requests are placed in a common queue. Requests in the queue are processed according to a first-in-first-out policy. Processing a request takes some service time, which follows a general distribution (G), and a maximum of s requests can be processed simultaneously. When quantum data packets are distributed sequentially ($m = 1$), each of the k forwarding stations is dedicated to meet one request and therefore $s = k$. When packets are distributed in parallel ($m = k$), all forwarding stations are reserved to meet one request at a time and therefore $s = 1$.

In Sections A and B, we show how to compute waiting and service times, respectively, in the M/G/s model for QCS.

A. Waiting time

Calculating the waiting time poses a major problem: no general analytical solution is known for the waiting time of an M/G/s queue. Only for $s = 1$, a closed-form solution is known (see, e.g., Chapter 14 from [29]):

$$\mathbb{E}\left[T_{\text{wait}}^{\text{M/G/1}}\right] = \frac{\lambda \mathbb{E}\left[\left(T_{\text{service}}^{\text{M/G/1}}\right)^2\right]}{2\left(1 - \lambda \mathbb{E}\left[T_{\text{service}}^{\text{M/G/1}}\right]\right)}, \qquad (10)$$

where λ is the request arrival rate (in our problem, $\lambda = \lambda_0 u(u-1)/2$, where λ_0 is the request rate per pair of users and u is the number of users). The mean waiting time can therefore be computed exactly when $s = 1$, provided that the first two moments of the service time are known.

Multiple approximations and bounds for the waiting time distribution have been found for $s > 1$ (see, e.g., [34]). One of the most well-known approximations is the early formula from ref. [35]:

$$\mathbb{E}\left[T_{\text{wait}}^{\text{M/G/}s}\right] \approx \frac{C_{\text{service}}^2 + 1}{2} \mathbb{E}\left[T_{\text{wait}}^{\text{M/M/}s}\right], \qquad (11)$$

where $C_{\text{service}}^2 \equiv \text{Var}\left[T_{\text{service}}^{\text{M/G/}s}\right] / \mathbb{E}\left[T_{\text{service}}^{\text{M/G/}s}\right]^2$ is the squared coefficient of variation of the service time distribution, and M/M/s is an auxiliary queuing system with a single queue, Markovian arrivals (M), exponentially distributed service times (M) with mean $\mu \equiv \mathbb{E}\left[T_{\text{service}}^{\text{M/G/}s}\right]$, and capacity to process s requests simultaneously. The mean waiting time of the auxiliary queue can be computed as follows (see Chapter 14 from [29]):

$$\mathbb{E}\left[T_{\text{wait}}^{\text{M/M/}s}\right] = \frac{\lambda^s}{s!s\mu^{s+1}z^2}\left(\frac{\lambda^s}{s!\mu^s z} + \sum_{i=0}^{s-1}\frac{\lambda^i}{i!\mu^i}\right)^{-1}, \quad (12)$$

with $z \equiv 1 - \lambda/(s\mu)$. Refs. [34] and [36] provide empirical evidence that the approximation (11) works well when the service time distribution has a small coefficient of variation. We observed empirically that, in the use cases studied in this paper, the squared coefficient of variation of the service time C_{service}^2 is small and therefore we expect this approximation to be accurate (see the extended version of this paper [33]). Note also that the approximation assumes the first two moments of the service time distribution are known.

B. Service time

As explained in the main text, the service time can be computed as

$$T_{\text{service}} = \frac{2L}{c} + t_{\text{fwd}}(2N+1)B_{n,w,p,m}, \qquad (13)$$

where L is the distance between each user and the central repeater, c is the speed of light in the physical channels, t_{fwd} is the forwarding time, and N is the number of repeaters between each user and the central repeater. Recall that $B_{n,w,p,m}$ is the

number of batches of quantum data packets that must be sent from user to user until n packets are successfully distributed within a time window w, assuming each packet has a success probability p and each batch contains m packets. The mean service time is then given by

$$\mathbb{E}\left[T_{\text{service}}\right] = \frac{2L}{c} + t_{\text{fwd}}(2N+1)\mathbb{E}\left[B_{n,w,p,m}\right], \qquad (14)$$

since $B_{n,w,p,m}$ is the only random variable involved. As shown in Section A, the second moment of the service time is also required to compute the waiting time. The second moment can be computed as

$$\mathbb{E}\left[T_{\text{service}}^2\right] = \left(\frac{2L}{c}\right)^2 + \frac{4L}{c}t_{\text{fwd}}(2N+1)\mathbb{E}\left[B_{n,w,p,m}\right]$$
$$+ \left(t_{\text{fwd}}(2N+1)\right)^2 \mathbb{E}\left[B_{n,w,p,m}^2\right]. \qquad (15)$$

From (14) and (15), we conclude that the service and the waiting times can be efficiently computed if we can efficiently compute the first two moments of $B_{n,w,p,m}$. Next, we discuss how to compute them.

Consider a sequence of i.i.d. random variables S_i, which follow a binomial distribution with m attempts and probability of success p. The value of S_i corresponds to the number of successful packets delivered in batch i. Let $B_{n,w,p,m}$ be the number of batches required until completion, i.e.,

$$B_{n,w,p,m} = \inf\left\{x \ : \ \sum_{j=x-w+1}^{x} S_j \geq n\right\}. \qquad (16)$$

It is possible to calculate the probability distribution of $B_{n,w,p,m}$ analytically when there is no multiplexing ($m = 1$) [19]. When $p = 1$, the solution is trivial since we deterministically need n batches to successfully deliver n packets (there is only one per batch). In that case, we have

$$\mathbb{E}[B_{n,w,1,1}] = n, \qquad (17)$$

$$\mathbb{E}\left[(B_{n,w,1,1})^2\right] = n^2. \qquad (18)$$

As shown in ref. [19], in the case of an infinite window ($w \to \infty$), $B_{n,w,p,1}$ follows a negative binomial distribution, which has the following first two moments:

$$\mathbb{E}[B_{n,\infty,p,1}] = \frac{n}{p}, \qquad (19)$$

$$\mathbb{E}\left[(B_{n,\infty,p,1})^2\right] = \frac{n(1-p) + n^2}{p^2}. \qquad (20)$$

For every other combination of values of n, p, and w, and with $m = 1$, there exists a nontrivial analytical solution, as shown in ref. [19]. However, computing the solution involves inverting a matrix whose size scales as $\mathcal{O}(w^{n-1})$. Hence, in practice we need to employ approximate methods (such as Monte Carlo sampling) for arbitrary values of n, w, and p.

In the multiplexed case ($m > 1$), there is no known analytical solution for every combination of n, w, p, and m, to the best of our knowledge. Next, we solve the problem for

two cases: (i) $p = 1$ and (ii) $w \to \infty$. In every other case, we employed Monte Carlo sampling to estimate the probability distribution of $B_{n,w,p,m}$ when needed. First, the case $p = 1$ is again trivial. When packets are always successfully delivered, we need $\lceil n/m \rceil$ batches to deliver n packets. Hence,

$$\mathbb{E}[B_{n,w,1,m}] = \left\lceil \frac{n}{m} \right\rceil, \qquad (21)$$

$$\mathbb{E}\left[(B_{n,w,1,m})^2\right] = \left\lceil \frac{n}{m} \right\rceil^2. \qquad (22)$$

For $w \to \infty$, we first calculate the survival function of $B_{n,\infty,p,m}$ as follows:

$$\Pr\left[B_{n,\infty,p,m} > b\right] \stackrel{a}{=} \Pr\left[\sum_{i=1}^{b} S_i < n\right] \qquad (23)$$

$$\stackrel{b}{=} \sum_{s_1,\ldots,s_b=0}^{m} \Pr\left[\sum_{i=1}^{b} s_i < n\right] \prod_{i=1}^{b} \Pr\left[S_i = s_i\right] \qquad (24)$$

$$\stackrel{c}{=} (1-p)^{mb} \sum_{s_1=0}^{z_1} \cdots \sum_{s_b=0}^{z_b} \left(\frac{p}{1-p}\right)^{\sum_{i=1}^{b} s_i} \prod_{i=1}^{b} \binom{m}{s_i}. \qquad (25)$$

with the following steps (and some additional algebra):

a) We define S_i as the number of successes in batch i. Completing the process in more than b batches ($B_{n,\infty,p,m} > b$) corresponds to obtaining less than n successful attempts in the first b batches ($\sum_{i=1}^{b} S_i < n$).
b) We use the law of total probability recursively, noting that S_i are i.i.d.
c) S_i follows a binomial distribution with m attempts and probability of success p. We define $z_i \equiv \min\left(m, n-1-\sum_{j=1}^{i-1} s_j\right)$.

The survival function can then be used to calculate the first two moments of $B_{n,\infty,p,m}$ as follows:

$$\mathbb{E}[B_{n,\infty,p,m}] = \sum_{b=0}^{\infty} \Pr\left[B_{n,\infty,p,m} > b\right], \qquad (26)$$

$$\mathbb{E}[B_{n,\infty,p,m}^2] = \sum_{b=1}^{\infty} b^2 \big(\Pr[B_{n,\infty,p,m} > b-1]$$
$$- \Pr[B_{n,\infty,p,m} > b]\big), \qquad (27)$$

where we used the fact that $B_{n,\infty,p,m}$ is a discrete and positive random variable.

REFERENCES

[1] M. Ruf, N. H. Wan, H. Choi, D. Englund, and R. Hanson, "Quantum networks based on color centers in diamond," *J. Appl. Phys.*, vol. 130, no. 7, 2021.

[2] 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, 2015.

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

[4] S. D. Barrett and P. Kok, "Efficient high-fidelity quantum computation using matter qubits and linear optics," *Phys. Rev. A*, vol. 71, no. 6, p. 060310, 2005.

[5] H. Bernien, B. Hensen, W. Pfaff, G. Koolstra, M. S. Blok, L. Robledo, T. H. Taminiau, M. Markham, D. J. Twitchen, L. Childress *et al.*, "Heralded entanglement between solid-state qubits separated by three metres," *Nature*, vol. 497, no. 7447, pp. 86–90, 2013.

[6] M. Żukowski, A. Zeilinger, M. A. Horne, and A. K. Ekert, ""Event-ready-detectors" Bell experiment via entanglement swapping," *Phys. Rev. Lett.*, vol. 71, no. 26, p. 4287, 1993.

[7] L.-M. Duan, M. D. Lukin, J. I. Cirac, and P. Zoller, "Long-distance quantum communication with atomic ensembles and linear optics," *Nature*, vol. 414, no. 6862, pp. 413–418, 2001.

[8] N. Sangouard, C. Simon, H. De Riedmatten, and N. Gisin, "Quantum repeaters based on atomic ensembles and linear optics," *Rev. Mod. Phys.*, vol. 83, no. 1, p. 33, 2011.

[9] W. J. Munro, A. M. Stephens, S. J. Devitt, K. A. Harrison, and K. Nemoto, "Quantum communication without the necessity of quantum memories," *Nature Photon.*, vol. 6, no. 11, pp. 777–781, 2012.

[10] S. Muralidharan, J. Kim, N. Lütkenhaus, M. D. Lukin, and L. Jiang, "Ultrafast and fault-tolerant quantum communication across long distances," *Phys. Rev. Lett.*, vol. 112, no. 25, p. 250501, 2014.

[11] J. Borregaard, H. Pichler, T. Schröder, M. D. Lukin, P. Lodahl, and A. S. Sørensen, "One-way quantum repeater based on near-deterministic photon-emitter interfaces," *Phys. Rev. X*, vol. 10, no. 2, p. 021071, 2020.

[12] S. DiAdamo, B. Qi, G. Miller, R. Kompella, and A. Shabani, "Packet switching in quantum networks: A path to the quantum internet," *Phys. Rev. Res.*, vol. 4, no. 4, p. 043064, 2022.

[13] S. C. Liew and T. T. Lee, *Principles of Broadband Switching and Networking*. John Wiley & Sons, 2010.

[14] P. Baran, "On distributed communications networks," *IEEE Trans. on Commun. Syst.*, vol. 12, no. 1, pp. 1–9, 1964.

[15] D. W. Davies, W. Price, D. Barber, and C. Solomonides, *Computer networks and their protocols*. John Wiley & Sons, Inc., 1979.

[16] G. Broomell and J. R. Heath, "Classification categories and historical development of circuit switching topologies," *ACM Comput. Surv. (CSUR)*, vol. 15, no. 2, pp. 95–133, 1983.

[17] L. Aparicio and R. Van Meter, "Multiplexing schemes for quantum repeater networks," in *Quantum Commun. and Quantum Imag. IX*, vol. 8163. International Society for Optics and Photonics, 2011, p. 816308.

[18] D. Leichtle, L. Music, E. Kashefi, and H. Ollivier, "Verifying bqp computations on noisy devices with minimal overhead," *PRX Quantum*, vol. 2, no. 4, p. 040302, 2021.

[19] B. Davies, T. Beauchamp, G. Vardoyan, and S. Wehner, "Tools for the analysis of quantum protocols requiring state generation within a time window," *arXiv preprint arXiv:2304.12673*, 2023.

[20] G. Vardoyan, S. Guha, P. Nain, and D. Towsley, "On the capacity region of bipartite and tripartite entanglement switching," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 48, no. 3, pp. 45–50, 2021.

[21] ——, "On the stochastic analysis of a quantum entanglement distribution switch," *IEEE Trans. Quantum Eng.*, vol. 2, pp. 1–16, 2021.

[22] S. Muralidharan, L. Li, J. Kim, N. Lütkenhaus, M. D. Lukin, and L. Jiang, "Optimal architectures for long distance quantum communication," *Scientific Rep.*, vol. 6, no. 1, pp. 1–10, 2016.

[23] C. Crépeau, D. Gottesman, and A. Smith, "Secure multi-party quantum computation," in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, 2002, pp. 643–652.

[24] V. Lipinska, G. Murta, J. Ribeiro, and S. Wehner, "Verifiable hybrid secret sharing with few qubits," *Phys. Rev. A*, vol. 101, no. 3, p. 032332, 2020.

[25] M. Pant, H. Krovi, D. Towsley, L. Tassiulas, L. Jiang, P. Basu, D. Englund, and S. Guha, "Routing entanglement in the quantum internet," *npj Quantum Inf.*, vol. 5, no. 1, p. 25, 2019.

[26] S. Khatri, "On the design and analysis of near-term quantum network protocols using markov decision processes," *AVS Quantum Sci.*, vol. 4, no. 3, 2022.

[27] G. Avis, F. Rozpedek, and S. Wehner, "Analysis of multipartite entanglement distribution using a central quantum-network node," *Phys. Rev. A*, vol. 107, no. 1, p. 012609, 2023.

[28] Á. G. Iñesta, G. Vardoyan, L. Scavuzzo, and S. Wehner, "Optimal entanglement distribution policies in homogeneous repeater chains with cutoffs," *npj Quantum Inf.*, vol. 9, no. 1, p. 46, 2023.

[29] P. Van Mieghem, *Performance analysis of complex networks and systems*. Cambridge University Press, 2014.

[30] O. Collins, S. Jenkins, A. Kuzmich, and T. Kennedy, "Multiplexed memory-insensitive quantum repeaters," *Phys. Rev. Lett.*, vol. 98, no. 6, p. 060502, 2007.

[31] D. Niu, Y. Zhang, A. Shabani, and H. Shapourian, "All-photonic one-way quantum repeaters with measurement-based error correction," *npj Quantum Inf.*, vol. 9, no. 1, p. 106, 2023.

[32] P. Panteleev and G. Kalachev, "Degenerate quantum ldpc codes with good finite length performance," *Quantum*, vol. 5, p. 585, 2021.

[33] Á. G. Iñesta, H. Choi, D. Englund, and S. Wehner, "Quantum circuit switching with one-way repeaters in star networks," *arXiv preprint arXiv:2405.19049*, 2024.

[34] V. Gupta, M. Harchol-Balter, J. G. Dai, and B. Zwart, "On the inapproximability of m/g/k: why two moments of job size distribution are not enough," *Queueing Syst.*, vol. 64, pp. 5–48, 2010.

[35] A. Lee and P. Longton, "Queueing processes associated with airline passenger check-in," *J. Oper. Res. Soc.*, vol. 10, no. 1, pp. 56–71, 1959.

[36] W. Whitt, "Approximations for the GI/G/m queue," *Prod. and Oper. Manage.*, vol. 2, no. 2, pp. 114–161, 1993.