

Performance metrics for the continuous distribution of entanglement in multiuser quantum networks

Iñesta, Álvaro G.; Wehner, Stephanie

DOI

[10.1103/PhysRevA.108.052615](https://doi.org/10.1103/PhysRevA.108.052615)

Publication date

2023

Document Version

Final published version

Published in

Physical Review A

Citation (APA)

Iñesta, Á. G., & Wehner, S. (2023). Performance metrics for the continuous distribution of entanglement in multiuser quantum networks. *Physical Review A*, *108*(5), Article 052615.
<https://doi.org/10.1103/PhysRevA.108.052615>

Important note

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


Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Performance metrics for the continuous distribution of entanglement in multiuser quantum networks

Álvaro G. Iñesta * and Stephanie Wehner

QuTech, Delft University of Technology, Lorentzweg 1, 2628 CJ Delft, The Netherlands;

EEMCS, Quantum Computer Science, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands;
and Kavli Institute of Nanoscience, Delft University of Technology, Lorentzweg 1, 2628 CJ Delft, The Netherlands



(Received 7 August 2023; accepted 7 November 2023; published 27 November 2023)

Entangled states shared among distant nodes are frequently used in quantum network applications. When quantum resources are abundant, entangled states can be continuously distributed across the network, allowing nodes to consume them whenever necessary. This continuous distribution of entanglement enables quantum network applications to operate continuously while being regularly supplied with entangled states. Here, we focus on the steady-state performance analysis of protocols for continuous distribution of entanglement. We propose the virtual neighborhood size and the virtual node degree as performance metrics. We utilize the concept of Pareto optimality to formulate a multiobjective optimization problem to maximize the performance. As an example, we solve the problem for a quantum network with a tree topology. One of the main conclusions from our analysis is that the entanglement consumption rate has a greater impact on the protocol performance than the fidelity requirements. The metrics that we establish in this paper can be utilized to assess the feasibility of entanglement distribution protocols for large-scale quantum networks.

DOI: [10.1103/PhysRevA.108.052615](https://doi.org/10.1103/PhysRevA.108.052615)

I. INTRODUCTION

Quantum networks are expected to enable multiparty applications that are provably impossible by using only classical information. These applications range from basic routines, such as quantum teleportation [1,2], to more complex tasks, such as quantum key distribution [3,4] and entanglement-assisted distributed sensing [5,6]. Some of these applications may operate in the *background* (e.g., a quantum key distribution subroutine that is continuously generating a secret key), as opposed to *sporadic* applications that are executed after the users actively trigger them. Most quantum network applications consume shared entanglement as a basic resource. *Entanglement distribution protocols* are used to generate and share multipartite entanglement among remote parties. There are two main approaches to distribute entanglement among the nodes [7,8].

(1) Protocols for *on-demand distribution of entanglement* distribute entangled states only after some nodes request them. The request may involve some quality-of-service requirements (e.g., a minimum quality of the entanglement). This type of protocol typically involves solving a routing problem and scheduling a set of operations on a subset of nodes [7,9–12].

(2) Protocols for *continuous distribution of entanglement* (CD protocols) continuously distribute entangled states among the nodes. These entangled states can be consumed by the nodes whenever they need them. This allows *background applications* to continuously operate and consume

entanglement in the background. In this paper, we focus on CD protocols that provide entanglement to background applications.

On-demand distribution is generally more efficient, since entanglement is only produced when it is needed. This makes on-demand distribution more suitable for quantum networks where the quantum resources are limited (e.g., networks with a small number of qubits per node). As a consequence, previous work, both theoretical [7,9–12] and experimental [13–19], has mostly focused on this type of protocol in quantum networks with a simple topology or with very limited number of qubits per node.

On-demand distribution requires a scheduling policy that tells the nodes when to perform each operation based on specific demands. If the number of nodes involved in the generation of entanglement is large, the scheduling policies become more complex. In contrast, the continuous distribution of entanglement does not necessarily require an elaborate application-dependent schedule. Therefore, CD protocols are expected to allocate resources faster and prevent traffic congestion in large quantum networks. Here, we focus on the performance evaluation of CD protocols. Specifically, we consider protocols that distribute bipartite entanglement among remote nodes. We refer to shared bipartite entanglement as an *entangled link*. We focus on entangled links because this is a basic resource needed in many quantum network applications [3,4,20,21], where nodes generally need many copies of a bipartite entangled state with high enough quality. Even when multipartite entanglement is required, it can be generated using entangled links [22–25].

We consider a quantum network with n nodes. Some pairs of nodes are *physical neighbors*: they are connected by a

*a.gomezinesta@tudelft.nl

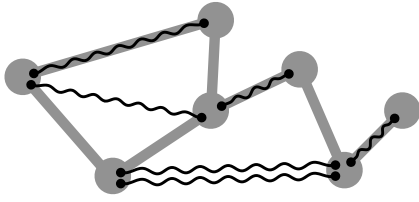


FIG. 1. Illustration of a seven-node quantum network. The nodes are represented as gray circles, and physical channels connecting neighboring nodes are represented as gray lines. Entangled links are represented as black lines connecting two occupied qubits (small black circles). The physical topology is static, while the entangled links are continuously created, discarded, and consumed.

physical channel, such as optical fibers [26,27] or free space [13,28]. This is depicted in Fig. 1. To generate long-distance entanglement, we assume the nodes can perform the following basic operations: (i) heralded generation of entanglement between physical neighbors [16,29], which successfully produces an entangled link with probability p_{gen} and otherwise raises a failure flag; (ii) entanglement swaps [30–32], which consume one entangled link between nodes A and B and another entangled link between nodes B and C to generate a single link between A and C with probability p_s ; (iii) removal of any entangled link that has existed for longer than some cutoff time t_{cut} to prevent the existence of low-quality entanglement in the network [33–37]; and (iv) consumption of entangled links in background applications at some constant rate p_{cons} . Note that the choice of cutoff time is determined by the minimum fidelity required by the applications, F_{app} . We allow for multiple entangled links to be shared simultaneously between the same pair of nodes (see Fig. 1).

Evaluating the performance of a CD protocol is a fundamentally different problem to evaluating the performance of on-demand protocols, since each type of protocol serves a different purpose. In on-demand protocols, one generally wants to maximize the rate of entanglement distribution among a specific set of end nodes and the quality of the entanglement (or some combined metric, such as the secret key rate [38]). By contrast, the goal of a CD protocol is (i) to distribute entanglement among the nodes such that it can be continuously consumed in background applications and (ii) to ensure that some entanglement is available for sporadic applications. To quantify the performance of a CD protocol, we need metrics that take these goals into account. A simple approach is to analyze the configuration of entangled links that a CD protocol can achieve. This configuration is time dependent due to the dynamic nature of the entangled links. Most previous work aimed at describing the connectivity of large-scale quantum networks disregards the time dependence of the system. As a consequence, previous results do not depend explicitly on parameters that determine the evolution of the entanglement, such as the coherence time. For example, in Refs. [39,40], the authors study a graph in which the edges are entangled links that exist at a specific instant. Some authors have described the connectivity of a quantum network using percolation theory [41–47], which also disregards the time evolution of the entangled states, and often assumes specific topologies and some form of preshared entanglement. Another line of related

work is the use of preshared entanglement for on-demand applications [48,49].

In this paper, we consider quantum networks with arbitrary topologies where entanglement is continuously being generated and consumed. We propose metrics to evaluate the performance of CD protocols. These metrics take into account the time dependence of the system and can be used to optimize the protocol performance.

Our main contributions are the following.

(1) We define metrics to evaluate the performance of CD protocols in heterogeneous quantum networks with an arbitrary topology, namely, we define the *virtual neighborhood size* and the *virtual node degree*. These metrics provide information about the number of nodes that are able to continuously run background applications and about the number of existing backup entangled links to run sporadic applications.

(2) We provide analytical and numerical tools to compute the performance metrics.

(3) We provide a mathematical framework to maximize the virtual neighborhood size of every node in a heterogeneous network, while providing some minimum quality-of-service requirements (e.g., a minimum number of backup links). We do this via the concept of Pareto optimality.

(4) We study the relation between the steady-state performance of the entanglement distribution protocol and the application requirements (minimum fidelity and link consumption rate) in a quantum network with a tree topology.

Our main findings are the following.

(1) The expected virtual neighborhood size rapidly drops to zero when the entanglement consumption rate increases beyond the entanglement generation rate.

(2) In a quantum network with a tree topology and with high entanglement generation rate, the consumption rate has a stronger effect on the virtual neighborhood size than the minimum fidelity required by the applications. In other words, background applications that require a high consumption rate affect the CD protocol performance more than applications that require a high fidelity.

(3) The set of protocol parameters that maximize the virtual neighborhood size is node dependent. Consequently, in heterogeneous networks with an arbitrary topology we need to solve a multiobjective optimization problem.

The structure of the paper is as follows. In Sec. II, we define the network model (physical topology, quantum operations, and quantum resources). In Sec. III, we provide an example of a CD protocol. In Sec. IV, we formally define the virtual neighborhood and the virtual node degree. We apply these definitions to evaluate the performance of a CD protocol using analytical and numerical methods. As an example, we analyze a CD protocol in a quantum network with a tree topology. In Sec. V, we discuss the implications and limitations of our paper.

II. NETWORK MODEL

In this section we describe the physical topology of the network and the quantum operations that the nodes can perform. We also discuss the background applications requirements and the management of quantum resources at each node.

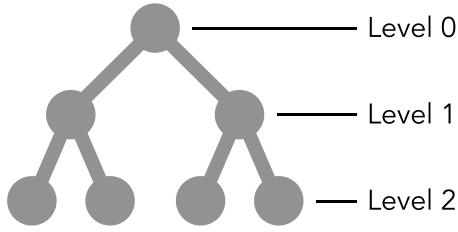


FIG. 2. (2,3)-tree network. Each node is represented as a gray circle and is connected to two other nodes in a lower level.

We consider a quantum network with n nodes (see Fig. 1). Nodes can store quantum states in the form of qubits, and they can manipulate them as we describe below. Additionally, some nodes are connected by a *physical channel* over which they can send quantum states. Qubits can be realized with different technologies, such as nitrogen vacancy (NV) centers [16–19,36], trapped ions [14,15], or neutral atoms [50], while physical channels can be realized with optical fibers [26,27] or free space [13,28].

A. Physical topology

Two nodes are *physical neighbors* if they share a physical channel. The *physical node degree* d_i of node i is the number of its physical neighbors. The set of nodes and physical channels constitutes the *physical topology* of the quantum network. Early quantum networks are expected to have simple physical topologies, such as a chain where each node is connected to two other nodes [9,12,51] and a star topology where all nodes are only connected to a central node [11,52]. More advanced networks are expected to display a more complex physical topology, such as a dumbbell structure with a backbone connecting two metropolitan areas.

The definitions and methods we develop in this paper are general and apply to an arbitrary physical topology, which can be described using an *adjacency matrix* A (element A_{ij} is 1 if nodes i and j are physical neighbors and zero otherwise). To illustrate how our methods can be valuable and effective, we apply them to a quantum network with a tree topology as an example. In a tree, any node can be reached from any other node by following exactly one path. This topology is particularly relevant as it has been shown that it requires a reduced number of qubits per node to avoid traffic congestion [53].

Definition 1. A (d, k) -tree network is an undirected unweighted graph where nodes are distributed in k levels, with d^l nodes in level $l \in 0, 1, \dots, k-1$. Each node in level l is connected to d nodes in the $(l+1)$ th level, and is only connected to one node in the $(l-1)$ th level.

The total number of nodes in a (d, k) -tree is $n = (d^k - 1)/(d - 1)$, and the network diameter is $2k$. A (2,3)-tree network is depicted in Fig. 2.

B. Entanglement distribution

The aim of a CD protocol is to distribute shared bipartite entangled states, which we call entangled links. Ideally, entangled links are maximally entangled states. However, entanglement generation and storage are generally noisy processes. Consequently, we assume that entangled links are

Werner states [54]: maximally entangled states that have been subjected to a depolarizing process, which is a worst-case noise model [55]. Werner states can be written as

$$\rho = \frac{4F - 1}{3} |\phi^+\rangle\langle\phi^+| + \frac{1 - F}{3} \mathbb{I}_4, \quad (1)$$

where $|\phi^+\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$ is a maximally entangled state, F is the fidelity of the Werner state to the state $|\phi^+\rangle$, and \mathbb{I}_m is the m -dimensional identity. Here, the fidelity of a mixed state ρ to a pure state $|\phi\rangle$ is defined as

$$F(\rho, |\phi\rangle) := \langle\phi| \rho |\phi\rangle. \quad (2)$$

We consider nodes that operate as first or second generation quantum repeaters [56]: physical neighbors generate entangled links via *heralded entanglement generation* using two-way signaling. This operation produces an entangled link with probability p_{gen} and otherwise raises a failure flag [16,29]. The fidelity of newly generated links, F_{new} , is generally a function of p_{gen} . For example, in the single-photon protocol [57], $F_{\text{new}} = 1 - \lambda p_{\text{gen}}$, for some $0 \leq \lambda \leq 1$ (as discussed in Ref. [58], the value of λ can be tuned by performing a batch of entanglement attempts as a single entanglement generation step [59]).

Long-distance entanglement between physically non-neighboring nodes can be generated using *entanglement swapping* [30–32], which consumes an entangled link between nodes A and B , with fidelity F_{AB} , and another one between B and C , with fidelity F_{BC} , to produce a link between A and C , with fidelity $F_{AC} \leq F_{AB}, F_{BC}$. This operation succeeds with probability p_{swap} (when it fails, both input links are lost and nothing is produced). Note that entanglement swapping also requires two-way classical signaling. See Appendix A for further details on entanglement swapping.

C. Quantum applications

The main goal of a CD protocol is to provide a continuous supply of entanglement for nodes to run applications without the need for explicitly demanding entanglement. We assume that each pair of nodes that share entanglement is continuously running quantum applications in the background, consuming entangled links at a rate p_{cons} . For simplicity, we assume $0 \leq p_{\text{cons}} \leq 1$. Since we will assume time to be slotted (see Sec. III), a consumption rate between zero and one can be interpreted as the probability that, in each time slot, two nodes that share some entangled links consume one link. We consider entanglement purification [10,60,61] as an application and therefore we omit it in our model (purification at the physical link level can be included in our model by modifying p_{gen} and F_{new} accordingly; see Appendix A for further details).

Background applications require entanglement of a high enough quality. Specifically, we assume that they need entangled links with fidelity larger than F_{app} .

D. Mitigating decoherence

The operations involving entangled links and the storage in memory have a negative impact on the quality of the links. Each entanglement swap produces a link with a lower fidelity than the input links [62]. To prevent the fidelity from dropping too low, we must limit the *maximum swap distance*, defined

as the maximum number of short-distance links that can be combined into longer distance entanglement via swaps. We denote this maximum number of links as M . Two nodes can only share entanglement if they are at most M physical links away.

Additionally, the fidelity of entangled links stored in memory decreases over time due to couplings to the environment [55,63], making old links unusable for applications that require high fidelity states. A simple technique to alleviate the effects of noisy storage consists in imposing a cutoff time t_{cut} : any link that has been stored for longer than the cutoff time must be discarded [34].

To ensure that the fidelity of every entangled link is above F_{app} in a network where new links are generated with fidelity F_{new} , it is enough to choose the values of t_{cut} and M such that [12]

$$t_{\text{cut}} \leq -T \ln \left[\frac{3}{4F_{\text{new}} - 1} \left(\frac{4F_{\text{app}} - 1}{3} \right)^{\frac{1}{M}} \right], \quad (3)$$

where T is a parameter that characterizes the exponential decay in fidelity of the whole entangled state due to the qubits being stored in noisy memories. In our analysis, we choose the largest cutoff that satisfies (3). For further details on the noise model, see Appendix A.

E. Limited quantum resources

Nodes have a limited number of qubits. These qubits can be used for communication (short coherence times) or for storage (long coherence times) [64,65]. Here, we assume a simplified setup where every qubit can be used for entanglement generation and for storage of an entangled link. Intuitively, nodes with a larger number of physical neighbors should have more resources available, to establish entanglement with many neighbors simultaneously. We assume that the maximum number of qubits that node i can store is $d_i r$, where d_i is the physical node degree of node i and $r \in \mathbb{N}$ is a hardware-dependent parameter that limits the maximum number of qubits per node.

We make an additional simplifying assumption: each qubit can only generate entanglement with a fixed neighboring node. The physical motivation behind this assumption is the lack of optical switches in the node. This assumption allows us to uniquely identify each qubit using a three-tuple address (i, j, m) . The first index, $i \in \{0, \dots, n-1\}$, corresponds to the node holding the qubit. The second index, $j \in \{0, \dots, n-1\}$, is the node with which the qubit can generate entanglement ($i \neq j$). The third index, $m \in \{0, \dots, r-1\}$, is used to distinguish qubits that share the same indices i and j . A graphical example is shown in Fig. 3.

III. PROTOCOL FOR CONTINUOUS DISTRIBUTION OF ENTANGLEMENT

The operations discussed above—entanglement generation, swaps, entanglement consumption, and application of cutoffs—are performed following a specific protocol for continuous distribution of entanglement (CD protocol). Here we consider a basic CD protocol that we will use to test our performance optimization tools. We assume a synchronous

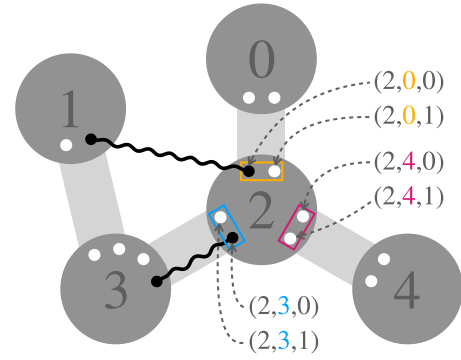


FIG. 3. Qubit addresses. Each qubit is identified by a qubit address consisting of three values (i, j, m) : i is the node holding the qubit, j is the neighboring node that can generate entanglement with that qubit, and m is used to distinguish qubits with the first two indices i and j . In this example, each node has two qubits per physical neighbor, i.e., $r = 2$.

protocol: time is divided into nonoverlapping time slots and each operation is allocated within a time slot. This is a common assumption in the field of quantum networking (see, e.g., Refs. [12,66]), since nodes generally have to agree to perform synchronized actions for heralded entanglement generation. In what follows, we focus on the single random swap (SRS) protocol, which is described in Algorithm 1. In this protocol, (i) entanglement generation is attempted sequentially on every physical link; (ii) swaps are performed using links chosen at random; and (iii) every pair of nodes that shares an entangled link consumes one link per time step with probability p_{cons} . The protocol has a single parameter, $q \in [0, 1]$, which determines how many nodes must perform a swap at each time step (if $q = 0$, no swaps are performed; if $q = 1$, every node must perform a swap if possible; if $0 < q < 1$, a random subset of nodes may perform swaps). In step 3.2 of the SRS protocol, the condition $A_{jk} = 0$ ensures that swaps will generally not connect physical neighbors.

In step 5, we remove links that have too low fidelity since they were produced after swapping too many shorter links. To stop these links from forming in the first place, we would need to consider a more complex swapping policy where nodes are allowed to coordinate their actions (or a simple policy where communication is assumed to be instantaneous).

In Table I we provide a summary of the network and protocol parameters. In the next section we present our performance metrics and how to use them to tune the protocol parameter(s) for an optimal performance. Note that our methods can be applied to any other (synchronous and non-synchronous) CD protocol.

IV. PERFORMANCE EVALUATION OF CD PROTOCOLS

As previously discussed, a CD protocol must ensure that as many pairs of nodes as possible share entangled links, such that they can run quantum applications at any time. Ideally, the protocol should also provide many links between each pair of nodes, as this would allow them to run more demanding applications (e.g., applications that consume entanglement at a high rate) or to have spare links to run sporadic one-time

Algorithm 1. SRS entanglement generation protocol.

Inputs:

- Quantum network with an arbitrary configuration of entangled links and
 - physical adjacency matrix A ;
 - probability of successful entanglement generation p_{gen} ;
 - probability of successful swap p_s ;
 - maximum swap distance M ;
 - probability of link consumption p_{cons} .
- q : probability of performing a swap.

Outputs:

- Quantum network with updated configuration of links.

Algorithm:

- 1: **Cutoffs** are applied and old links are removed.
- 2: **Entanglement generation** is attempted at every physical link if enough qubits are available. One entangled link is generated at each physical link with probability p_{gen} .
- 3: **Swaps** are performed. Every node i does the following, in parallel to each other:
 - 3.1: Pick at random a qubit entangled to some qubit in another node j .
 - 3.2: Pick at random a qubit entangled to some qubit in node $k \neq j$, and with $A_{jk} = 0$. If not possible, go to step 4.
 - 3.3: With probability q , perform a swap on both qubits, which succeeds with probability p_s . If it fails, both links involved in the swap are discarded.
- 4: **Classical communication**: every node gains updated information about every qubit (where it is connected to) and about every entangled link (link age and number of swaps used to create the link).
- 5: **Long links removal**: links that were produced as a consequence of swapping more than M elementary-level links are removed.
- 6: **Consumption**: each pair of nodes that share links consume one of them with probability p_{cons} .

applications. These notions of a good CD protocol motivate the definition of the following performance metrics.

Definition 2. In a quantum network, the *virtual neighborhood* of node i , $V_i(t)$, is the set of nodes that share an entangled link with node i at time t . Two nodes are *virtual neighbors* if they share at least one entangled link. The *virtual neighborhood size* is denoted as $v_i(t) := |V_i(t)|$.

Definition 3. In a quantum network, the *virtual node degree* of node i , $k_i(t)$, is the number of entangled links connected to node i at time t .

The virtual neighborhood size and the virtual node degree combined are useful metrics to evaluate the performance of a CD protocol. The size of the virtual neighborhood of node i corresponds to the number of nodes that can run background applications together with node i . Since our model includes consumption of entanglement in such applications, the virtual degree provides information about how many resources are left to run sporadic applications.

The definitions above are similar to the notions of node neighborhood and node degree in classical graph theory. However, the configuration of entangled links changes over time, and therefore performance metrics from graph theory are ill suited for this problem, as they generally do not include this

TABLE I. Parameters of the quantum network. The number of nodes is given by the size of the adjacency matrix A . When considering a (d, k) -tree topology, the adjacency matrix A can be replaced by d and k . The cutoff time t_{cut} is given by p_{gen} , F_{new} , F_{app} , and M via (3).

<i>Physical topology</i>	
A	Physical adjacency matrix
<i>Hardware</i>	
p_{gen}	Probability of successful heralded entanglement generation
F_{new}	Fidelity of newly generated entangled links
p_{swap}	Probability of successful entanglement swap
r	Number of qubits per node per physical neighbor
<i>Software (application related)</i>	
F_{app}	Minimum fidelity to run background applications
M	Maximum number of short-distance links involved in a sequence of swaps
p_{cons}	Probability that two nodes sharing some links consume one of them in each time slot
<i>CD protocol</i>	
q	Probability of performing swaps according to the SRS protocol

type of time dependence. In contrast to those metrics, $v_i(t)$ and $k_i(t)$ are not random variables but stochastic processes, i.e., the value at each time slot is a random variable.

When consuming entanglement at a constant rate, the steady state of the system is of particular interest since it will provide information about the performance of the protocols in the long term. In Appendix B, we show that, when running the SRS protocol (Algorithm 1), the network undergoes a transient state and then reaches a unique steady-state regime (the proof also applies to similar CD protocols that use heralded entanglement generation, entanglement swaps, and cutoffs). In what follows, we will focus on evaluating the performance of the protocol during the steady state via the steady-state expected value of the virtual neighborhood size, $v_i \equiv \lim_{t \rightarrow \infty} \mathbb{E}[v_i(t)]$, and the virtual node degree, $k_i \equiv \lim_{t \rightarrow \infty} \mathbb{E}[k_i(t)]$.

Next, in Sec. IV A, we analyze the behavior of v_i and k_i in the absence of swaps. In Sec. IV B, we analyze the relationship between these metrics and the protocol parameter q in a treelike network (although our methods are general and apply to any arbitrary topology) and we find the optimal q that maximizes the virtual neighborhood size of the nodes in the lowest level of the tree. In Sec. IV C, we provide a mathematical framework, based on Pareto optimization, to provide a good quality of service in heterogeneous networks.

A. No swaps

To gain some intuition about the dynamics of the network and to set a benchmark, we consider the SRS protocol with $q = 0$, i.e., no swaps. In the absence of swaps, only physical neighbors can share entanglement, and the virtual neighbor-

hood size and the virtual node degree of node i in the steady state are given by

$$v_i \equiv \lim_{t \rightarrow \infty} \mathbb{E}[v_i(t)] = d_i \frac{1 - \frac{1-p_{\text{cons}}}{1-p_{\text{gen}}} \lambda^r}{1 - \frac{p_{\text{cons}}}{p_{\text{gen}}} \lambda^r}, \quad (4)$$

$$k_i \equiv \lim_{t \rightarrow \infty} \mathbb{E}[k_i(t)] = d_i p_{\text{gen}} \frac{r + \frac{p_{\text{cons}}(1-p_{\text{cons}})}{p_{\text{gen}} - p_{\text{cons}}} (\lambda^r - 1)}{p_{\text{gen}} - p_{\text{cons}} \lambda^r}, \quad (5)$$

where $\lambda \equiv \frac{p_{\text{cons}}(1-p_{\text{gen}})}{p_{\text{gen}}(1-p_{\text{cons}})}$; p_{gen} is the probability of successful entanglement generation at the physical link level; p_{cons} is the link consumption rate; d_i is the physical node degree of node i ; and r is the number of qubits available at node i per physical neighbor. Equations (4) and (5) are derived in Appendix C using general random walks. Note that in the derivation we assume large enough cutoffs, such that links are consumed with a high enough probability before reaching the cutoff time.

In the absence of swaps, both v_i and k_i are proportional to the physical node degree d_i but independent of the rest of the physical topology. This allows us to study these performance metrics without assuming any specific physical topology. Figure 4 shows the analytical solution for v_i and k_i when each node has five qubits per physical neighbor ($r = 5$). The figure shows a transition from large to small virtual neighborhood size when increasing p_{cons} beyond p_{gen} . When the consumption rate is smaller than the generation rate, the size of the virtual neighborhood saturates and converges to the number of physical neighbors. When p_{cons} increases beyond p_{gen} , the virtual neighborhood size goes to zero. A similar behavior is observed for the virtual degree, which takes larger values for $p_{\text{cons}} < p_{\text{gen}}$. The same behavior is observed for different values of r , as shown in Appendix C.

We conclude that, when the consumption rate is below the generation rate and the cutoffs are large enough, each node can produce sufficient entangled links with its neighboring nodes for background applications and an extra supply of links for sporadic applications.

In Appendix C, we use simulations to show that $\mathbb{E}[v_i(t)]$ and $\mathbb{E}[k_i(t)]$ indeed converge to the steady-state values predicted by our analytical calculations as t goes to infinity.

B. Homogeneous set of users

Let us now consider a more general setting: the SRS protocol with $q > 0$. Nodes are now allowed to perform swaps with some probability q . In this setup, a natural question arises: what value of q should we choose to achieve the best performance?

First, let us recall how we measure the performance. We use the expected virtual neighborhood size in the steady state, v_i , to determine the number of nodes that can run applications with node i in the background. We want to maximize v_i . The expected node degree k_i determines the number of additional entangled links that can be used for sporadic applications. Whenever possible, we will try to have a large k_i too, although maximizing k_i is not the purpose of a CD protocol (in fact, k_i is maximized when no swaps are performed, since they always reduce the total number of entangled links, even when they are successful). In what follows, we show how to optimize

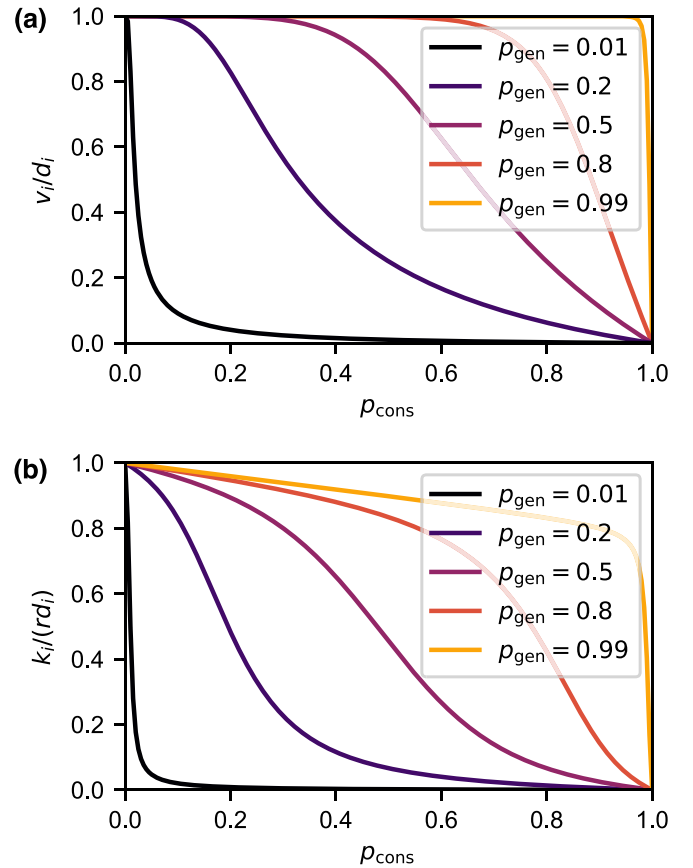


FIG. 4. Larger consumption rates decrease the virtual neighborhood size and the virtual node degree. Expected virtual neighborhood size (a) and virtual node degree (b) in the steady state in a quantum network with no swaps, with cutoff $t_{\text{cut}} = 10/p_{\text{cons}}$ time steps, and with five qubits per node per physical channel ($r = 5$). Both quantities are normalized by the physical degree of node i , d_i . The curves were calculated using (4) and (5).

the SRS protocol in a quantum network with a (2,3)-tree topology (although our methods apply to any quantum network and any CD protocol). This tree network is particularly interesting because it corresponds to a dumbbell network, which could be used to model users (level-2 nodes) in two metropolitan areas (level-1 nodes) connected by a central link via the level-zero node. If we assume distances of the order of 10 km, the communication time over optical fibers is of the order of 1 ms. Hence, the time step must be at least of the order of 1 ms. For demonstration purposes, we assume a coherence time of $T = 2000$ time steps, which is of the order of 1 s. As a reference, state-of-the-art coherence times lie between milliseconds (e.g., $T \approx 11.6$ ms in the NV centers experiment from Ref. [19]) and seconds (e.g., $T \approx 50$ s in the trapped-ion experiment from Ref. [67]). Additionally, also for demonstration purposes, we assume probabilistic entanglement generation, deterministic swaps, maximum swap distance $M = 4$ (such that every node can share links with every other node), and background applications that can be executed with low fidelity links ($F_{\text{app}} = 0.6$). We analyzed the system by simulating the evolution of the network over time and using Monte Carlo sampling. For further details about

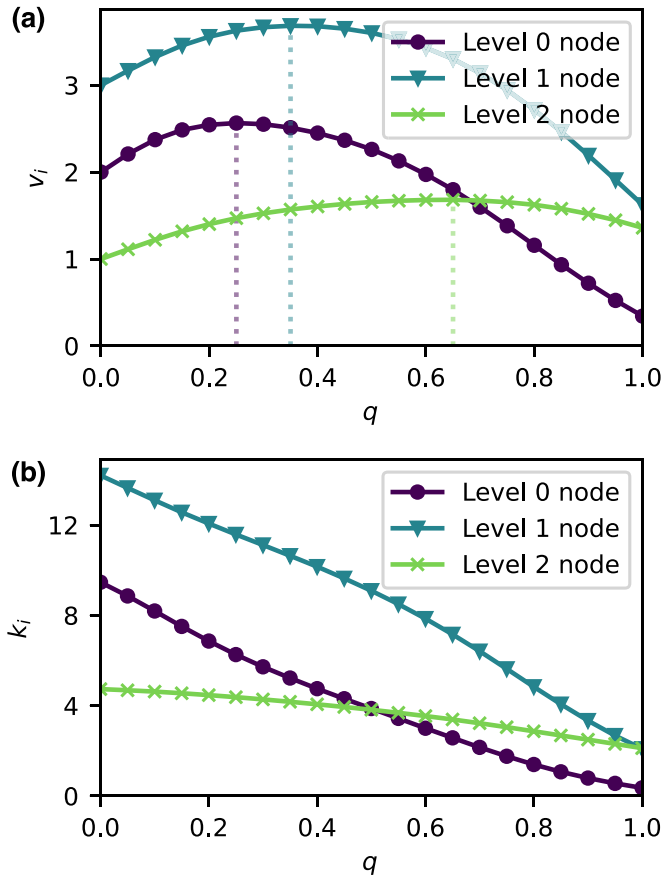


FIG. 5. The virtual neighborhood size of every node cannot be maximized simultaneously. Expected virtual neighborhood size (a) and virtual node degree (b) in the steady state in a (2,3)-tree network running the SRS protocol vs the protocol parameter q . The value of q that maximizes the virtual neighborhood size, indicated by the dotted lines, is node dependent. The virtual node degree decreases monotonically with increasing q , since more links are consumed in swaps when q is large. Other parameter values used in this experiment: $p_{\text{gen}} = 0.9$, $F_{\text{new}} = 0.888$, $p_{\text{swap}} = 1$, $r = 5$, $T = 2000$ time steps, $M = 4$, $p_{\text{cons}} = p_{\text{gen}}/4 = 0.225$, $F_{\text{app}} = 0.6$, $t_{\text{cut}} = 56$ time steps [given by (3)]. Results are obtained using a network simulation and Monte Carlo sampling with 10^6 samples. Error bars are not shown since they are smaller than the line width—the standard errors are below 0.003 and 0.006 for the v_i and k_i , respectively. The standard error is defined as $2\hat{\sigma}/\sqrt{N_{\text{samples}}}$, where $\hat{\sigma}$ is the sample standard deviation and N_{samples} is the number of samples.

how we find the steady state and how we compute expectation values from simulation data see Appendix D.

Figure 5 shows v_i and k_i for three different nodes. Due to the symmetry of the topology, every node in the same level of the tree has the same statistical behavior. Therefore, we can describe the behavior of the whole tree network by looking at one node per level. When no swaps are performed ($q = 0$), the virtual neighborhood size v_i [Fig. 5(a)] is upper bounded by the number of physical neighbors d_i ($d_i = 2, 3$, and 1, for nodes in level 0, 1, and 2, respectively). Increasing q leads to an increase in v_i , which reaches a maximum value before decreasing again. If too many swaps are performed (q close to 1), then v_i decreases, since each swapping operation

consumes two links and produces only one. The maximum virtual neighborhood size, $\max_q v_i$, is achieved at a different value of q for each node. The virtual node degree k_i [Fig. 5(b)] behaves qualitatively in a similar way for every node: it is maximized at $q = 0$ and, as we perform more swaps (increasing q), more links are swapped and fewer links remain in the system. A similar behavior was observed for larger trees and for probabilistic swaps (see Appendix E).

In some cases, we may be only interested in providing a good service to a subset of nodes U , the *user nodes*. The users run applications but also perform swaps to support the entanglement distribution among other pairs of users. The only purpose of the rest of the nodes (*repeater nodes*) is to aid the users to meet their needs. In the literature, users that consume entanglement, but do not perform swaps to help other nodes, are generally called *end nodes*. Here we assume every node is a user or a repeater node. When some nodes are users and some are repeaters, the performance metrics of repeater nodes become irrelevant and we want to maximize v_i , $\forall i \in U$. When the set of users is homogeneous (i.e., all user nodes have the same properties), the statistical behavior of all users is the same and we can formulate a single-objective optimization problem where we want to maximize v_i for a single $i \in U$. For example, in a tree quantum network, users are generally the nodes at the lowest level [53]. In the example from Fig. 5, the lowest-level nodes are the level-2 nodes (green line with crosses). If the level-2 nodes are the only users, the performance of the protocol is optimized for $q \approx 0.65$, which maximizes their v_i . The protocol optimization problem also becomes a single-objective optimization problem in other networks with a strong symmetry, such as regular networks [68].

In Fig. 6 we consider a (2,3)-tree network where the users are the nodes at the lowest level, and we study the influence of the background application requirements (F_{app} and p_{cons}) on the maximum expected virtual neighborhood size of the users. Here, we assume that the entanglement generation rate is much larger than the consumption rate ($p_{\text{gen}} \geq 3p_{\text{cons}}$). Otherwise, links are consumed shortly after they are generated and the behavior of the system is not interesting, as discussed in Sec. IV A. From the figure, we observe that the consumption rate has a stronger effect on the virtual neighborhood. For example, for $F_{\text{app}} = 0.8$, decreasing p_{cons} from 0.3 to 0.1 increases the maximum expected virtual neighborhood size by 20.3%. However, when decreasing F_{app} from 0.8 to 0.5, the maximum increase in v_i is 3.5% (for $p_{\text{cons}} = 0$). The consumption rate has a bigger effect on the virtual neighborhood because it directly impacts the configuration of virtual links, while F_{app} only affects links via the cutoff. In this case, the smallest cutoff is 17 time steps for $F_{\text{app}} = 0.8$ and the largest is 411 time steps for $F_{\text{app}} = 0.5$. When the generation rate is large, virtual neighbors are likely to share multiple entangled links. In that case, cutoffs barely impact the virtual neighborhood size since links can be regenerated quickly and they are only removed after some time t_{cut} . However, link consumption can still have a strong impact on the virtual neighborhood size since any link can be consumed at any time step. If the cutoffs are very close to unity (e.g., when applications require a fidelity $F_{\text{app}} > 0.8$), the cutoff value may strongly affect the virtual neighborhood size.

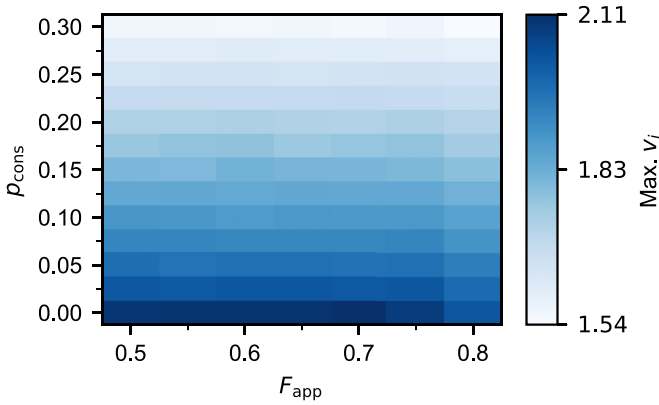


FIG. 6. The consumption rate has a stronger impact on the performance than the application fidelity when the entanglement generation rate is high. Maximum virtual neighborhood size (maximized over q) of a layer-2 node in a (2,3)-tree network vs the application fidelity, F_{app} , and the consumption rate, p_{cons} . Other parameter values used in this experiment: $p_{\text{gen}} = 0.9$, $F_{\text{new}} = 0.95$, $p_{\text{swap}} = 1$, $r = 5$, $T = 2000$ time steps, $M = 4$. The cutoff time t_{cut} is given by (3). Results are obtained using a network simulation and Monte Carlo sampling with 10^4 samples. The maximum error is 0.015 (the error is defined as $2\hat{\sigma}/\sqrt{N_{\text{samples}}}$, where $\hat{\sigma}$ is the sample standard deviation and N_{samples} is the number of samples). Note that $\max_q(v_i)$ should be monotonic in F_{app} and p_{cons} but in this plot we observe small deviations due to the sample size.

C. Heterogeneous set of users and multiobjective optimization

In a more general topology, the user nodes may have different properties and different physical degrees. In that case, the size of the virtual neighborhood of each user may be maximized for a different value of q . Hence, optimizing the protocol for node i generally means that the protocol will be suboptimal for some other node $j \neq i$. This leads to a multiobjective optimization problem where we must find a tradeoff between the variables that we want to maximize. In such a problem, optimality can be defined in different ways [69]. A practical definition is the Pareto frontier.

Definition 4. Let U be the set of user nodes. Let $\vec{\theta} \in \Theta$ be a combination of parameter values describing the topology, the hardware, and the software of the quantum network, where Θ is the parameter space. Let $v_i(\vec{\theta})$, with $i \in U$, be the set of variables that we want to maximize. The *Pareto frontier* is defined as

$$P = \{\vec{\theta} \mid \forall \vec{\theta}' \in \Theta \exists i \text{ s.t. } v_i(\vec{\theta}) \geq v_i(\vec{\theta}')\}. \quad (6)$$

Lemma 1. If the parameter space is nonempty, i.e., $\Theta \neq \emptyset$, then the Pareto frontier is nonempty, i.e., $P \neq \emptyset$.

Proof. If $\Theta \neq \emptyset$, there exists some $\vec{\theta}_j = \text{argmax}_{\vec{\theta} \in \Theta} (v_j(\vec{\theta}))$, for any $j \in U$. Then, $v_j(\vec{\theta}_j) \geq v_j(\vec{\theta}')$, $\forall \vec{\theta}' \in \Theta$, which means that $\vec{\theta}_j \in P$. Since $\vec{\theta}_j$ always exists, we conclude that $P \neq \emptyset$. ■

Note that the parameter space Θ can be a constrained space, i.e., it does not necessarily include all combinations of parameters values. For example, combinations of parameters that are experimentally unfeasible may be excluded from Θ . The Pareto frontier achieves a tradeoff in maximizing every v_i , $i \in U$. For all the points $\vec{\theta}$ in the Pareto frontier, we cannot

obtain an increase in $v_i(\vec{\theta})$ without decreasing or keeping constant some other $v_j(\vec{\theta})$. Moreover, Lemma 1 ensures that there is at least one point $\vec{\theta}$ in the Pareto frontier.

Note that the Pareto frontier may allow situations in which the distribution of entangled links is not equitable (e.g., one user may maximize its virtual neighborhood size at the expense of another user minimizing it). To avoid such situations, we can explicitly take into account quality-of-service requirements from every user node. An example of simple requirement from node i is to have some minimum number of virtual neighbors c_i . Then, the set of points that meet the quality-of-service requirements can be written as

$$Q = \{\vec{\theta} \mid v_i(\vec{\theta}) \geq c_i\}. \quad (7)$$

An example of more specific requirement is to keep the number of entangled links between two specific nodes always above a certain threshold.

Definition 5. The *optimal region* P^* is the set of parameters that are in the Pareto frontier and meet the quality-of-service requirements, i.e.,

$$P^* = P \cap Q, \quad (8)$$

where P is the Pareto frontier and Q is the set of points that meet the quality-of-service requirements.

As an example, we consider a (2,3)-tree network where the nodes in levels 1 and 2 are users. Due to the symmetry of the topology, we only need to explicitly optimize v_i for one node in each level. In this case, it is possible to provide a graphical representation of the Pareto frontier and the optimal region. Figure 7 shows the expected virtual neighborhood size in the steady state for a level-1 user and a level-2 user in a quantum network with a (2,3)-tree topology running the SRS protocol with probabilistic entanglement generation, deterministic swaps, and entanglement consumption at a fixed rate. Each data point corresponds to a different value of the protocol parameter q . The data points highlighted with blue crosses form the Pareto frontier P . In this example, we want the users in the first and second level to have an expected virtual neighborhood size larger than 3 and 1.6, respectively. Then,

$$Q = \{\vec{\theta} \mid v_1(\vec{\theta}) \geq 3, v_2(\vec{\theta}) \geq 1.6\}. \quad (9)$$

The regions shaded in red correspond to forbidden regions where the quality-of-service requirements are not met. That is, the points in the white region are in Q . The data points in the optimal region P^* are the blue crosses in the white region. This corresponds to $q \in [0.4, 0.65]$. All these values of q can be considered optimal, as they are part of the Pareto frontier and meet the minimum user requirements.

As a final remark, note that we have used this multiobjective optimization framework to optimize the performance of a single-parameter CD protocol. However, it can also be used to choose from several CD protocols. This method can be applied to heterogeneous quantum networks with arbitrary topologies.

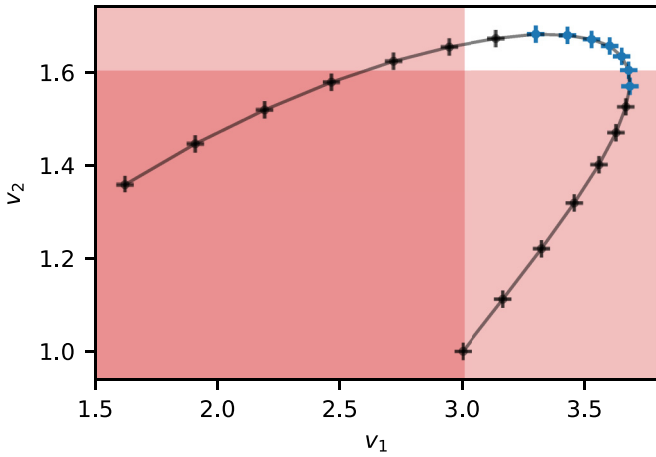


FIG. 7. The optimal region determines the combinations of parameters that provide an optimal performance. Virtual neighborhood size of a level-1 node, v_1 , and a level-2 node, v_2 , in a (2,3)-tree network running the SRS protocol for different values of the protocol parameter q (for $q = 0$, we have $v_1 = 3$ and $v_2 = 1$; we increase q in intervals of 0.05 following the black line up to $q = 1$). The data points with blue crosses form the Pareto frontier P . The regions shaded in red are forbidden by the quality-of-service requirements ($c_1 = 3$, $c_2 = 1.6$). The optimal region P^* is formed by the blue crosses in the white region. Other parameter values used in this experiment: $p_{\text{gen}} = 0.9$, $F_{\text{new}} = 0.888$, $p_{\text{swap}} = 1$, $r = 5$, $T = 2000$ time steps, $M = 4$, $p_{\text{cons}} = p_{\text{gen}}/4 = 0.225$, $F_{\text{app}} = 0.6$, $t_{\text{cut}} = 56$ time steps [given by (3)]. Results are obtained using a network simulation and Monte Carlo sampling with 10^6 samples. Error bars are not shown since they are smaller than the linewidth—the standard errors are below 0.003 and 0.002 for v_1 and v_2 , respectively. The standard error is defined as $2\hat{\sigma}/\sqrt{N_{\text{samples}}}$, where $\hat{\sigma}$ is the sample standard deviation and N_{samples} is the number of samples.

V. DISCUSSION

In this paper we have introduced metrics to evaluate the performance of protocols for continuous distribution of entanglement. The virtual neighborhood of a node is the set of nodes that share entanglement with the node, and the virtual degree of a node is the number of entangled states it shares with other nodes. The goal of the protocol is to maximize the size of the virtual neighborhood of every user. Here, as an example, we have considered a simple tree network and we have demonstrated how to formulate a single-objective and a multiobjective optimization problem that can be used to optimize the performance when the set of users is homogeneous and heterogeneous, respectively.

In our calculations, we assumed that background applications consume entanglement at a given rate. We found that, when the entanglement generation rate is large, the consumption rate has a stronger impact on the size of the virtual neighborhood than the fidelity requirements imposed by the quantum applications.

Our formulation also allows the study of protocols that continuously distribute entanglement to maintain a supply of high quality preshared entanglement. Specifically, the SRS protocol described in Algorithm 1 delivers preshared entanglement when the consumption rate is set to zero. This can be useful to determine the feasibility of quantum network

protocols that assume preshared entanglement among the nodes of the network. In this case, an application that uses the available entanglement would disrupt the distribution of entangled states and would bring the system to a new transient state. Hence, an additional useful metric would be the time required to converge to a steady state after such a disruption. We leave this analysis for future work.

We also leave the generalization of the network model and the protocol as future work. As an example, one can consider nodes that have a pool of qubits that can be used for any purpose, instead of having r specific qubits that can generate entanglement with each physical neighbor. One can also define node-dependent protocols, where each node follows a different set of instructions.

Lastly, note that we expect the coexistence of protocols for on-demand and continuous distribution of entanglement in large-scale quantum networks. Continuous distribution can be used to supply entanglement to applications running at a constant rate while on-demand distribution can support this process during peak demands from sporadic applications.

Our code and the data shown in this paper can be found in Ref. [70].

ACKNOWLEDGMENTS

We thank B. Davies, T. Coopmans, and G. Vardoyan for discussions and feedback. We also thank J. van Dam, B. van der Vecht, and L. Talsma for feedback on this paper. Á.G.I. acknowledges financial support from the Netherlands Organisation for Scientific Research (NWO/OCW), as part of the Frontiers of Nanoscience program. S.W. acknowledges support from a European Research Council Starting Grant.

Á.G.I. defined the project, analyzed the results, and prepared this paper. S.W. supervised the project and provided active feedback at every stage of the project.

APPENDIX A: FURTHER DETAILS ON THE NETWORK MODEL

1. Entanglement swap

Two nodes that are not physical neighbors cannot generate entanglement directly between them. Instead, they rely on entanglement swap operations to produce a shared entangled state between them [30–32]. As an example, consider two end nodes A and B , which are not physically connected but share a physical link with an intermediate node C . To generate an entangled link between A and B , they need to first generate entangled links between A and C , and also between C and B . Then, node C can perform a Bell state measurement to transform links A - C and C - B into a single entangled link between A and B . When both input links are Werner states with fidelities F_1 and F_2 , the output state in a swap operation is also a Werner state with fidelity [62]

$$F_{\text{swap}}(F_1, F_2) = F_1 F_2 + \frac{(1 - F_1)(1 - F_2)}{3}. \quad (\text{A1})$$

Note that this operation generally decreases the fidelity: $F_{\text{swap}}(F_1, F_2) \leq F_1, F_2$.

Additionally, entanglement swaps can be either probabilistic [31,71,72] or deterministic [19], depending on the hardware employed. With probability p_s , the swap operation succeeds and both input states are consumed to produce a single entangled link. With probability $1 - p_s$, the swap operation fails: both input states are consumed but no other entangled state is produced.

2. Purification

If the fidelity of an entangled state is not large enough for a specific application, nodes can run a purification protocol to increase its fidelity. In general, these protocols take as input multiple entangled states and output a single state with larger fidelity [10,60,61].

For simplicity, we do not consider any kind of purification in our analysis. Nevertheless, it is possible to integrate purification of entangled states at the physical link level into our model by decreasing the value of p_{gen} , to account for all the states that must be prepared in advance to perform the purification protocol. This would also impact the fidelity of newly generated links, F_{new} , which would correspond now to the fidelity of the links after purification at the physical link level. The cutoff time would also need to be adjusted, since the time step would take a longer time (it would have to include more than one entanglement generation attempt). If physically distant nodes require larger fidelity links, they can run a purification subroutine as part of the application once they have generated enough entangled links.

3. Cutoff times

Quantum states decohere, mainly due to environmental couplings [55,63]. Decoherence decreases the fidelity of states over time. We consider a depolarizing noise model, which is a worst-case scenario (other types of noise can be converted to depolarizing noise via twirling [55,73,74]). As shown in Appendix A from Ref. [12], if we assume that each qubit of a Werner state is stored in a different memory and experiences depolarizing noise independently, the fidelity of the Werner states evolves as

$$F(t + \Delta t) = \frac{1}{4} + \left(F(t) - \frac{1}{4}\right)e^{-\frac{\Delta t}{T}}, \quad (\text{A2})$$

where $F(t)$ is the fidelity of the state at time t , Δt is an arbitrary interval of time, and T is a parameter that characterizes the exponential decay in fidelity of the whole entangled state.

When the fidelity of the entangled links drops below some threshold, they are no longer useful. Hence, a common practice is to discard states after a cutoff time t_{cut} to prevent wasting resources on states that should not be used anymore [34]. We refer to the time passed since the creation of a quantum state as the age of the state. Whenever the age of an entangled state equals the cutoff time, the state is removed, i.e., the qubits involved are reset. As shown in Ref. [12], to ensure that any two nodes that are at most M physical links away will only share entangled states with fidelity larger than F_{app} , the cutoff time must satisfy

$$t_{\text{cut}} \leq -T \ln \left[\frac{3}{4F_{\text{new}} - 1} \left(\frac{4F_{\text{app}} - 1}{3} \right)^{1/M} \right], \quad (\text{A3})$$

where F_{new} is the fidelity of newly generated entangled links. This condition assumes that the output state in a swap operation takes the age of the oldest input link.

APPENDIX B: EXISTENCE OF A UNIQUE STEADY STATE

In this Appendix, we show that there is a unique steady-state value for the expected number of virtual neighbors and expected virtual degree of any node when a quantum network is running CD Protocol 1, under the assumption that entanglement generation is probabilistic ($p_{\text{gen}} < 1$).

We consider the stochastic processes $v_i(t)$ and $k_i(t)$, which correspond to the number of virtual neighbors of node i and the virtual degree of node i , respectively. The expected values over many realizations of the processes are denoted as $\mathbb{E}[v_i(t)]$ and $\mathbb{E}[k_i(t)]$.

The state of the network can be represented using the ages of all entangled links present in the network (the age is measured in number of time slots). This can be written as an array s with $\frac{1}{2}r \sum_{i=0}^{n-1} d_i$ components, since there are n nodes and each node i can store up to rd_i entangled links, where d_i is the physical degree of node i and r is a hardware-dependent parameter that limits the maximum number of qubits per node. Since we impose cutoff times on the memories, each of the components of this vector can only take a finite set of values. Let \mathcal{S} be the set of all possible states, which is also finite.

Given a state $s(t)$ at time $t \in \mathcal{N}$ (recall that we consider discrete time steps in our protocols), the transition to a new state only depends on the number of available memories at each node for generation of new links and on the number of available links for performing swaps and for consumption in applications. Hence, the transition does not depend on past information:

$$\begin{aligned} \Pr[s(t+1) = \sigma \mid s(0), s(1), \dots, s(t)] \\ = \Pr[s(t+1) = \sigma \mid s(t)]. \end{aligned}$$

Consequently, the state of the network can be modeled as a Markov chain with the following three properties.

(1) The chain is irreducible, since every state is reachable from every other state. If $p_{\text{gen}} < 1$, there is a nonzero probability that no links are generated over many time slots until all existing links expire due to cutoffs and therefore the network returns to the starting state with no links—from this initial state, every other state can be reached.

(2) The chain is aperiodic. A sufficient condition for an irreducible chain to be aperiodic is that $\Pr[s(t+1) = \sigma \mid s(t) = \sigma] > 0$ for some state $\sigma \in \mathcal{S}$ [75]. When entanglement generation is probabilistic ($p_{\text{gen}} < 1$), the state with no entangled links satisfies the previous condition (if all entanglement generation attempts fail, the network will remain in a state with no links), and therefore the chain is aperiodic.

(3) The chain is positive recurrent (i.e., the mean time to return to any state is finite), since it is irreducible and it has a finite state space \mathcal{S} (see Theorem 9.3.5 from Ref. [75]).

According to Theorem 9.3.6 from Ref. [75], from the three properties above we can conclude that there exists a unique steady-state probability distribution, i.e., the following limit exists: $\lim_{t \rightarrow \infty} \Pr[s(t) = \sigma]$, $\forall \sigma \in \mathcal{S}$.

Let us now compute the expected number of virtual neighbors in the steady state:

$$\begin{aligned}
v_i &\equiv \lim_{t \rightarrow \infty} \mathbb{E}[v_i(t)] \\
&= \lim_{t \rightarrow \infty} \sum_{v=0}^n v \Pr[v_i(t) = v] \\
&= \sum_{v=0}^n v \lim_{t \rightarrow \infty} \Pr[v_i(t) = v] \\
&= \sum_{v=0}^n v \lim_{t \rightarrow \infty} \sum_{\sigma \in \mathcal{S}} \Pr[v_i(t) = v \mid s(t) = \sigma] \Pr[s(t) = \sigma] \\
&= \sum_{v=0}^n v \sum_{\sigma \in \mathcal{S}} \lim_{t \rightarrow \infty} \Pr[v_i(t) = v \mid s(t) = \sigma] \lim_{t \rightarrow \infty} \Pr[s(t) = \sigma] \\
&= \sum_{\sigma \in \mathcal{S}} \lim_{t \rightarrow \infty} \Pr[s(t) = \sigma] \sum_{v=0}^n v \lim_{t \rightarrow \infty} \Pr[v_i(t) = v \mid s(t) = \sigma].
\end{aligned} \tag{B1}$$

Let us define a function $\kappa(s; i, j)$ that takes as input a state s and two node indices i and j . This function returns the number of entangled links shared by nodes i and j in state s . The virtual neighborhood size of node i at time t , $v_i(t)$, is given by the state of the network at time t , $s(t)$, and it can be written as

$$v_i(t) = v_i[s(t)] = \sum_{j \in V \setminus \{i\}} \min(1, \kappa(s(t); i, j)).$$

Consequently,

$$\begin{aligned}
\Pr[v_i(t) = v \mid s(t) = \sigma] \\
&= \begin{cases} 1, & \text{if } v = \sum_{j \in V \setminus \{i\}} \min(1, \kappa(\sigma; i, j)) \\ 0, & \text{otherwise} \end{cases}.
\end{aligned} \tag{B2}$$

Using (B2), we can write (B1) as

$$v_i = \sum_{\sigma \in \mathcal{S}} \lim_{t \rightarrow \infty} \Pr[s(t) = \sigma] \sum_{j \in V \setminus \{i\}} \min(1, \kappa(\sigma; i, j)). \tag{B3}$$

The expected virtual degree can be calculated similarly but using its corresponding definition, $k_i[s(t)] = \sum_{j \in V \setminus \{i\}} \kappa(s(t); i, j)$:

$$k_i \equiv \lim_{t \rightarrow \infty} \mathbb{E}[k_i(t)] = \sum_{\sigma \in \mathcal{S}} \lim_{t \rightarrow \infty} \Pr[s(t) = \sigma] \sum_{j \in V \setminus \{i\}} \kappa(\sigma; i, j). \tag{B4}$$

Since we have shown that the probability distributions that appear in (B3) and (B4) exist and are unique, then the quantities v_i and k_i also exist and are unique. That is, there is a unique steady-state value for the expected number of virtual neighbors and the expected virtual degree of any node i .

From our simulations, we also expect a unique steady state for $p_{\text{gen}} = 1$. The main difficulty in proving its existence is that the Markov chain is not always irreducible (the state with no links may not be reachable from some other states since links are generated at maximum rate). However, if one can show that there is a unique equivalence class (i.e., a unique set of states that are reachable from each other) that is reached

after a finite number of transitions, the derivation above may be applicable to this equivalence class, which would constitute an irreducible Markov chain.

Lastly, note that in practice one may find an initial transient state with periodic behavior. This happens in quasideterministic systems, i.e., systems in which all probabilistic events (e.g., successful entanglement generation) happen with probability very close to 1. In quasideterministic systems, all realizations of the stochastic processes are identical at the beginning with a very large probability. For some combinations of parameters, these processes may display a periodic behavior with a period on the order of the cutoff time. Over time, each realization starts to behave differently due to some random events yielding different outcomes. Consequently, the periodic oscillations will dephase, and they will cancel out after averaging over all realizations. In the example from Fig. 8, we find that both $\mathbb{E}[v_i(t)]$ and $\mathbb{E}[k_i(t)]$ are periodic with period approximately t_{cut} . The amplitude of the oscillations vanishes after a few periods.

APPENDIX C: ANALYTICAL PERFORMANCE METRICS IN THE ABSENCE OF SWAPS

In this Appendix, we consider a CD protocol with the same structure as the SRS protocol (see Algorithm 1 from the main text) in the absence of swaps and with a large enough cutoff time ($t_{\text{cut}} > r$ and $t_{\text{cut}} \gg \frac{1}{p_{\text{cons}}}$, where the cutoff is measured in number of time steps). As discussed in the main text, when no swaps are performed, we can derive closed-form expressions to gain some intuition about the dynamics of the network and to set a benchmark. Here, we show that the virtual neighborhood size and the virtual node degree of node i in the steady state are given by

$$v_i \equiv \lim_{t \rightarrow \infty} \mathbb{E}[v_i(t)] = d_i \frac{1 - \frac{1-p_{\text{cons}}}{1-p_{\text{gen}}} \lambda^r}{1 - \frac{p_{\text{cons}}}{p_{\text{gen}}} \lambda^r} \tag{C1}$$

and

$$k_i \equiv \lim_{t \rightarrow \infty} \mathbb{E}[k_i(t)] = d_i p_{\text{gen}} \frac{r + \frac{p_{\text{cons}}(1-p_{\text{cons}})}{p_{\text{gen}} - p_{\text{cons}}} (\lambda^r - 1)}{p_{\text{gen}} - p_{\text{cons}} \lambda^r}, \tag{C2}$$

where $\lambda \equiv \frac{p_{\text{cons}}(1-p_{\text{gen}})}{p_{\text{gen}}(1-p_{\text{cons}})}$; p_{gen} is the probability of successful entanglement generation at the physical link level; p_{cons} is the link consumption probability; d_i is the physical node degree of node i ; and r is the number of qubits per physical link available at each node.

We define w_{ij} as the number of entangled links shared between nodes i and j (similar to the definition of κ in Appendix B). In the absence of swaps, nodes i and j can only share entangled links if they are physical neighbors, since the only mechanism available is heralded entanglement generation. If i and j are not physical neighbors, then $w_{ij} = 0$. The entangled links shared between nodes i and j can be consumed in some application or discarded when applying cutoffs. However, we also assume that entangled links are always consumed before they reach the cutoff time, i.e., $t_{\text{cut}} \gg \frac{1}{p_{\text{cons}}}$ (cutoff measured in number of time steps). This assumption allows us to model w_{ij} using the general random walk shown in Fig. 9.

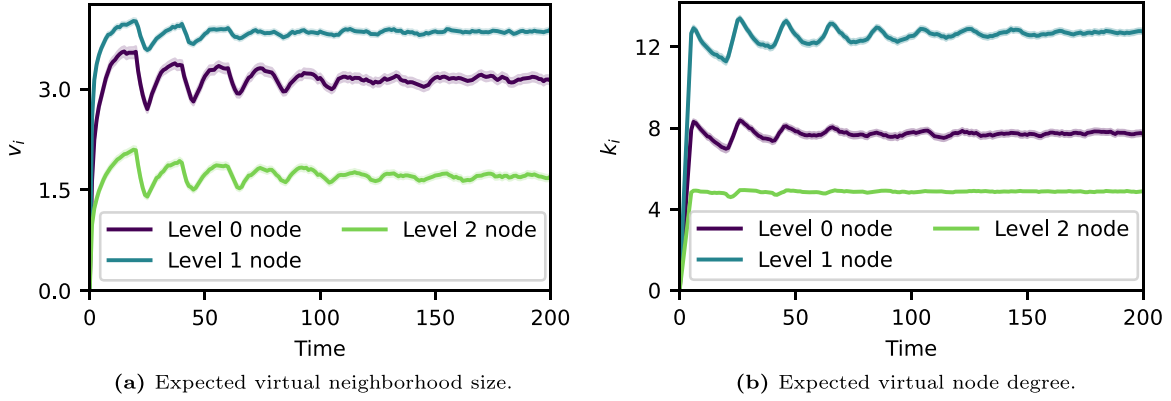


FIG. 8. A transient state with periodic oscillations may exist in quasideterministic systems. Evolution of v_i and k_i in a quantum network with a (2,3)-tree topology running the SRS protocol described in the main text. Each line (purple, blue, and green) corresponds to a node in a different level of the tree (level 0, 1, and 2). The error for each solid line is shown as a shaded region, although it is hard to notice since its maximum value is 0.040 in (a) and 0.084 in (b) (the error is defined as $2\hat{\sigma}/N_{\text{samples}}$, where $\hat{\sigma}$ is the sample standard deviation and N_{samples} is the number of samples). Other parameters used in this experiment: $p_{\text{gen}} = 0.99$, $F_{\text{new}} = 0.88$, $p_{\text{swap}} = 1$, $r = 5$, $T = 2000$ time steps, $M = 4$, $p_{\text{cons}} = 0.01$, $q = 0.2$, $F_{\text{app}} = 0.6$, $t_{\text{cut}} = 20$ time steps. Numerical results are obtained using a network simulation and Monte Carlo sampling with 10^3 samples.

(1) The maximum value for w_{ij} is the number of qubits available per physical link, r . This state is reachable even when entanglement generation is done sequentially, since links can be stored for longer than r time steps (we assume $t_{\text{cut}} > r$).

(2) The probabilities of transition forward are $p_k = p_{\text{gen}}(1 - p_{\text{cons}})$, $\forall k < r$, and $p_r = 0$.

(3) The probabilities of transition backwards are $q_0 = 0$, $q_k = p_{\text{cons}}(1 - p_{\text{gen}})$, $\forall 0 < k < r$, and $q_r = p_{\text{cons}}$.

(4) The no-transition probability is $z_k = 1 - p_k - q_k$, $\forall k$.

The steady-state probability distribution of this Markov chain is given by [75]

$$\lim_{t \rightarrow \infty} \Pr[w_{ij}(t) = w \mid A_{ij} = 1] = \begin{cases} (1 + \sum_{k=1}^r \prod_{m=0}^{k-1} \frac{p_m}{q_{m+1}})^{-1}, & w = 0 \\ (1 + \sum_{k=1}^r \prod_{m=0}^{k-1} \frac{p_m}{q_{m+1}})^{-1} \prod_{m=0}^{w-1} \frac{p_m}{q_{m+1}}, & w > 0 \end{cases}, \quad (\text{C3})$$

where A_{ij} is a binary variable that indicates if nodes i and j are physical neighbors ($A_{ij} = 1$) or not ($A_{ij} = 0$). After some algebra, the previous equation can be rewritten in terms of the original variables of the problem:

$$\lim_{t \rightarrow \infty} \Pr[w_{ij}(t) = w \mid A_{ij} = 1] = \begin{cases} \pi_0, & w = 0 \\ \pi_0 \rho^w, & 0 < w < r, \\ \pi_0 \rho^r (1 - p_{\text{gen}}), & w = r \end{cases}, \quad (\text{C4})$$

where

$$\pi_0 \equiv \frac{p_{\text{gen}} - p_{\text{cons}}}{(1 - p_{\text{gen}})(p_{\text{gen}} \rho^r - p_{\text{cons}})} \quad \text{and} \quad \rho \equiv \frac{p_{\text{gen}}(1 - p_{\text{cons}})}{p_{\text{cons}}(1 - p_{\text{gen}})}. \quad (\text{C5})$$

The expected value of w_{ij} is

$$\begin{aligned} \lim_{t \rightarrow \infty} \mathbb{E}[w_{ij}(t) \mid A_{ij} = 1] &= \lim_{t \rightarrow \infty} \sum_{w=0}^r w \Pr[w_{ij}(t) = w \mid A_{ij} = 1] \\ &= \sum_{w=0}^r w \lim_{t \rightarrow \infty} \Pr[w_{ij}(t) = w \mid A_{ij} = 1] \\ &\stackrel{a}{=} \pi_0 \sum_{w=0}^{r-1} w \rho^w + r \pi_0 \rho^r (1 - p_{\text{gen}}) \\ &= \pi_0 \frac{\rho - r \rho^r + (r-1) \rho^{r+1}}{(1-\rho)^2} + r \pi_0 \rho^r (1 - p_{\text{gen}}) \\ &= \frac{p_{\text{gen}}}{(p_{\text{gen}} - p_{\text{cons}})(p_{\text{gen}} \rho^r - p_{\text{cons}})} [r(p_{\text{gen}} - p_{\text{cons}}) \rho^r \\ &\quad + p_{\text{cons}}(1 - p_{\text{cons}})(1 - \rho^r)], \end{aligned} \quad (\text{C6})$$

where we have used (C4) in step a.

The virtual neighborhood size of node i is defined in terms of the variables w_{ij} as $v_i(t) = \sum_{j=1}^n \min(w_{ij}(t), 1)$, and the expectation value can be calculated as follows:

$$\begin{aligned} v_i &\equiv \lim_{t \rightarrow \infty} \mathbb{E}[v_i(t)] \\ &= \lim_{t \rightarrow \infty} \mathbb{E} \left[\sum_{j=1}^n \min(w_{ij}(t), 1) \right] \end{aligned}$$

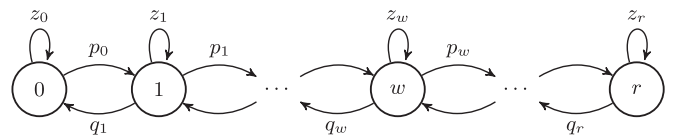


FIG. 9. General random walk modeling the number of entangled links w_{ij} between nodes i and j in the absence of swaps.

$$\begin{aligned}
&= \sum_{j=1}^n \lim_{t \rightarrow \infty} \mathbb{E}[\min(w_{ij}(t), 1)] \\
&\stackrel{a}{=} \sum_{j=1}^n \lim_{t \rightarrow \infty} \sum_{x=0}^r x \Pr[\min(w_{ij}(t), 1) = x] \\
&= \sum_{j=1}^n \sum_{x=0}^r x \lim_{t \rightarrow \infty} \Pr[\min(w_{ij}(t), 1) = x] \\
&\stackrel{b}{=} \sum_{j=1}^n \lim_{t \rightarrow \infty} \Pr[\min(w_{ij}(t), 1) = 1] \\
&= \sum_{j=1}^n \lim_{t \rightarrow \infty} \Pr[w_{ij}(t) > 0] \\
&\stackrel{c}{=} \sum_{j=1}^n \lim_{t \rightarrow \infty} \Pr(A_{ij} = 1) \Pr[w_{ij}(t) > 0 \mid A_{ij} = 1] \\
&\stackrel{d}{=} \sum_{j=1}^n A_{ij} \lim_{t \rightarrow \infty} \{1 - \Pr[w_{ij}(t) = 0 \mid A_{ij} = 1]\} \\
&\stackrel{e}{=} \sum_{j=1}^n A_{ij} (1 - \pi_0) \\
&\stackrel{f}{=} d_i (1 - \pi_0)
\end{aligned}$$

$$\begin{aligned}
&\stackrel{g}{=} d_i \frac{p_{\text{gen}}^{r+1} (1 - p_{\text{cons}})^r - p_{\text{gen}} (1 - p_{\text{gen}})^{r-1} p_{\text{cons}}^r (1 - p_{\text{cons}})}{p_{\text{gen}}^{r+1} (1 - p_{\text{cons}})^r - (1 - p_{\text{gen}})^r p_{\text{cons}}^{r+1}} \\
&= d_i \frac{1 - \frac{1-p_{\text{cons}}}{1-p_{\text{gen}}} \lambda^r}{1 - \frac{p_{\text{cons}}}{p_{\text{gen}}} \lambda^r}, \tag{C7}
\end{aligned}$$

where $\lambda \equiv \frac{p_{\text{cons}}(1-p_{\text{gen}})}{p_{\text{gen}}(1-p_{\text{cons}})}$, d_i is the physical degree of node i , and n is the total number of nodes, and with the following steps.

(a) We use the definition of expected value and the fact that $w_{ij}(t) \leq r$.

(b) We use the fact that $\min(w_{ij}(t), 1) \in \{0, 1\}$.

(c) We use the law of total probability, i.e., $\Pr(X) = \sum_n \Pr(Y_n) \Pr(X|Y_n)$. Moreover, if two nodes i and j are not physical neighbors ($A_{ij} = 0$), they cannot share any entangled links due to the absence of swaps, i.e., $\Pr[w_{ij}(t) > 0 \mid A_{ij} = 0] = 0$.

(d) Given the topology, A_{ij} is a binary variable with a fixed value. Therefore, $\Pr(A_{ij} = 1) = A_{ij}$.

(e) We use (C4).

(f) The physical node degree of node i can be computed as $d_i = \sum_{j=1}^n A_{ij}$.

(g) We use (C5).

The virtual degree of node i is defined in terms of the variables w_{ij} as $k_i(t) = \sum_{j=1}^n w_{ij}(t)$, and the expectation value can be calculated in a similar way to v_i :

$$\begin{aligned}
k_i &\equiv \lim_{t \rightarrow \infty} \mathbb{E}[k_i(t)] = \lim_{t \rightarrow \infty} \mathbb{E}\left[\sum_{j=1}^n w_{ij}(t)\right] = \lim_{t \rightarrow \infty} \sum_{j=1}^n \mathbb{E}[w_{ij}(t)] \\
&\stackrel{a}{=} \lim_{t \rightarrow \infty} \sum_{j=1}^n A_{ij} \mathbb{E}[w_{ij}(t) \mid A_{ij} = 1] \\
&\stackrel{b}{=} \lim_{t \rightarrow \infty} \mathbb{E}[w_{ij}(t) \mid A_{ij} = 1] \sum_{j=1}^n A_{ij} \\
&\stackrel{c}{=} d_i \lim_{t \rightarrow \infty} \mathbb{E}[w_{ij}(t) \mid A_{ij} = 1] \\
&\stackrel{d}{=} d_i p_{\text{gen}} \frac{r(p_{\text{gen}} - p_{\text{cons}}) p_{\text{gen}}^r (1 - p_{\text{cons}})^r + p_{\text{cons}} (1 - p_{\text{cons}}) [p_{\text{cons}}^r (1 - p_{\text{gen}})^r - p_{\text{gen}}^r (1 - p_{\text{cons}})^r]}{(p_{\text{gen}} - p_{\text{cons}}) [p_{\text{gen}}^{r+1} (1 - p_{\text{cons}})^r - p_{\text{cons}}^{r+1} (1 - p_{\text{gen}})^r]} \\
&= d_i p_{\text{gen}} \frac{r + \frac{p_{\text{cons}}(1-p_{\text{cons}})}{p_{\text{gen}}-p_{\text{cons}}} (\lambda^r - 1)}{p_{\text{gen}} - p_{\text{cons}} \lambda^r}, \tag{C8}
\end{aligned}$$

where $\lambda \equiv \frac{p_{\text{cons}}(1-p_{\text{gen}})}{p_{\text{gen}}(1-p_{\text{cons}})}$, and with the following steps.

(a) We use the law of total probability, i.e., $\Pr(X) = \sum_n \Pr(Y_n) \Pr(X|Y_n)$. Moreover, if two nodes i and j are not physical neighbors ($A_{ij} = 0$), they cannot share any entangled links due to the absence of swaps, i.e., $\Pr[w_{ij}(t) > 0 \mid A_{ij} = 0] = 0$. Given the topology, A_{ij} is a binary variable with a fixed value, therefore, $\Pr(A_{ij} = 1) = A_{ij}$.

(b) In a homogeneous network with no swaps, w_{ij} depends on A_{ij} but is otherwise independent of the nodes i and j . Hence, $\mathbb{E}[w_{ij}(t) \mid A_{ij} = 1]$ does not depend on j . This can also be seen in (C6).

(c) The physical node degree of node i can be computed as $d_i = \sum_{j=1}^n A_{ij}$.

(d) We use (C6).

Equations (C7) and (C8) can be used to study the performance of the protocol in the limit of large number of resources ($r \rightarrow \infty$). When $p_{\text{gen}} > p_{\text{cons}}$ we find

$$\lim_{r \rightarrow \infty} v_i = d_i, \quad \lim_{r \rightarrow \infty} k_i = \lim_{r \rightarrow \infty} r d_i = \infty. \tag{C9}$$

This means that, when the generation rate exceeds the consumption rate, the virtual neighborhood size will eventually saturate and every node will share entanglement with every

physical neighbor. In particular, the average number of entangled links will increase infinitely (for large but finite r , k_i reaches a maximum value of $\sim rd_i$). When $p_{\text{gen}} < p_{\text{cons}}$,

$$\lim_{r \rightarrow \infty} v_i = d_i \frac{p_{\text{gen}}(1 - p_{\text{cons}})}{p_{\text{cons}}(1 - p_{\text{gen}})},$$

$$\lim_{r \rightarrow \infty} k_i = d_i p_{\text{gen}} \frac{1 - p_{\text{cons}}}{p_{\text{cons}} - p_{\text{gen}}}. \quad (\text{C10})$$

In Fig. 10, we plot the expected virtual neighborhood size and expected virtual degree for different combinations of parameters, focusing on the interplay between p_{gen} and p_{cons} . Both quantities decrease with increasing consumption rate, as one would expect, and quickly drop to zero for $p_{\text{cons}} > p_{\text{gen}}$.

Lastly, Fig. 11 shows an example of the convergence of $\mathbb{E}[v_i(t)]$ and $\mathbb{E}[k_i(t)]$ to v_i and k_i over time, respectively. The time-dependent quantities have been calculated using a simulation on a quantum network with a (2,3)-tree physical topology. The dashed lines correspond to the steady-state values in the absence of swaps predicted by (C7) and (C8).

APPENDIX D: STEADY STATE OF A STOCHASTIC PROCESS

In this Appendix we provide an algorithm to find the steady-state expected value of a stochastic process given a set of samples. In our paper, we employ this algorithm to estimate the steady-state expected value of the virtual neighborhood size, $\lim_{t \rightarrow \infty} \mathbb{E}[v_i(t)]$, and the virtual node degree, $\lim_{t \rightarrow \infty} \mathbb{E}[k_i(t)]$, from numerical simulations.

Finding the steady state of a stochastic process using realizations of the process is not a trivial task. Algorithm 2 can be used to estimate the start of the steady state of a stochastic process given N realizations of the process. The algorithm ensures that the expected values of the process at any two times in the steady state are arbitrarily close with a large probability. We provide formal definitions and a proof below.

Theorem 1. Let $X(t) \in [a, b]$, with $a, b \in \mathbb{R}$, be a stochastic process with constant steady-state mean, i.e., $\lim_{t \rightarrow \infty} \mathbb{E}[X(t)] = X_\infty < \infty$. Let $\bar{X}_N(t_k)$ be a sample mean over N samples at time $t_k \in \{t_0, t_1, \dots, t_{M-1}\}$, with $t_0 < t_1 < \dots < t_{M-1}$. Consider a minimum size of the steady-state window w . When $N \rightarrow \infty$, Algorithm 2 with inputs $\bar{X}_N(t_k)$, a , b , and w finds α such that

$$\Pr[\mathbb{E}[X(t_i)] \in \text{IC}_{ij}] \geq 0.815, \quad \forall i, j \geq \alpha$$

for an interval of confidence $\text{IC}_{ij} = [\max(\bar{X}_N(t_i), \bar{X}_N(t_j)) - \varepsilon, \min(\bar{X}_N(t_i), \bar{X}_N(t_j)) + \varepsilon]$, with $\varepsilon = \frac{b-a}{\sqrt{N}}$, or the algorithm aborts.

Proof. Let us consider a stochastic process $X(t) \in [a, b]$ with constant steady-state mean, i.e., $\lim_{t \rightarrow \infty} \mathbb{E}[X(t)] = X_\infty < \infty$, and with finite variance $\sigma(t)^2$. Assume that we have N realizations of the process where we took samples at times $t_0 < t_1 < t_2, \dots$. We denote the value taken in realization $n \in \{0, 1, \dots, N-1\}$ at time t as $x_n(t)$. We define the sample average as

$$\bar{X}_N(t) = \frac{1}{N} \sum_{n=0}^{N-1} x_n(t). \quad (\text{D1})$$

Algorithm 2. Steady state estimation.

Inputs:

- $\bar{X}_N(t)$, $t = t_0, t_1, \dots, t_{M-1}$: sample mean of a stochastic process $X(t)$ over N realizations at $t = t_0, t_1, t_2, \dots, t_{M-1}$.
- a : minimum value of the stochastic process $X(t)$.
- b : maximum value of the stochastic process $X(t)$.
- w : minimum size of the steady state window.

Outputs:

- α : the steady state is assumed to start at $t = t_\alpha$. The protocol aborts if it is not possible to find an α such that $\alpha \leq M - w$.

Algorithm:

- 1: Define the error as $\varepsilon \leftarrow \frac{b-a}{\sqrt{N}}$.
- 2: Define the steady state window: $W \leftarrow \{M - w, M - w + 1, M - w + 2, \dots, M - 1\}$.
- 3: Calculate $\Delta_{ij} \leftarrow 2\varepsilon - |\bar{X}_N(t_i) - \bar{X}_N(t_j)|$, $\forall i, j \in W$ and $i \neq j$.
- 4: If $\Delta_{ij} < \frac{3}{2}\varepsilon$ for any i, j , then **abort** (steady state not found).
- 5: **for** z in $[1, 2, \dots, M-w]$ **do**
- 6: $k \leftarrow M - w - z$.
- 7: Calculate $\Delta_{ik} \leftarrow 2\varepsilon - |\bar{X}_N(t_i) - \bar{X}_N(t_k)|$, $\forall i \in W$.
- 8: If $\Delta_{ik} < \frac{3}{2}\varepsilon$ for any i , then $\alpha \leftarrow k + 1$ and go to step 12.
- 9: $W \leftarrow W \cup \{k\}$
- 10: **end for**
- 11: $\alpha \leftarrow k$.
- 12: **return** α .

The central limit theorem states that the distribution of the random variable $\sqrt{N}\{\bar{X}_N(t) - \mathbb{E}[X(t)]\}$ converges to a normal distribution $\mathcal{N}(0, \sigma(t)^2)$ as N approaches infinity. After rescaling and shifting this distribution, we find that $\mathbb{E}[X(t)]$ converges to a normal distribution $\mathcal{N}(\bar{X}_N(t), \sigma(t)^2/N)$ as N approaches infinity. By the properties of the normal distribution,

$$\Pr\left[\mathbb{E}[X(t)] \in \left(\bar{X}_N(t) - \frac{2\sigma(t)}{\sqrt{N}}, \bar{X}_N(t) + \frac{2\sigma(t)}{\sqrt{N}}\right)\right] > 0.95. \quad (\text{D2})$$

The values of $X(t)$ are constrained to the interval $[a, b]$, and therefore the standard deviation is upper bounded by [76]

$$\sigma(t) \leq (b - a)/2. \quad (\text{D3})$$

Let us define the error as $\varepsilon = \frac{b-a}{\sqrt{N}}$, and the interval of confidence for the expected value of $X(t_i)$ as

$$\text{IC}_i = (\bar{X}_N(t_i) - \varepsilon, \bar{X}_N(t_i) + \varepsilon). \quad (\text{D4})$$

Using (D2), (D3), and (D4), we can write

$$\Pr\{\mathbb{E}[X(t_i)] \in \text{IC}_i\} > 0.95. \quad (\text{D5})$$

This result means that the expected value is arbitrarily close to the sample mean with high probability. Next, we need to show that any two expected values in the time window defined by the algorithm are arbitrarily close to each other to conclude that the window captures the steady-state behavior.

Let us define the interval of confidence ij as the overlap in the intervals of confidence for the expected values of $X(t_i)$ and $X(t_j)$:

$$\text{IC}_{ij} = [\max(\bar{X}_N(t_i), \bar{X}_N(t_j)) - \varepsilon, \min(\bar{X}_N(t_i), \bar{X}_N(t_j)) + \varepsilon]. \quad (\text{D6})$$

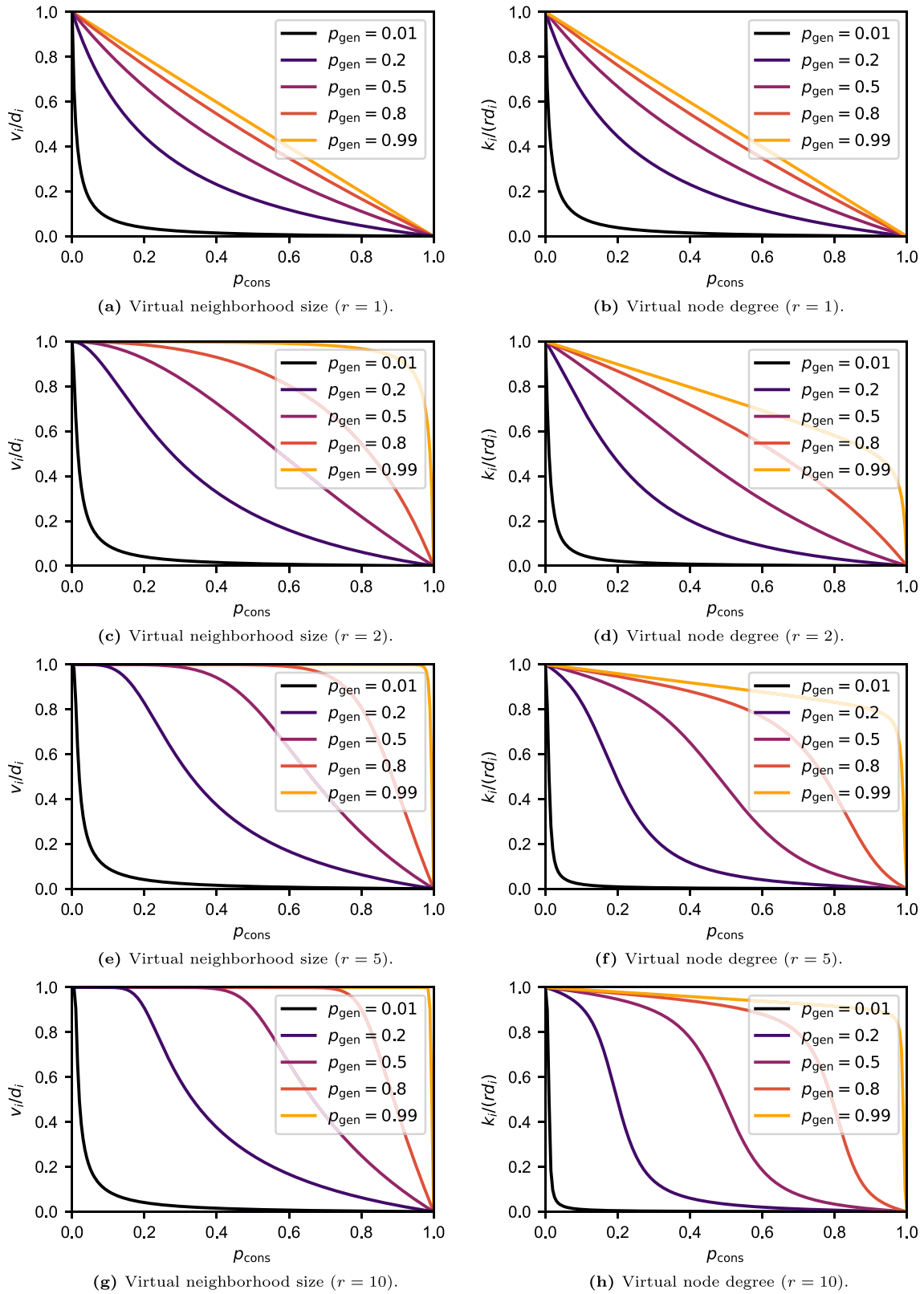


FIG. 10. Larger consumption rates decrease the virtual neighborhood size and the virtual node degree. Expected virtual neighborhood size (left) and virtual node degree (right) in the steady state in a quantum network with no swaps, cutoff $t_{\text{cut}} = 10/\rho_{\text{cons}}$ time steps, and $r = 1, 2, 5, 10$ qubits per node per physical channel (from top to bottom). All curves were calculated using Eqs. (C7) and (C8).

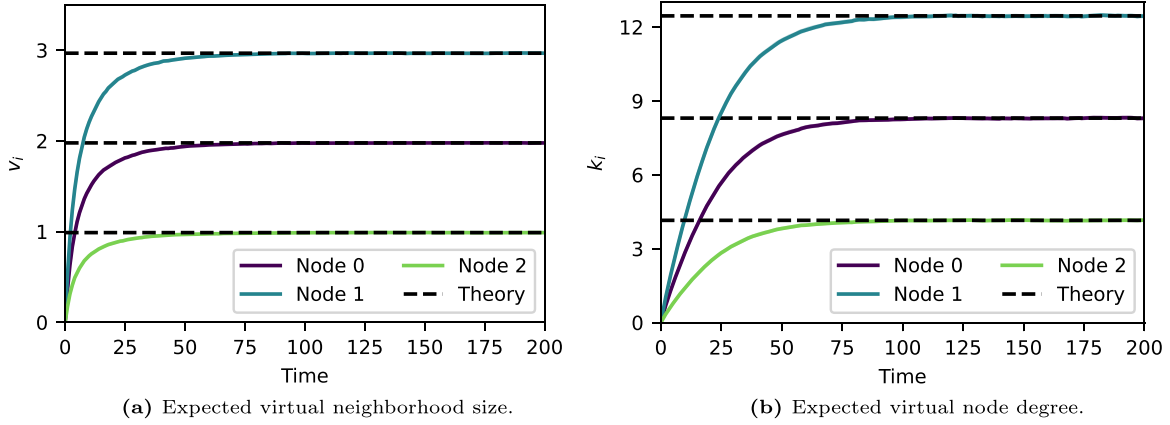


FIG. 11. The expected virtual neighborhood size and the expected virtual node degree converge to the steady-state analytical values in the absence of swaps. In this example, we ran the SRS protocol (Algorithm 1 from the main text) with $q = 0$ (i.e., no swaps) on a quantum network with (2,3)-tree topology. Nodes 0, 1, and 2 correspond to nodes in levels 0, 1, and 2 of the tree, respectively (i.e., they have physical node degrees $d_0 = 2, d_1 = 3,$ and $d_2 = 1,$ respectively). Each solid line corresponds to each of the three nodes. The dashed lines correspond to the expected steady-state values predicted by Eqs. (C7) and (C8) for each of the nodes. The standard error for each solid line is shown as a shaded region, although it is hard to notice since its maximum value is 0.017 in (a) and 0.056 in (b). Other parameters used in this experiment: $p_{\text{gen}} = 0.2, F_{\text{new}} = 0.9, r = 5, T = 2000$ time steps, $p_{\text{cons}} = 0.1, F_{\text{app}} = 0.6, t_{\text{cut}} = 221$ time steps [given by (3)]. Numerical results are obtained using a network simulation and Monte Carlo sampling with 10^4 samples.

The size of this interval of confidence is

$$\Delta_{ij} = 2\varepsilon - |\bar{X}_N(t_i) - \bar{X}_N(t_j)|. \tag{D7}$$

We provide a graphical intuition in Fig. 12.

Algorithm 2 finds the smallest α such that $\alpha \leq M - w$ and $\Delta_{ij} \geq \frac{3}{2}\varepsilon$, for any $i, j > \alpha$. Then, we say that the steady state starts at t_α . If α does not exist, the algorithm aborts. Next, we show that the condition stated in the theorem,

$$\Pr\{\mathbb{E}[X(t_i)] \in \text{IC}_{ij}\} \geq 0.815, \forall i, j > \alpha, \tag{D8}$$

is equivalent to $\Delta_{ij} \geq \frac{3}{2}\varepsilon$, for any $i, j > \alpha$. We proceed as follows:

$$\begin{aligned} &\Pr\{\mathbb{E}[X(t_i)] \in \text{IC}_{ij}\} \\ &\stackrel{a}{=} \Pr\{\mathbb{E}[X(t_i)] \in (\bar{X}_N(t_i) - \varepsilon, \bar{X}_N(t_i) + \Delta_{ij} - \varepsilon)\} \end{aligned}$$

$$\begin{aligned} &\stackrel{b}{=} \int_{\bar{X}_N(t_i) - \varepsilon}^{\bar{X}_N(t_i)} f_N(x_i) dx_i + \int_{\bar{X}_N(t_i)}^{\bar{X}_N(t_i) + \Delta_{ij} - \varepsilon} f_N(x_i) dx_i \\ &\stackrel{c}{\geq} \int_{\bar{X}_N(t_i) - \frac{2\sigma(t)}{\sqrt{N}}}^{\bar{X}_N(t_i)} f_N(x_i) dx_i + \int_{\bar{X}_N(t_i)}^{\bar{X}_N(t_i) + \Delta_{ij} - \varepsilon} f_N(x_i) dx_i \\ &\stackrel{d}{\geq} \frac{0.95}{2} + \int_{\bar{X}_N(t_i)}^{\bar{X}_N(t_i) + \Delta_{ij} - \varepsilon} f_N(x_i) dx_i \\ &\stackrel{e}{\geq} 0.475 + \int_{\bar{X}_N(t_i)}^{\bar{X}_N(t_i) + \frac{\varepsilon}{2}} f_N(x_i) dx_i \\ &\stackrel{f}{\geq} 0.475 + \int_{\bar{X}_N(t_i)}^{\bar{X}_N(t_i) + \frac{\sigma}{\sqrt{N}}} f_N(x_i) dx_i \\ &\stackrel{g}{\geq} 0.475 + \frac{0.68}{2} \\ &= 0.815 \end{aligned} \tag{D9}$$

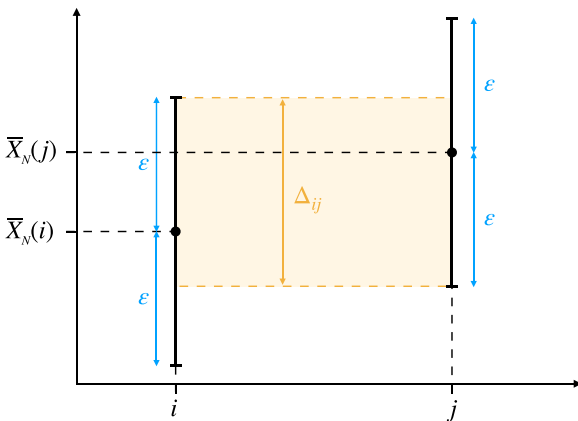


FIG. 12. Graphical intuition for the interval of confidence ij used to identify the steady state. $\bar{X}_N(i)$ corresponds to the sample mean at i , ε is the error, and Δ_{ij} is the size of the interval of confidence ij (highlighted in yellow).

with the following steps.

- (a) Without loss of generality, assume $\bar{X}_N(t_i) \geq \bar{X}_N(t_j)$.
- (b) Let $f_N(x_i)$ be the probability distribution function of $\mathbb{E}[X(t_i)]$. As previously shown, when N goes to infinity, this distribution converges to a normal distribution $\mathcal{N}(\bar{X}_N(t_i), \sigma(t)^2/N)$. We assume N is sufficiently large.
- (c) Using (D3), $\varepsilon = \frac{b-a}{\sqrt{N}} \geq \frac{2\sigma(t)}{\sqrt{N}}$.
- (d) The probability that a normally distributed random variable takes a value between the mean and two standard deviations away is larger than $\frac{0.95}{2}$, i.e., $\int_{\mu-2\sigma}^{\mu} f(z) dz = \int_{\mu}^{\mu+2\sigma} f(z) dz \geq \frac{0.95}{2}$, where $f(z)$ is the probability distribution of $Z \sim \mathcal{N}(\mu, \sigma^2)$.
- (e) The algorithm only considers i and j such that $\Delta_{ij} \geq \frac{3}{2}\varepsilon$.
- (f) Using (D3) again, $\varepsilon \geq \frac{2\sigma(t)}{\sqrt{N}}$.

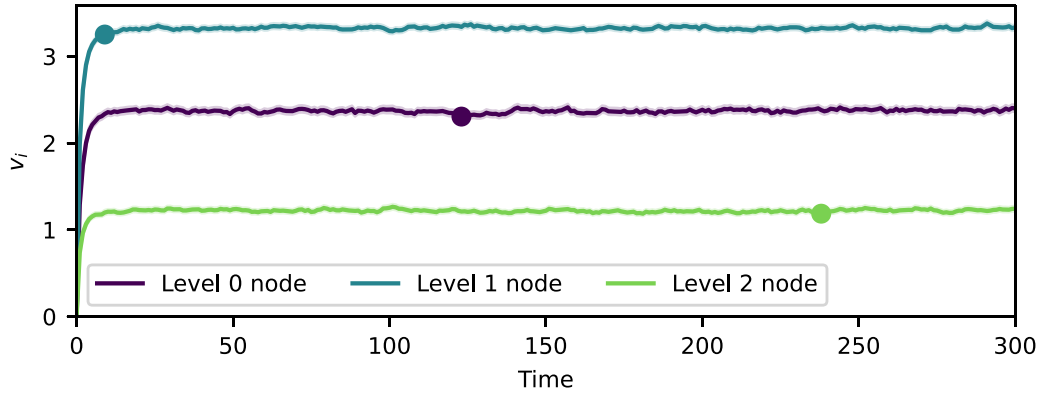


FIG. 13. Algorithm 2 can identify the steady state of a stochastic process. Evolution of the average virtual neighborhood size in a quantum network with a (2,3)-tree topology running the SRS protocol described in the main text. Each line (purple, blue, and green) corresponds to a node in a different level of the tree (level 0, 1, and 2). Dots indicate that the steady state has been reached, according to Algorithm 2. The error for each solid line is shown as a shaded region, although it is hard to notice since its maximum value is 0.029 (the error is defined as $2\hat{\sigma}/N_{\text{samples}}$, where $\hat{\sigma}$ is the sample standard deviation and N_{samples} is the number of samples). Other parameters used in this experiment: $p_{\text{gen}} = 0.9$, $F_{\text{new}} = 0.88$, $p_{\text{swap}} = 1$, $r = 5$, $T = 2000$ time steps, $M = 4$, $p_{\text{cons}} = 0.225$, $q = 0.1$, $F_{\text{app}} = 0.6$, $t_{\text{cut}} = 56$ time steps. Numerical results are obtained using a network simulation and Monte Carlo sampling with 10^3 samples. The simulation was run over 560 time steps (only the first 300 are shown here) and the steady-state window was 112 time steps.

(g) The probability that a normally distributed random variable takes a value between the mean and one standard deviation away is larger than $\frac{0.68}{2}$, i.e., $\int_{\mu-\sigma}^{\mu} f(z) dz = \int_{\mu}^{\mu+\sigma} f(z) dz \geq \frac{0.68}{2}$, where $f(z)$ is the probability distribution of $Z \sim \mathcal{N}(\mu, \sigma^2)$. ■

Note that the validity of this method depends on the number of samples N , which must be sufficiently large in order to apply the central limit theorem.

In our simulations, we employ Algorithm 2 to check the existence of the steady state in the virtual neighborhood size, $v_i(t)$, and the virtual node degree, $k_i(t)$, of every node i . After identifying the steady state, we take the average at the final simulation time as an estimate for the expected steady-state value, i.e., $\lim_{t \rightarrow \infty} \mathbb{E}[v_i(t)] \approx \bar{v}_{i,N}(t_{M-1})$ and $\lim_{t \rightarrow \infty} \mathbb{E}[k_i(t)] \approx \bar{k}_{i,N}(t_{M-1})$, where $\bar{v}_{i,N}(t)$ and $\bar{k}_{i,N}(t)$ are the sample averages at time t . The virtual neighborhood size of node i is upper bounded by $b = \min(rd_i, n)$, where rd_i is the total number of qubits at node i and n is the total number of nodes. The virtual degree of node i is upper bounded by $b = rd_i$. In this paper, each simulation was run over $10t_{\text{cut}}$ time steps, and the window used to estimate the steady state was $w = 2t_{\text{cut}}$.

When the standard error is very small and the mean value is slowly converging to the steady-state value, the overlaps between intervals of confidence (Δ_{ij}) may be too small. Then, our algorithm may abort, indicating that there is not steady state. In practice, we would like the algorithm to declare that the steady state has been reached once we are close enough to the steady-state value. To prevent the algorithm from aborting in such a situation, we can increase the value of b to increase the size of the interval of confidence (ε) in the algorithm.

We considered employing other data analysis techniques, such as bootstrapping and data blocking [77], to improve

our estimates. However, we decided to not use them since (i) bootstrapping would require running the simulations over many more time steps to be able to take many samples spaced an autocorrelation time and (ii) data blocking requires a much larger storage space.

As a final remark, we measure the error in the estimate of the expected steady-state values using the standard error $\epsilon = s_N/\sqrt{N}$, where s_N is the sample standard deviation. In particular, the error bars used in this paper correspond to $\pm 2\epsilon$, which provide a 95% interval of confidence.

Figure 13 shows an example of our algorithm finding the steady state of the virtual neighborhood size when running the SRS protocol in a network with a tree topology. The virtual neighborhood size of three nodes is shown in different colors. Dots correspond to the time t_α at which the algorithm declares that the steady state has been reached.

APPENDIX E: EXTRA EXPERIMENTS ON A TREE NETWORK

Here, we provide more examples of the dependence of the virtual neighborhood size, v_i , and the virtual node degree, k_i , on the SRS protocol parameter q (probability that a node performs a swap). In the main text, we discuss the dependence on q using a network with the following baseline set of parameters: (2,3)-tree topology, $p_{\text{gen}} = 0.9$, $F_{\text{new}} = 0.888$, $p_{\text{swap}} = 1$, $r = 5$, $T = 2000$ time steps, $M = 4$, $p_{\text{cons}} = 0.225$, $F_{\text{app}} = 0.6$, $t_{\text{cut}} = 56$ time steps. Figure 14 shows similar plots for networks with slightly different combinations of parameters that correspond to larger trees, smaller consumption rate, and probabilistic swapping. In all situations we observe the same qualitative behavior as in the baseline case: the value of q that maximizes the virtual neighborhood size is node dependent, and k_i is monotonically decreasing with increasing q .

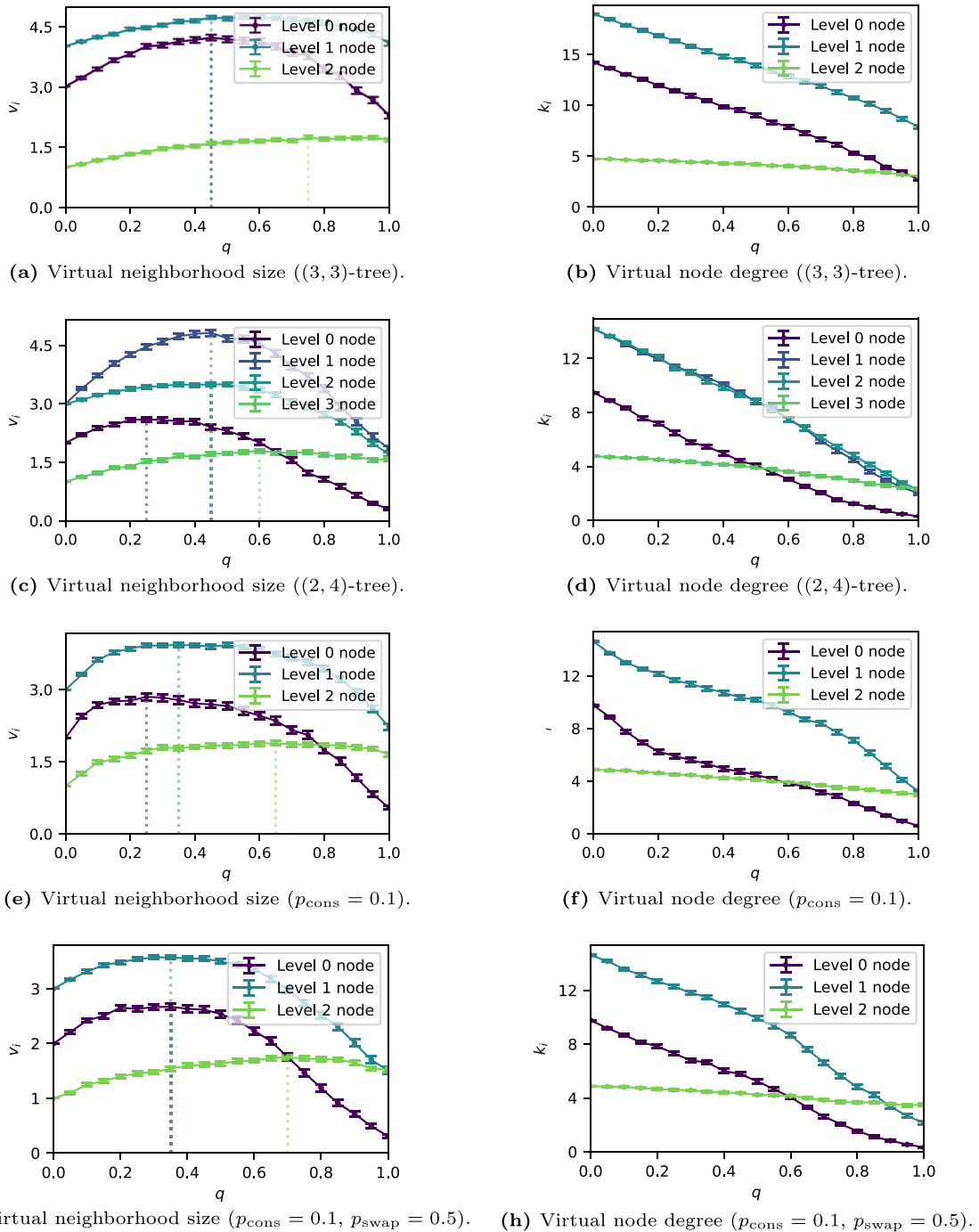


FIG. 14. Our performance metrics show the same qualitative behavior for different combinations of parameters. Expected virtual neighborhood size (a), (c), (e), (g) and virtual node degree (b), (d), (f), (h) in the steady state in a tree network running the SRS protocol vs the protocol parameter q . The value of q that maximizes the virtual neighborhood size is indicated by the dotted lines. Baseline parameters: (2,3)-tree topology, $p_{\text{gen}} = 0.9$, $F_{\text{new}} = 0.888$, $p_{\text{swap}} = 1$, $r = 5$, $T = 2000$ time steps, $M = 4$, $p_{\text{cons}} = 0.225$, $F_{\text{app}} = 0.6$, $t_{\text{cut}} = 56$ time steps. The subfigures in each row correspond to a different experiment (each caption indicates the parameters that have a different value in that experiment). Results are obtained using a network simulation and Monte Carlo sampling with 10^3 samples. The error in the error bars is defined as $2\hat{\sigma}/N_{\text{samples}}$, where $\hat{\sigma}$ is the sample standard deviation and N_{samples} is the number of samples.

[1] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, Teleporting an unknown quantum state

via dual classical and Einstein-Podolsky-Rosen channels, *Phys. Rev. Lett.* **70**, 1895 (1993).

- [2] D. Gottesman and I. L. Chuang, Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations, *Nature (London)* **402**, 390 (1999).
- [3] A. K. Ekert, Quantum cryptography based on Bell's theorem, *Phys. Rev. Lett.* **67**, 661 (1991).
- [4] C. H. Bennett, G. Brassard, and N. D. Mermin, Quantum cryptography without Bell's theorem, *Phys. Rev. Lett.* **68**, 557 (1992).
- [5] Y. Xia, W. Li, W. Clark, D. Hart, Q. Zhuang, and Z. Zhang, Demonstration of a reconfigurable entangled radio-frequency photonic sensor network, *Phys. Rev. Lett.* **124**, 150502 (2020).
- [6] M. R. Grace, C. N. Gagatsos, and S. Guha, Entanglement-enhanced estimation of a parameter embedded in multiple phases, *Phys. Rev. Res.* **3**, 033114 (2021).
- [7] K. Chakraborty, F. Rozpedek, A. Dahlberg, and S. Wehner, Distributed routing in a quantum internet, [arXiv:1907.11630](https://arxiv.org/abs/1907.11630).
- [8] J. Illiano, M. Caleffi, A. Manzalini, and A. S. Cacciapuoti, Quantum internet protocol stack: A comprehensive survey, *Comput. Net.* **213**, 109092 (2022).
- [9] H.-J. Briegel, W. Dür, J. I. Cirac, and P. Zoller, Quantum repeaters: The role of imperfect local operations in quantum communication, *Phys. Rev. Lett.* **81**, 5932 (1998).
- [10] M. Victora, S. Krastanov, A. S. de la Cerda, S. Willis, and P. Narang, Purification and entanglement routing on quantum networks, [arXiv:2011.11644](https://arxiv.org/abs/2011.11644).
- [11] G. Vardoyan, S. Guha, P. Nain, and D. Towsley, On the stochastic analysis of a quantum entanglement distribution switch, *IEEE Trans. Quantum Eng.* **2**, 1 (2021).
- [12] Á. G. Iñesta, G. Vardoyan, L. Scavuzzo, and S. Wehner, Optimal entanglement distribution policies in homogeneous repeater chains with cutoffs, *npj Quantum Inf.* **9**, 46 (2023).
- [13] R. Ursin, F. Tiefenbacher, T. Schmitt-Manderbach, H. Weier, T. Scheidl, M. Lindenthal, B. Blauensteiner, T. Jennewein, J. Perdigues, P. Trojek *et al.*, Entanglement-based quantum communication over 144 km, *Nat. Phys.* **3**, 481 (2007).
- [14] D. L. Moehring, P. Maunz, S. Olmschenk, K. C. Younge, D. N. Matsukevich, L.-M. Duan, and C. Monroe, Entanglement of single-atom quantum bits at a distance, *Nature (London)* **449**, 68 (2007).
- [15] L. Slodička, G. Hétet, N. Röck, P. Schindler, M. Hennrich, and R. Blatt, Atom-atom entanglement by single-photon detection, *Phys. Rev. Lett.* **110**, 083603 (2013).
- [16] H. Bernien, B. Hensen, W. Pfaff, G. Koolstra, M. S. Blok, L. Robledo, T. H. Taminiu, M. Markham, D. J. Twitchen, L. Childress *et al.*, Heralded entanglement between solid-state qubits separated by three metres, *Nature (London)* **497**, 86 (2013).
- [17] B. Hensen, H. Bernien, A. E. Dréau, A. Reiserer, Norbert Kalb, M. S. Blok, J. Ruitenber, R. F. L. Vermeulen, R. N. Schouten, C. Abellán *et al.*, Loophole-free bell inequality violation using electron spins separated by 1.3 kilometres, *Nature (London)* **526**, 682 (2015).
- [18] P. C. Humphreys, N. Kalb, J. P. J. Morits, R. N. Schouten, R. F. L. Vermeulen, D. J. Twitchen, M. Markham, and Ronald Hanson, Deterministic delivery of remote entanglement on a quantum network, *Nature (London)* **558**, 268 (2018).
- [19] M. Pompili, S. L. N. Hermans, S. Baier, H. K. C. Beukers, P. C. Humphreys, R. N. Schouten, R. F. L. Vermeulen, M. J. Tiggeleman, L. dos Santos Martins, B. Dirkse *et al.*, Realization of a multinode quantum network of remote solid-state qubits, *Science* **372**, 259 (2021).
- [20] M. Ben-Or, C. Crépeau, D. Gottesman, A. Hassidim, and A. Smith, Secure multiparty quantum computation with (only) a strict honest majority, in *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)* (IEEE, New York, 2006), pp. 249–260.
- [21] D. Leichtle, L. Music, E. Kashfi, and H. Ollivier, Verifying BQP computations on noisy devices with minimal overhead, *PRX Quantum* **2**, 040302 (2021).
- [22] C. Kruszynska, S. Anders, W. Dür, and H. J. Briegel, Quantum communication cost of preparing multipartite entanglement, *Phys. Rev. A* **73**, 062328 (2006).
- [23] A. Pirker, J. Wallnöfer, and W. Dür, Modular architectures for quantum networks, *New J. Phys.* **20**, 053054 (2018).
- [24] C. Meignant, D. Markham, and F. Grosshans, Distributing graph states over arbitrary quantum networks, *Phys. Rev. A* **100**, 052333 (2019).
- [25] L. Bugalho, B. C. Coutinho, F. A. Monteiro, and Y. Omar, Distributing multipartite entanglement over noisy quantum networks, *Quantum* **7**, 920 (2023).
- [26] K.-I. Yoshino, T. Ochi, M. Fujiwara, M. Sasaki, and A. Tajima, Maintenance-free operation of WDM quantum key distribution system through a field fiber over 30 days, *Opt. Express* **21**, 31395 (2013).
- [27] L. J. Stephenson, D. P. Nadlinger, B. C. Nichol, S. An, P. Drmota, T. G. Ballance, K. Thirumalai, J. F. Goodwin, D. M. Lucas, and C. J. Ballance, High-rate, high-fidelity entanglement of qubits across an elementary quantum network, *Phys. Rev. Lett.* **124**, 110501 (2020).
- [28] J. S. Sidhu, S. K. Joshi, M. Gündoğan, T. Brougham, D. Lowndes, L. Mazzarella, M. Krutzik, S. Mohapatra, D. Dequal, G. Vallone *et al.*, Advances in space quantum communications, *IET Quantum Comm.* **2**, 182 (2021).
- [29] S. D. Barrett and P. Kok, Efficient high-fidelity quantum computation using matter qubits and linear optics, *Phys. Rev. A* **71**, 060310(R) (2005).
- [30] M. Żukowski, A. Zeilinger, M. A. Horne, and A. K. Ekert, Event-ready-detectors Bell experiment via entanglement swapping, *Phys. Rev. Lett.* **71**, 4287 (1993).
- [31] L.-M. Duan, M. D. Lukin, J. Ignacio Cirac, and P. Zoller, Long-distance quantum communication with atomic ensembles and linear optics, *Nature (London)* **414**, 413 (2001).
- [32] N. Sangouard, C. Simon, H. De Riedmatten, and N. Gisin, Quantum repeaters based on atomic ensembles and linear optics, *Rev. Mod. Phys.* **83**, 33 (2011).
- [33] O. A. Collins, S. D. Jenkins, A. Kuzmich, and T. A. B. Kennedy, Multiplexed memory-insensitive quantum repeaters, *Phys. Rev. Lett.* **98**, 060502 (2007).
- [34] F. Rozpedek, K. Goodenough, J. Ribeiro, N. Kalb, V. C. Vivoli, A. Reiserer, R. Hanson, S. Wehner, and D. Elkouss, Parameter regimes for a single sequential quantum repeater, *Quantum Sci. Technol.* **3**, 034002 (2018).
- [35] S. Khatri, C. T. Matyas, A. U. Siddiqui, and J. P. Dowling, Practical figures of merit and thresholds for entanglement distribution in quantum networks, *Phys. Rev. Res.* **1**, 023032 (2019).
- [36] F. Rozpedek, R. Yehia, K. Goodenough, M. Ruf, P. C. Humphreys, R. Hanson, S. Wehner, and D. Elkouss, Near-term quantum-repeater experiments with nitrogen-vacancy centers:

- Overcoming the limitations of direct transmission, *Phys. Rev. A* **99**, 052330 (2019).
- [37] B. Li, T. Coopmans, and D. Elkouss, Efficient optimization of cut-offs in quantum repeater chains, in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, New York, 2020), pp. 158–168.
- [38] D. Gottesman, H.-K. Lo, N. Lutkenhaus, and J. Preskill, Security of quantum key distribution with imperfect devices, in *International Symposium on Information Theory, 2004. ISIT 2004. Proceedings* (IEEE, New York, 2004), p. 136.
- [39] S. Brito, A. Canabarro, R. Chaves, and D. Cavalcanti, Statistical properties of the quantum internet, *Phys. Rev. Lett.* **124**, 210501 (2020).
- [40] S. Brito, A. Canabarro, D. Cavalcanti, and R. Chaves, Satellite-based photonic quantum networks are small-world, *PRX Quantum* **2**, 010304 (2021).
- [41] A. Acín, J. I. Cirac, and M. Lewenstein, Entanglement percolation in quantum networks, *Nat. Phys.* **3**, 256 (2007).
- [42] M. Cuquet and J. Calsamiglia, Entanglement percolation in quantum complex networks, *Phys. Rev. Lett.* **103**, 240503 (2009).
- [43] S. Perseguers, M. Lewenstein, A. Acín, and J. I. Cirac, Quantum random networks, *Nat. Phys.* **6**, 539 (2010).
- [44] M. Cuquet and J. Calsamiglia, Limited-path-length entanglement percolation in quantum complex networks, *Phys. Rev. A* **83**, 032319 (2011).
- [45] L. Wu and S. Zhu, Entanglement percolation on a quantum internet with scale-free and clustering characters, *Phys. Rev. A* **84**, 052304 (2011).
- [46] H. Choi, M. Pant, S. Guha, and D. Englund, Percolation-based architecture for cluster state creation using photon-mediated entanglement between atomic memories, *npj Quantum Inf.* **5**, 104 (2019).
- [47] X. Meng, J. Gao, and S. Havlin, Concurrence percolation in quantum networks, *Phys. Rev. Lett.* **126**, 170501 (2021).
- [48] S. Pouryousef, N. K. Panigrahy, and D. Towsley, A quantum overlay network for efficient entanglement distribution, in *IEEE INFOCOM 2023: IEEE Conference on Computer Communications* (IEEE, New York, 2023), pp. 1–10.
- [49] A. Kolar, A. Zang, J. Chung, M. Suchara, and R. Kettimuthu, Adaptive, continuous entanglement generation for quantum networks, in *IEEE INFOCOM 2022: IEEE Conference on Computer Communications Workshops* (IEEE, New York, 2022), pp. 1–6.
- [50] S. Welte, B. Hacker, S. Daiss, S. Ritter, and G. Rempe, Photon-mediated quantum gate between two neutral atoms in an optical cavity, *Phys. Rev. X* **8**, 011018 (2018).
- [51] T. Coopmans, S. Brand, and D. Elkouss, Improved analytical bounds on delivery times of long-distance entanglement, *Phys. Rev. A* **105**, 012608 (2022).
- [52] G. Vardoyan, S. Guha, P. Nain, and D. Towsley, On the capacity region of bipartite and tripartite entanglement switching, *ACM SIGMETRICS Performance Evaluation Review* **48**, 45 (2021).
- [53] H. Choi, M. G. Davis, Á. G. Iñesta, and D. R. Englund, Scalable quantum networks: Congestion-free hierarchical entanglement routing with error correction, [arXiv:2306.09216](https://arxiv.org/abs/2306.09216).
- [54] R. F. Werner, Quantum states with Einstein-Podolsky-Rosen correlations admitting a hidden-variable model, *Phys. Rev. A* **40**, 4277 (1989).
- [55] W. Dür, M. Hein, J. I. Cirac, and H.-J. Briegel, Standard forms of noisy quantum operations via depolarization, *Phys. Rev. A* **72**, 052326 (2005).
- [56] S. Muralidharan, L. Li, J. Kim, N. Lütkenhaus, M. D. Lukin, and L. Jiang, Optimal architectures for long distance quantum communication, *Sci. Rep.* **6**, 20463 (2016).
- [57] S. Hermans, M. Pompili, L. dos Santos Martins, A. R.-P. Montblanch, H. Beukers, S. Baier, J. Borregaard, and R. Hanson, Entangling remote qubits using the single-photon protocol: An in-depth theoretical and experimental study, *New J. Phys.* **25**, 013011 (2023).
- [58] B. Davies, T. Beauchamp, G. Vardoyan, and S. Wehner, Tools for the analysis of quantum protocols requiring state generation within a time window, [arXiv:2304.12673](https://arxiv.org/abs/2304.12673).
- [59] M. Pompili, C. D. Donne, I. te Raa, B. van der Vecht, M. Skrzypczyk, G. Ferreira, L. de Kluijver, A. J. Stolk, S. L. N. Hermans, P. Pawelczak *et al.*, Experimental demonstration of entanglement delivery using a quantum network stack, *npj Quantum Inf.* **8**, 121 (2022).
- [60] W. Dür and H. J. Briegel, Entanglement purification and quantum error correction, *Rep. Prog. Phys.* **70**, 1381 (2007).
- [61] L. Hartmann, B. Kraus, H.-J. Briegel, and W. Dür, Role of memory errors in quantum repeaters, *Phys. Rev. A* **75**, 032310 (2007).
- [62] W. J. Munro, K. Azuma, K. Tamaki, and K. Nemoto, Inside quantum repeaters, *IEEE J. Sel. Top. Quantum Electron.* **21**, 78 (2015).
- [63] L. Chirolli and G. Burkard, Decoherence in solid-state qubits, *Adv. Phys.* **57**, 225 (2008).
- [64] S. C. Benjamin, D. E. Browne, J. Fitzsimons, and J. J. L. Morton, Brokered graph-state quantum computation, *New J. Phys.* **8**, 141 (2006).
- [65] 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.* **8**, 75 (2022).
- [66] M. Skrzypczyk and S. Wehner, An architecture for meeting quality-of-service requirements in multi-user quantum networks, [arXiv:2111.13124](https://arxiv.org/abs/2111.13124).
- [67] T. P. Harty, D. T. C. Allcock, C. J. Ballance, L. Guidoni, H. A. Janacek, N. M. Linke, D. N. Stacey, and D. M. Lucas, High-fidelity preparation, gates, memory, and readout of a trapped-ion quantum bit, *Phys. Rev. Lett.* **113**, 220501 (2014).
- [68] L. Talsma, Continuous distribution of entanglement in quantum networks with regular topologies M.Sc. thesis, Delft University of Technology, 2023, <https://repository.tudelft.nl/islandora/object/uuid:7a891675-6bb9-4353-b510-e1e01d945023>.
- [69] R. T. Marler and J. S. Arora, Survey of multi-objective optimization methods for engineering, *Struct. Multidiscip. Optim.* **26**, 369 (2004).
- [70] Á. G. Iñesta and S. Wehner, Data for “Performance metrics for the continuous distribution of entanglement in multi-user quantum networks,” *4TU ResearchData* (2023); GitHub repository, <https://github.com/AlvaroGI/optimizing-cd-protocols>.
- [71] J. Calsamiglia and N. Lütkenhaus, Maximum efficiency of a linear-optical bell-state analyzer, *Appl. Phys. B* **72**, 67 (2001).
- [72] F. Ewert and P. van Loock, 3/4-efficient Bell measurement with passive linear optics and unentangled ancillae, *Phys. Rev. Lett.* **113**, 140403 (2014).

- [73] M. Horodecki, P. Horodecki, and R. Horodecki, General teleportation channel, singlet fraction, and quasidistillation, *Phys. Rev. A* **60**, 1888 (1999).
- [74] Z. Cai and S. C. Benjamin, Constructing smaller pauli twirling sets for arbitrary error channels, *Sci. Rep.* **9**, 11281 (2019).
- [75] P. Van Mieghem, *Performance Analysis of Complex Networks and Systems* (Cambridge University, New York, 2014).
- [76] R. Bhatia and C. Davis, A better bound on the variance, *The American Mathematical Monthly*, **107**, 353 (2000).
- [77] J. Thijsen, *Computational Physics* (Cambridge University, New York, 2007).