Delft University of Technology

Efficient computation of the waiting time and fidelity in quantum repeater chains

Brand, Sebastiaan; Coopmans, Tim; Elkouss, David

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

Efficient computation of the waiting time and fidelity in quantum repeater chains

Sebastiaan Brand*†, Tim Coopmans*‡, David Elkouss‡

Abstract—Quantum communication enables a host of applications that cannot be achieved by classical communication means, with provably secure communication as one of the prime examples. The distance that quantum communication schemes can cover via direct communication is fundamentally limited by losses on the communication channel. By means of quantum repeaters, the reach of these schemes can be extended and chains of quantum repeaters could in principle cover arbitrarily long distances. In this work, we provide two efficient algorithms for determining the generation time and fidelity of the first generated entangled pair between the end nodes of a quantum repeater chain. The runtime of the algorithms increases polynomially with the number of segments of the chain, which improves upon the exponential runtime of existing algorithms. Our first algorithm is probabilistic and can analyze refined versions of repeater chain protocols which include intermediate entanglement distillation. Our second algorithm computes the waiting time distribution up to a pre-specified truncation time, has faster runtime than the first one and is moreover exact up to machine precision. Using our proof-of-principle implementation, we are able to analyze repeater chains of thousands of segments for some parameter regimes. The algorithms thus serve as useful tools for the analysis of large quantum repeater chain protocols and topologies of the future quantum internet.

Index Terms—Quantum communication, quantum repeater chains.

I. INTRODUCTION

THE quantum internet is the vision of a global network, running parallel to our current internet, that will enable the transmission of quantum information between arbitrary points on earth [1], [2]. The reason to investigate such a futuristic scenario is that quantum communication enables the implementation of applications beyond the reach of their classical counterparts [2]. Examples of these tasks range from secure key distribution and communications [3], [4], clock synchronization [5], distributed sensing [6], [7] and secure delegated quantum computing [8] to extending the baseline of telescopes [9]. One of the key elements to enable these applications is the distribution of entanglement between remote parties. However, the transmission of long-distance long-lived entanglement remains an open experimental challenge. The main problem to overcome are the losses in the physical transmission medium, typically glass fibre or free space. Although the impossibility of copying quantum information [10] renders signal amplification impossible, it is still possible

to reach long distances by means of a chain of intermediate nodes, known as quantum repeaters [11], between sender and receiver. Here, we aim at fully characterizing the behavior of an important class of entanglement distribution protocols over repeater chains as a tool for the analysis of quantum networks.

The key idea behind quantum repeater protocols is to divide the distance separating the two distant parties in a number of segments connected via a quantum repeater. At these points both losses and errors can be tackled. A large number of repeater protocols have been proposed [11]–[32] and to a large extent they can be classified [27], [33] depending on whether or not they use error correction codes to handle these issues. In the absence of coding, losses can be dealt with via heralded entanglement generation and errors via entanglement distillation [34]–[41]. Here, we will focus our interest in this type of protocols as their implementation is closer to experimental reach.

Existing analytical work is mostly aimed at estimating the mean waiting time or fidelity (see also [22], [42], [43] for other figures of merit). Some of this work builds on an approximation of the mean waiting time under the small-probability assumption [17], [23], [31], [44], while for a small number of segments or for some protocols it is possible to compute the waiting time probability distribution exactly [13], [30], [43], [45], [46]. However, depending on the application different statistics become relevant. For instance, in the presence of decoherence, one is also interested in the variations around the mean. In order to connect two segments via an intermediate repeater, both segments need to produce an entangled pair. When the first pair in one of the segments is ready, it has to wait until the second segment finalizes, and it decoheres while waiting. In this context, one may need to discard the entanglement after some maximum amount of time [43], [47], [48]. Entanglement is also used as a resource for implementing non-local gates in distributed quantum computers [49]. In this context, it is relevant to understand the time it takes to generate a pair of the desired quality with probability larger than some threshold, i.e. in the cumulative distribution. Here, we undertake the problem of fully characterizing the probability distribution of the waiting time and the associated fidelity to the maximally entangled state.

An algorithm to characterize the full waiting time distribution was first obtained in [45] using Markov chain theory. Its runtime scales with the number of vertices in the Markov chain, which grows exponentially with the number of repeater segments. In more recent work, Vinay and Kok show how to improve the runtime using results from complex analysis [46]. However, this method still remains exponential in the number

*These authors contributed equally.

† S. Brand is with Leiden Institute of Advanced Computer Science, Leiden University, Leiden, The Netherlands.

‡ T. Coopmans and D. Elkouss are with QuTech, Delft University of Technology, Delft, The Netherlands.

of repeater segments. Here, we provide two algorithms for computing the full distribution of the waiting time and fidelity following the same model as in [45]. Both algorithms are polynomial in the number of segments. Our main tool is the description of the waiting time and fidelity of the first produced end-to-end link as a recursively defined random variable, in line with the recursive structure of the repeater chain protocol. The first algorithm is a Monte Carlo algorithm which samples from this random variable, whereas our second algorithm is deterministic and computes the waiting time distribution up to a pre-specified truncation time. The power of the former algorithm lies in its extendibility: it can be used to analyze refined versions of repeater chain protocols which include intermediate entanglement distillation. The second algorithm is faster and exact: it computes the probability distribution of the waiting time and corresponding fidelity up to a pre-specified truncation point where the only source of error is machine precision. The speed of our algorithms allows us to analyze repeater chains with more than a thousand segments for some parameter regimes.

The organization of this work is as follows. In sec. II, we introduce notation and the family of repeater protocols under study. Then, in sec. III, we recursively define the waiting time and fidelity of the generation of a single entangled pair between the end nodes as a random variable. In sec. IV, we provide the two algorithms for computing the probability distribution of this random variable. We show in sec. V how to calculate tighter bounds on the mean waiting time than known in previous work. Numerical results are given in sec. VI. In Section VII we discuss the results obtained and provide an outlook for future research.

II. PRELIMINARIES

In this section, we elaborate on the repeater chain protocols we study in this work and explain how we model the quantum repeater hardware.

A. Quantum repeater chains

A quantum repeater chain connects two end points via a series of intermediate nodes. The goal of the two end points is to share an entangled state of two quantum bits or qubits, the unit of quantum information. In the entire paper, we refer to both the end points as well as the repeaters as 'nodes', so that a repeater chain of N segments has $N + 1$ nodes. We refer to two-qubit entanglement as a 'link' and to a link which spans n hops or segments in the repeater chain as an n-hop link. By end-to-end link, we mean a link between the end nodes of the repeater chain.

In the family of repeater chain protocols we study in this work, nodes are able to perform the following three actions: generate fresh entanglement with adjacent nodes, transform short-range entanglement into long-range entanglement by means of entanglement swapping, and increase the quality of links through entanglement distillation. In this section, we first explain each of these actions in more detail and subsequently elaborate on the family of repeater chain protocols.

First, a node can generate fresh entanglement with each of its adjacent nodes. We also refer to this single-hop entanglement as 'elementary links'.

Furthermore, if the middle node M of a two-segment chain with A and B as end nodes shares a link with node A and another link with node B, then M can perform an *entanglement swap* or merely 'swap', which results in a shared link between A and B. Performing an entanglement swap is identical to performing quantum teleportation on one part of an entangled pair using another entangled pair [50]. It consists of two parts: first, node M performs a measurement in the Bell basis on the two qubits it holds, which entangles the two qubits held by A and B. Next, node M sends a classical message to A and B in order to notify them of the outcome of this measurement, which determines the precise state of the entangled pair they hold. In this work, we model the entanglement swap as an operation which succeeds probabilistically: in the case of failure, both involved entangled pairs are lost.

The last action that a node can perform is entanglement distillation, which transforms two or more low-quality links between the same two nodes and produces a single high-quality link [36], [37]. In this work, we use entanglement distillation schemes which are probabilistic: the success or failure of a single entanglement distillation step depends on the joint outcomes on the two sides. These entanglement distillation schemes consist of two steps: first, both nodes perform local operations including a measurement on their parts of the two links, followed by transmitting the measurement outcome to each other. The combination of measurement outcomes from the two nodes determines whether the distillation step was successful. In the case of failure, both involved links are lost.

The repeater chain protocols we study in this work are all based on the seminal work of Briegel et al. [11], whose scheme was designed for a repeater chain of $N = W^n$ segments with $n \in \{1, 2, \dots\}$. For simplicity, we assume $W = 2$ in this work. We distinguish between two versions of the protocol. The first is SWAP-ONLY, where nodes generate elementary entanglement and transform it into end-to-end entanglement by means of entanglement swaps in a particular order explained below. The next is d-DIST-SWAP, which is identical to the SWAP-ONLY version except for the fact that every n-hop link is produced 2^d times for some integer $d \geq 1$ and then turned into a single high-quality link by performing entanglement distillation multiple times (more details below).

We start by explaining how the SWAP-ONLY protocol works for two segments and subsequently generalize to 2^n segments. On a chain of two segments, the SWAP-ONLY scheme starts with both end nodes generating a single entangled pair with the repeater node (fig. 1(a)). Once a link is generated, the two involved nodes store the state in memory. As soon as both pairs have been produced, the repeater node performs an entanglement swap on the two qubits it holds, which probabilistically produces a 2-hop link between the end nodes. In the case that the entanglement swap did not succeed, both end nodes will be notified of the failure by the heralding message from the repeater node and subsequently each restart generation of the single-hop entanglement.

In the generalization to repeater chains of 2^n segments

(fig. 1(b) and (c)), the SWAP-ONLY scheme starts with the two-segment repeater scheme as explained above on the first and second segment, on the third and fourth, and so on until segments $2^n - 1$ and 2^n. Approximately half of the intermediate nodes are thus involved in two instances of the scheme; as soon as both instances have finished generating 2-hop entanglement, the node will perform an entanglement swap to generate a single 4-hop entangled link. In case the entanglement swap fails, all nodes under the span of the 4 hops will start to generate single-hop entanglement again as part of the two-segment scheme. In general, to produce entanglement that spans 2^ℓ hops, the node that is located precisely in the middle of this span will wait for the production of two $2^{\ell-1}$-hop links and then perform an entanglement swap (see also fig. 1(c)). The failure of this swap requires to regenerate both $2^{\ell-1}$-hop links. We refer to $\ell \in \{0, 1, \ldots, n\}$ as the 'nesting level' of the protocol, such that single-hop entanglement is produced at the base level $\ell = 0$ and the entanglement swap at level $\ell \geq 1$ transforms two $2^{\ell-1}$-hop links into a single 2^ℓ-hop entangled pair.

In addition to the steps of the SWAP-ONLY version as described above, the original proposal of Briegel et al. included entanglement distillation in order to increase the quality of the input links to each entanglement swap. In this work, we specifically define a version of the repeater protocol, denoted by d-DIST-SWAP for some $d \geq 1$, where d rounds of distillation are performed at every nesting level. That is, instead of a single link, 2^d links are generated at every nesting level. These links are subsequently used as input to a recurrence distillation scheme: our description of this scheme follows the review work by Dür and Briegel [34]. In the first step of the recurrence protocol, the 2^d links are split up in pairs and used as input to entanglement distillation, which produces 2^{d-1} entangled pairs of higher quality. This process is repeated with the remaining links until only a single link is left, which is then used by the repeater node as input link to the entanglement swap. Rather than waiting for all 2^d links to have been generated before performing the first distillation step, the protocol performs the entanglement distillation as soon as two links are available. The failure of a distillation step requires the two involved nodes to regenerate the links. For $d = 0$, the d-DIST-SWAP scheme is identical to SWAP-ONLY since no distillation is performed.

Generating, distilling and swapping entanglement can in general all be probabilistic operations, which makes the total time it takes to distribute a single entangled pair between the end nodes of a repeater chain a random variable. We use the notation T_n to refer to the waiting time until a single end-to-end link in a 2^n-segment repeater chain has been produced. By F_n we refer to the link's fidelity, a measure of the quality of the state (see sec. II-B). Every time the quantities T_n and F_n are used in this work we explicitly state whether they correspond to the waiting time of the SWAP-ONLY version or the d-DIST-SWAP version for given d. The goal of this work is to find the joint probability distribution of T_n and F_n for both schemes.

B. Model

In the quantum repeater protocols we study (see sec. II-A), nodes can generate, store, distill and swap entangled links. We show here how we model each of these four operations.

For the generation of single-hop entanglement between two adjacent nodes, we choose generation schemes which perform heralded attempts of fixed duration L/c where c is the speed of light and L is the distance over which entanglement is generated [33]. In this work we study the topology where all nodes are equally spaced with distance $L = L_0$.

We model entanglement generation to succeed with a fixed probability $0 < p_{\text{gen}} \leq 1$. For simplicity, we also assume that the success probability p_{gen} is identical for all pairs of adjacent nodes. This implies independence between different entanglement generation attempts, i.e. the success or failure of a previous attempt has no influence on future attempts.

The first step of the entanglement swapping, the Bell-state measurement, is modelled as a probabilistic operation with fixed success probability $0 < p_{\text{swap}} \leq 1$ which is identical for all nodes. This success probability is independent of the state of the qubits that it acts upon. For simplicity, we assume that the duration of the Bell-state measurement is negligible. The Bell-state measurement is followed by a classical heralding signal to notify the nodes holding the other sides of the pair whether the Bell-state measurement was successful. An entanglement swap on two 2^n-hop links thus takes $2^n \cdot L_0/c$ time. Although our algorithms can account for this communication time (see sec. III-C), we will assume this time to be negligible in most of this work.

The fidelity $F(\rho, \sigma) \in [0, 1]$ between two quantum states on the same number of qubits, represented as density matrices ρ and σ, is a measure of their closeness, defined as

$$F(\rho, \sigma) := \text{Tr}\left(\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}\right)^2$$

which implies that $F(\rho, \sigma) = 1$ precisely if $\rho = \sigma$. By Bell-state fidelity, we mean the fidelity between σ and $\rho = |\Phi^+\rangle\langle\Phi^+|$ where $|\Phi^+\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$ is a Bell state.

We assume that the single-hop entangled states that are generated are two-qubit Werner states parameterized by a single parameter $0 \leq w_0 \leq 1$ [51]:

$$\rho(w_0) = w_0 |\Phi^+\rangle\langle\Phi^+| + (1 - w_0)\frac{\mathbb{1}_4}{4} \quad (1)$$

where $\mathbb{1}_4/4 = (|00\rangle\langle00| + |01\rangle\langle01| + |10\rangle\langle10| + |11\rangle\langle11|)/4$ is the maximally-mixed state on two qubits. A straightforward computation shows that the fidelity between $\rho(w)$ and the Bell state $|\Phi^+\rangle$ equals

$$F\left(\rho(w), |\Phi^+\rangle\langle\Phi^+|\right) = \langle\Phi^+|\rho(w)|\Phi^+\rangle = (1 + 3w)/4. \quad (2)$$

Quantum states that are stored in the memories decohere over time with the following noise: a Werner state $\rho(w)$ residing in memory for a time Δt will transform into the Werner state $\rho(w_{\text{decayed}})$ with

$$w_{\text{decayed}} = w \cdot e^{-\Delta t/T_{\text{coh}}} \quad (3)$$

where T_{coh} is the joint coherence time of the two quantum memories holding the qubits. We assume that access to a
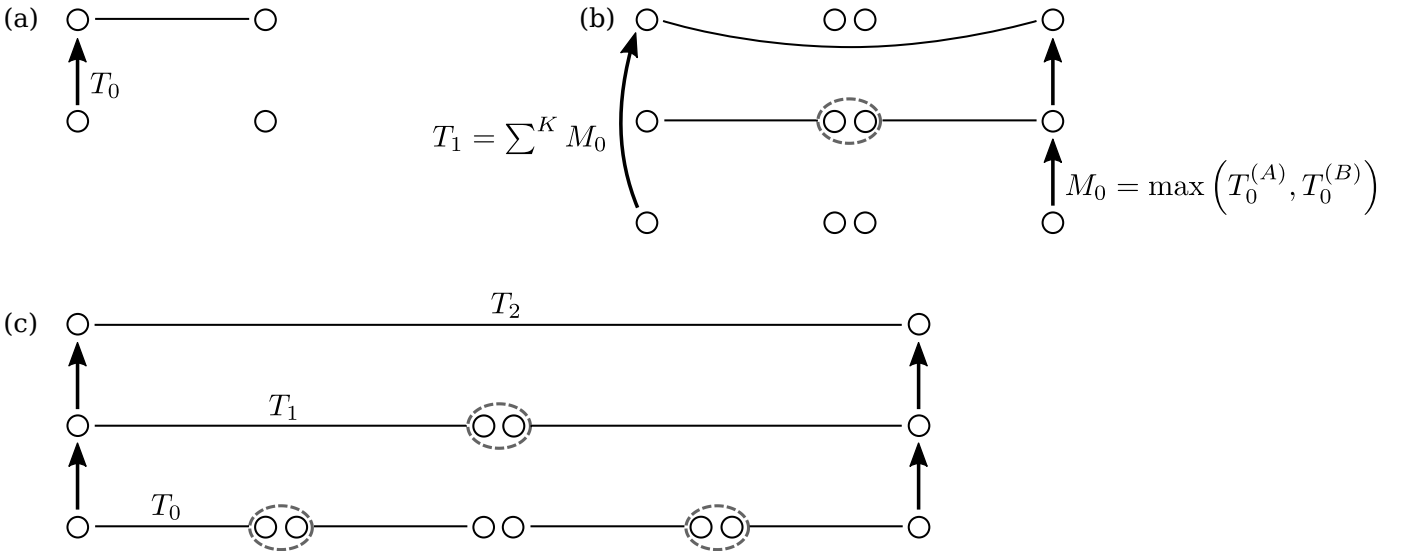
Fig. 1: The SWAP-ONLY version of the BDCZ protocol [11] and its completion time T_n as a random variable, where 2^n is the number of segments in the repeater chain (see also sec. II-A). (a) For two segments, T_0 represents the waiting time for the generation of a single link between two nodes without any intermediate repeater nodes. (b) Nested level structure of the protocol over $2^1 = 2$ segments. The production of entanglement over two segments first requires the generation of two links, each of which spans a single segment. The total time until both links have been generated equals M_0, the maximum of their individual generation times $T_0^{(A)}$ and $T_0^{(B)}$, which are independent random variables that are identically distributed (i.i.d.). Once the two links have been produced, a probabilistic entanglement swap is performed at both links. Failure of the entanglement swap requires the two single-hop links to be regenerated, each of which adds to the total waiting time T_1. The random variable K corresponds to the number of failing entanglement swaps up to and including the first successful swap. In this work we assume that K follows a geometric distribution with parameter p_{swap} (see sec. III-A). (c) A link that spans 2^n segments is produced in a nested fashion, where at each nesting level two links are produced and subsequently swapped.

quantum memory is on-demand, i.e. the quantum states can be stored and retrieved at any time and moreover there is no fidelity penalty associated with such memory access.

A successful entanglement swap acting on two Werner states $\rho(w)$ and $\rho(w')$ will produce the Werner state

$$\rho_{\text{swap}} = \rho(w \cdot w'). \qquad (4)$$

We assume that the Bell-state measurement and the local operations that the entanglement swap consists of are noiseless and instantaneous.

As base for entanglement distillation, we use the BBPSSW-scheme [36]. We modify it slightly by bringing the output state back into Werner form. The last step does not change the Bell-state fidelity of the output state. If two Werner states with parameters w_A and w_B are used as input to entanglement distillation, both the output Werner parameter w_{dist} and the success probability p_{dist} depend on the Werner parameters w_A and w_B of the states it acts upon (see appendix A):

$$w_{\text{dist}}(w_A, w_B) = \frac{1 + w_A + w_B + 5w_A w_B}{6p_{\text{dist}}} - \frac{1}{3} \qquad (5)$$

$$p_{\text{dist}}(w_A, w_B) = (1 + w_A w_B)/2. \qquad (6)$$

The two nodes involved in distillation on two 2^n-hop states send their individual measurement outcomes to each other, which takes $2^n \cdot (L_0/c)$ time but we will assume this time to be negligible for simplicity. We also assume that the duration of the local operations needed for the distillation is negligible.

C. Notation: random variables

In this section, we fix notation on random variables and operations on them.

Most random variables in this paper are discrete with (a subset of) the nonnegative integers as domain. Let X be such a random variable, then its probability distribution function $p_X : x \mapsto \Pr(X = x)$ describes the probability that its outcome will be $x \in \{0, 1, 2, \dots\}$. Equivalently, X is described by its cumulative distribution function $\Pr(X \leq x) = \sum_{y=0}^{x} \Pr(X = y)$, which is transformed to the probability distribution function as $\Pr(X = x) = \Pr(X \leq x) - \Pr(X \leq x - 1)$. Two random variables X and Y are independent if $\Pr(X = x \text{ and } Y = y) = \Pr(X = x) \cdot \Pr(Y = y)$ for all x and y in the domain. By a 'copy' of X, we mean a fresh random variable which is independent from X and identically distributed (i.i.d.). We will denote a copy by a superscript in parentheses. For example, $X^{(1)}$, $X^{(142)}$ and $X^{(A)}$ are all copies of X.

The mean of X is denoted by $E[X] = \sum_{x=0}^{\infty} \Pr(X = x) \cdot x$ and can equivalently be computed as $E[X] = \sum_{x=1}^{\infty} \Pr(X \geq x)$. If f is a function which takes two nonnegative integers as input, then the random variable $f(X, Y)$ has probability distribution function

$$\Pr(f(X, Y) = z) := \sum_{\substack{x=0, y=0: \\ f(x,y)=z}}^{\infty} \Pr(X = x \text{ and } Y = y).$$

An example of such a function is addition. Define $Z := X + Y$ where X and Y are independent, then the probability distribution p_Z of Z is given by the convolution of the distributions p_X and p_Y, denoted as $p_Z = p_X * p_Y$, which means [52]

$$p_Z(z) = \Pr(Z = z) = \sum_{x=0}^{z} p_X(x) \cdot p_Y(z - x).$$

The convolution operator $*$ is associative $((a * b) * c = a * (b * c))$ and thus writing $a * b * c$ is well-defined, for functions a, b, c from the nonnegative integers to the real numbers. In general, the probability distribution of sums of independent random variables equals the convolutions of their individual probability distribution functions.

III. RECURSIVE EXPRESSIONS FOR THE WAITING TIME AND FIDELITY AS A RANDOM VARIABLE

In this section, we derive expressions for the waiting time and fidelity of the first generated end-to-end link in the SWAP-ONLY repeater chain protocol. First, in sec. III-A, we derive a recursive definition for the random variable T_n, which represents the waiting time in a 2^n-segment repeater chain. Section III-B is devoted to extending this definition to the Werner parameter W_n of the pair, which stands in one-to-one correspondence to its fidelity F_n using eq. (2):

$$F_n = (1 + 3W_n)/4. \tag{7}$$

In sec. III-C, we show how to include the communication time after the entanglement swap and in sec. III-D, we extend the analysis of the waiting time and Werner parameter in the SWAP-ONLY protocol to the d-DIST-SWAP scheme.

A. Recursive expression for the waiting time in the SWAP-ONLY protocol

In the following, we will derive a recursive expression for the waiting time T_n of a SWAP-ONLY repeater chain of 2^n segments (see also fig. 1).

Before stating the expression, let us note that all three operations in the repeater chain protocols we study in this work, entanglement generation over a single hop, distillation and swapping, take a duration that is a multiple of L_0/c, the time to send information over a single segment (see sec. II-B for our assumptions on the duration of operations). For this reason, it is common to denote the waiting time in discrete units of L_0/c, which is a convention we comply with for T_n.

Let us first state the description of T_n before explaining it.

Waiting time in the SWAP-ONLY protocol

We recursively describe the random variable T_n that represents the waiting time until the first end-to-end link in a 2^n-segment SWAP-ONLY repeater chain is generated, for $n \in \{0, 1, \dots\}$. The waiting time T_0 for generating point-to-point entanglement follows a geometric distribution with parameter p_{gen}. At the recursive step, the waiting time is given as the geometric compound sum

$$T_{n+1} := \sum_{j=1}^{K_n} M_n^{(j)} \tag{8}$$

where M_n is an auxiliary random variable given by

$$M_n := g_{\text{T}}\left(T_n^{(A)}, T_n^{(B)}\right) \tag{9}$$

and the function g_{T} is defined as

$$g_{\text{T}}(t_A, t_B) := \max\{t_A, t_B\}. \tag{10}$$

The sum in eq. (8) is taken over the number of entanglement swaps K_n until the first success, which is geometrically distributed with parameter p_{swap} for every n. See fig. 1 for a depiction of T_n and K_n.

Let us now elaborate on each of the steps in the expression of T_n.

We start with the base case T_0, the waiting time for the generation of elementary entanglement. Since we model the generation of single-hop entanglement by attempts which succeed with a fixed probability p_{gen} (see sec. II-B), the waiting time T_0 is a discrete random variable (in units of L_0/c) which follows a geometric distribution with probability distribution given by $\Pr(T_0 = t) = p_{\text{gen}}(1 - p_{\text{gen}})^{t-1}$ for $t \in \{1, 2, 3, \dots\}$. For what follows, it will be more convenient to specify T_0 by its cumulative distribution function

$$\Pr(T_0 \le t) = 1 - (1 - p_{\text{gen}})^t. \tag{11}$$

Let us now assume that we have found an expression for T_n and we want to construct T_{n+1}. In order to perform the entanglement swap to produce a single 2^{n+1}-hop link, a node needs to wait for the production of two 2^n-hop links, one on each side. Denote the waiting time for one of the pairs by $T_n^{(A)}$ and the other by $T_n^{(B)}$, both of which are i.i.d. with T_n. The time until both pairs are available is now given by $M_n := \max(T_n^{(A)}, T_n^{(B)})$ which is distributed according to

$$
\begin{aligned}
\Pr(M_n \le t) &= \Pr\left(T_n^{(A)} \le t \text{ and } T_n^{(B)} \le t\right) \\
&= \Pr(T_n \le t)^2
\end{aligned} \tag{12}
$$

where the last equality follows from the fact that $T_n^{(A)}, T_n^{(B)}$ and T_n are pairwise i.i.d. Since we assume that both the duration of the Bell-state measurement and the communication time of the heralding signal after the entanglement swap are negligible (see sec. II-B), M_n is also the time at which the entanglement swap ends. We will drop the assumption on negligible communication time in sec. III-C.

In order to find the relation between M_n and T_{n+1}, first note that the number of swaps K_n at level n until the first successful swap follows a geometric distribution with parameter p_{swap}. This is a direct consequence of our choice to model the success probability p_{swap} to be independent of the state of the two input links (see sec. II-B). Next, recall that the two input links of a failing entanglement swap are lost and need to be regenerated. The regeneration of fresh entanglement after each failing entanglement swap adds to the waiting time. Thus, T_{n+1} is a *compound random variable*: it is the sum of K_n copies of M_n. Since the number of entanglement swaps K_n is geometrically distributed, we say that T_{n+1} is a *geometric compound sum* of K_n copies of M_n. To be precise, we write

$$T_{n+1} = \sum_{k=1}^{K_n} M_n^{(k)} \qquad (13)$$

which means that the probability distribution of the waiting time T_{n+1} is computed as the marginal of the waiting time conditioned on a fixed number of swaps:

$$\Pr(T_{n+1} = t) = \sum_{k=1}^{\infty} \Pr(K_n = k) \cdot \Pr\left[\left(\sum_{j=1}^{k} M_n^{(j)} \right) = t \right]$$

where the $M_n^{(j)}$ are copies of M_n.

The waiting time T_n is the same quantity as was studied by Shchukin et al. [45]. Indeed, in sec. VI, we show that our algorithms for computing the probability distribution of T_n recover their numerical results.

B. Joint recursive expression of waiting time and Werner parameter for the SWAP-ONLY protocol

In this section, we extend the expression of the waiting time for the first end-to-end link produced using the SWAP-ONLY protocol with the link's state. To be precise, we give a recursive expression for the waiting time T_n and Werner parameter W_n of this state, which is well-defined since all states that the SWAP-ONLY repeater chain protocol holds at any time during its execution are Werner states. The latter statement is a direct consequence of the fact that in our modeling, all operations in the SWAP-ONLY protocol only output Werner states: we choose to model the generated single-hop entanglement as Werner states and furthermore the class of Werner states is invariant under memory errors and entanglement swaps (see sec. II-B). The fidelity F_n of the first end-to-end state on 2^n segments can be computed from its Werner parameter using eq. (7).

We express the waiting time and Werner parameter as a joint random variable (T_n, W_n). Describing the two as a tuple allows us to capture the fact that the Werner parameter of a link depends on the time it was produced at. In sec. III-A, we found that the failure of multiple swapping attempts corresponds to the sum of their waiting times. In order to extend this description to the tuple of waiting time and Werner parameter, we define the *forgetting sum* $\widehat{\sum}$ on sequences of tuples $\{(x_j, y_j) | 1 \le j \le m\}$ for some $m \in \{1, 2, \dots\}$ as

$$\widehat{\sum}_{j=1}^{m} (x_j, y_j) := \left(\sum_{j=1}^{m} x_j, y_m \right). \qquad (14)$$

In analogy to the geometric compound sum from eq. (13), we define the *geometric compound forgetting sum* $(X', Y') := \widehat{\sum}_{j=1}^{K}(X, Y)$, which formally means

$$\Pr(X' = x \text{ and } Y' = y)$$
$$= \sum_{k=1}^{\infty} p(1-p)^{k-1} \cdot \Pr\left(\widehat{\sum}_{j=1}^{k}(X, Y)^{(j)} = (x, y) \right)$$

where X and Y and their primed version are random variables, and K is a geometrically distributed random variable with parameter p.

Making use of the compound forgetting sum, we give the expression for the joint random variable of waiting time T_n and Werner parameter W_n.

Waiting time and Werner parameter in the SWAP-ONLY protocol

The joint random variable (T_n, W_n) is defined as follows. The waiting time T_0 is the same as in sec. III-A and $\Pr(W_0 = w_0) = 1$ where $w_0 \in [0, 1]$ is some pre-specified constant that determines the state of the single-hop entanglement that is produced between adjacent nodes. At the recursive step, the waiting time and Werner parameter are given by the geometric compound forgetting sum

$$(T_{n+1}, W_{n+1}) := \widehat{\sum}_{k=1}^{K_n} (M_n, V_n)^{(k)} \qquad (15)$$

where, as in sec. III-A, K_n follows a geometric distribution with parameter p_{swap}. The auxiliary joint random variable (M_n, V_n) is defined as

$$(M_n, V_n) := g\left((T_n, W_n)^{(A)}, (T_n, W_n)^{(B)} \right). \qquad (16)$$

The function g is given by

$$g((t_A, w_A), (t_B, w_B)) := \qquad (17)$$
$$(g_{\text{T}}(t_A, t_B), g_{\text{W}}((t_A, w_A), (t_B, w_B)))$$

where g_{T} is defined in eq. 10 and

$$g_{\text{W}}((t_A, w_A), (t_B, w_B)) := w_A \cdot w_B \cdot e^{-|t_A - t_B|/T_{\text{coh}}} \qquad (18)$$

with T_{coh} the quantum memory coherence time as described in sec. II-B.

We now explain the above expressions. For a single segment $(n = 0)$, the waiting time and Werner parameter are uncorrelated because we model the attempts at generating single-hop entanglement to be independent and to each take equally long (see sec. II-B). At the recursive step, an entanglement swap which produces 2^{n+1}-hop entanglement requires the generation of two 2^n-hop links. The expression for the waiting time T_{n+1} is identical to eq. (8) in sec. III-A. In order to argue that eq. (15) also gives the correct expression for W_{n+1}, we first show that the Werner parameter of the output link of an entanglement swap is given by V_n in eq. (16), provided the swap succeeded. Since M_n as defined in eq. (16) is identical to its expression in eq. (9) in sec. III-A, we only need to argue

why g_W in eq. (18) correctly computes the Werner parameter of the output link after an entanglement swap.

In order to do so, denote by A and B the input links to the entanglement swap and denote by (t_A, w_A) and (t_B, w_B) their respective delivery times and Werner parameters. Without loss of generality, choose $t_A \geq t_B$, i.e. link A is produced after link B. Link A is produced last, so the entanglement swap will be performed directly after its generation and hence link A will enter the entanglement swap with Werner parameter w_A. Link B is produced earliest and will therefore decohere until production of link A. It follows from eq. (3) that B's Werner parameter immediately before the swap equals

$$w'_B = w_B \cdot e^{-|t_A - t_B|/T_{\text{coh}}}. \tag{19}$$

Once two links have been delivered, the entanglement swap would produce the 2^{n+1}-hop state with Werner parameter

$$w_A \cdot w'_B \tag{20}$$

as in eq. (4), provided the swap is successful. Combining eqs. (19) and (20) yields the definition of g_W in eq. (18).

Note that in the definition of g_W in eq. (18) we used the same assumption on the duration of the entanglement swap as in sec. III-A, i.e. that both the Bell-state measurement and the subsequent communication time are negligible (see also sec. II-B). This implies that V_n in eq. (16) expresses the Werner parameter of the produced 2^{n+1}-hop link in case the swap is successful. We treat the case of nonzero communication time in sec. III-C.

The last step in finding the Werner parameter W_{n+1} in eq. (15) is to bridge the gap with (M_n, V_n) from eq. (16). If the entanglement swap fails, then the 2^{n+1}-hop link with its Werner parameter in eq. (20) will never be produced since both initial 2^n-hop entangled pairs are lost. Instead, two fresh 2^n-hop links will be generated. In order to find how the Werner parameter on level $n + 1$ is expressed as a function of the waiting times and Werner parameters at level n, consider a sequence (m_j, v_j) of waiting times m_j and Werner parameters v_j, where j runs from 1 to the first successful swap k. The m_j correspond to the waiting time until the end of the entanglement swap that transforms two 2^n-hop links into a single 2^{n+1}-hop link and the v_j to the output link's Werner parameter if the swap were successful. We have found in sec. III-A that the total waiting time is given by $\sum_{j=1}^{k} m_j$, the sum of the duration of the production of the lost pairs (see eq. (8)). Note, however, that the Werner parameter of the 2^{n+1}-hop link is only influenced by the links that the *successful* entanglement swap acted upon. Since the entanglement swaps are performed until the first successful one, the output link is the last produced link and therefore its Werner parameter equals v_k. We thus find that the waiting time t_{final} of the first 2^{n+1}-hop link and its Werner parameter w_{final} are given by the forgetting sum from eq. (14):

$$(t_{\text{final}}, w_{\text{final}}) = \left(\sum_{j=1}^{k} m_j, v_k \right) = \widehat{\sum_{j=1}^{k}} (m_j, v_j).$$

Taking into account that the number of swaps k that need to be performed until the first successful one is an instance of the random variable K_n, we arrive at the full recursive expression for the waiting time and Werner parameter at level $n + 1$ as given in eq. (15).

It is not hard to see that the projection $(T_n, W_n) \mapsto T_n$ recovers the definition of waiting time from III-A. Indeed, following the recursive definition of (T_n, W_n) in eqs. (15)-(18), the waiting time T_n is not affected by the Werner parameters W_ℓ at lower nesting levels $\ell < n$.

C. Including communication time

While deriving the expressions for waiting time and Werner parameter of the first produced end-to-end link in secs. III-A and III-B, we have explicitly assumed that the total time the entanglement swap takes is negligible. Here, we include the communication time of the heralding signal from the entanglement swap into the expressions for M_n and V_n (eqs. (9) and (16)), which represent the waiting time and Werner parameter directly after the entanglement swap if it were successful. This communication time equals 2^n time steps (in units of L_0/c) for a swap that transforms two 2^n-hop links into a single 2^{n+1}-hop link (see sec. II-B). The expressions for M_n and V_n are modified by replacing g_T in eq. 10 by

$$g_T^n(t_A, t_B) := g_T(t_A, t_B) + 2^n. \tag{21}$$

and replacing g_W from eq. (18) by

$$g_W^n((t_A, w_A), (t_B, w_B)) := g_W((t_A, w_A), (t_B, w_B)) \cdot e^{-2^n/T_{\text{coh}}}. \tag{22}$$

Equation (21) expresses that the entanglement swap takes 2^n timesteps longer, while eq. (22) captures the decoherence of the state during the communication time of the entanglement swap, following eq. (3).

D. Waiting time and Werner parameter for the d-DIST-SWAP protocol

In this section, we sketch how to extend the expression of the waiting time T_n and Werner parameter W_n from secs. III-A-III-C to the case of the d-DIST-SWAP repeater protocol presented in sec. II-A. Recall that the d-DIST-SWAP protocol is identical to the SWAP-ONLY protocol except for the fact that each entanglement swap is performed on the output of a recurrence distillation scheme with d nesting levels. By a d'-distilled 2^n-hop link we denote a 2^n-hop link which is the result of successful entanglement distillation on two $(d' - 1)$-distilled 2^n-hop links and by a 0-distilled 2^n-hop link we mean a link that is the result of a successful entanglement swap on two 2^n-hop links. Thus, every entanglement swap in the d-DIST-SWAP protocol is performed on d-distilled links only.

Note that at every level of the nested swapping, there are d levels of nested distillation. To tackle the 'double nesting' we modify the waiting time in the SWAP-ONLY protocol by splitting up the tuple of random variables (T_n, W_n) in eq. (15), which represents the waiting time and Werner parameter at level n, into $d + 1$ tuples of random variables $(T_n^{d'}, W_n^{d'})$ for $d' \in \{0, 1, \ldots, d\}$. The random variable $T_n^{d'}$ corresponds to the waiting time until the end of the first successful distillation

attempt on two d'-distilled 2^n-hop links, and $W_n^{d'}$ to the link's Werner parameter.

We first analyze the recurrence distillation protocol at a single swapping nesting level and subsequently tie this analysis in with the nested swapping structure.

If we fix the nesting level n, we can straightforwardly apply the analysis of sec. III-B to the nested distillation. First, we define $(M_n^{d'}, V_n^{d'})$, which characterizes a link after a single distillation attempt on two 2^n-hop d'-distilled links in case the attempt is successful. This joint random variable is the analogue of (M_n, V_n) from eq. (16), which has the same interpretation but in this case for a swapping attempt. The analysis resulting in eq. (16) carries over and yields

$$(M_n^{d'}, V_n^{d'}) := g_{\mathrm{D}}\left((T_n^{d'}, W_n^{d'})^{(A)}, (T_n^{d'}, W_n^{d'})^{(B)}\right). \quad (23)$$

where g_{D} is the analogue of g in eq. (17) and describes how two input links are transformed into one high-quality link by a successful distillation step:

$$g_{\mathrm{D}}((t_A, w_A), (t_B, w_B)) = (g_{\mathrm{T}}(t_A, w_A), w)$$

where

$$w := \begin{cases} w_{\mathrm{dist}}\left(w_A \cdot e^{-|t_A - t_B|/T_{\mathrm{coh}}}, w_B\right) & \text{if } t_A \le t_B \\ w_{\mathrm{dist}}\left(w_A, w_B \cdot e^{-|t_A - t_B|/T_{\mathrm{coh}}}\right) & \text{if } t_A > t_B \end{cases}$$

and w_{dist} is given in eq. (5). The function g_{D} outputs a tuple of waiting time and Werner parameter of the output state after distillation. The waiting time requires two links to be generated and is thus given by g_{T} in eq. (10). The Werner parameter equals the Werner parameter of distillation as given by w_{dist} in eq. (5) on the two input links, of which the earlier suffered decoherence as given in eq. (3).

The random variables $(T_n^{d'}, W_n^{d'})$ correspond to the waiting time and Werner parameter after the first successful distillation attempt on two d'-distilled 2^n-hop links, so in line with the analysis leading to eq. (15) we obtain

$$(T_n^{d'+1}, W_n^{d'+1}) = \widehat{\sum}_{j=1}^{\mathcal{D}_n^{d'}} \left(M_n^{d'}, V_n^{d'}\right)^{(j)}. \quad (24)$$

The random variable $\mathcal{D}_n^{d'}$ corresponds to the number of distillation attempts with two d'-distilled 2^n-hop links as input, up to and including the first successful attempt. It is the analogue of K_n in eq. (15), the number of swap attempts until the first success.

At this point, we have an expression for (T_n^d, W_n^d), the waiting time and Werner parameter of the resulting link after performing a d-level recurrence protocol on 0-distilled input links that each span 2^n hops. Since the recurrence protocol is performed at every swapping nesting level of the d-DIST-SWAP protocol, we can insert this expression into our previous analysis using the following two remarks. First, a 0-distilled link is the output of an entanglement swap, so (T_n^0, W_n^0) in the d-DIST-SWAP scheme takes the role that (T_n, W_n) has in the SWAP-ONLY protocol:

$$(T_n^0, W_n^0) = (T_n, W_n). \quad (25)$$

Second, since an entanglement swap takes as input two d-distilled links, we find that we should replace the definition of (M_n, V_n) in eq. (16) by

$$(M_n, V_n) = g\left(\left(T_n^d, W_n^d\right)^{(A)}, \left(T_n^d, W_n^d\right)^{(B)}\right). \quad (26)$$

where g is defined in eq. (17).

We finish this section by remarking that for the d-DIST-SWAP protocol, we cannot treat waiting time independently of the Werner parameter of the produced link, as we did for the SWAP-ONLY scheme in sec. III-A. The reason behind this is the following difference between the nested swaps and the nested distillation: in the former, the success probability p_{swap} and therefore the number of swaps K_n is independent of the time and state of the produced links, whereas the success probability of entanglement distillation is a function of their states (see eq. (6)). Consequently, the summation bound $\mathcal{D}_n^{d'}$ and the Werner parameter $V_n^{d'}$ in the summands $\left(M_n^{d'}, V_n^{d'}\right)$ in eq. (24) are correlated. Therefore, both the waiting time and Werner parameter at any swapping level depend on both waiting time and Werner parameter at the levels below.

IV. ALGORITHMS FOR COMPUTING WAITING TIME AND FIDELITY OF THE FIRST END-TO-END LINK

In this section, we present two algorithms for determining the probability distribution of the waiting time T_n and average Werner parameter W_n of the first end-to-end link produced by the repeater chain (see sec. III). The first algorithm is a Monte Carlo algorithm which applies to both families of repeater chain protocols considered in this work: SWAP-ONLY and d-DIST-SWAP. The second algorithm only applies to the SWAP-ONLY protocol and is faster than the first. We summarize the runtime of the different algorithms presented in this section in table I.

A. First algorithm: Monte Carlo simulation

The first algorithm is a randomized function which produces a sample from the probability distribution of the joint random variable (T_n, W_n). By running the algorithm many times, sufficient statistics can be produced to reconstruct the distribution of the joint random variable up to arbitrary precision (see below for a rigorous statement). We first outline the algorithm that samples from the waiting time in the SWAP-ONLY protocol following sec. III-A, after which we show how to extend it to track the Werner parameter (sec. III-B), how to include the communication time after a swap (sec. III-C) and how to adjust it for the d-DIST-SWAP protocol (sec. III-D). Pseudocode can be found in algorithm 1.

We start by explaining the Monte Carlo algorithm for the waiting time in the SWAP-ONLY protocol. Let $s(X)$ denote a randomized function that yields a sample from the random variable X. We remark that if the cumulative distribution function of X is known, then sampling from X can be done efficiently using inverse transform sampling, which is a standard technique to produce a sample from an arbitrary distribution by evaluating its inverse cumulative distribution function on a sample from the uniform distribution on the interval $[0, 1]$.

Repeater chain protocol (sec. II-A)		Markov-chain-approach [45] [46]	Algorithms in this work	
			Monte-Carlo	Deterministic
SWAP-ONLY	Waiting time up to 99% of the cumulative probabilities	$\Theta\left(\exp\left(N\right)\right)$	$\mathcal{O}\left(\operatorname{poly}\left(N\right)\right)$	$\Theta\left(\operatorname{poly}\left(N\right)\right)$
	Waiting time up to fixed truncation time $t_{\mathrm{trunc}} = 1000$	$\Theta\left(\exp\left(N\right)\right)$	$\mathcal{O}\left(\operatorname{poly}\left(N\right)\right)$	$\Theta\left(\log\left(N\right)\right)$
	Fidelity	$\times$	$\mathcal{O}\left(\operatorname{poly}\left(N\right)\right)$	$\Theta\left(\operatorname{poly}\left(N\right)\right)$
d-DIST-SWAP	Waiting time & fidelity	$\times$	$\mathcal{O}\left(\operatorname{poly}\left(N\right)\right)$	$\times$

TABLE I: The time complexity of the algorithms for computing waiting time and fidelity of entanglement distribution through repeater chains as presented in this work compared to existing algorithms. The algorithms have exponential (exp) or polynomial (poly) runtime in $N = 2^n$, the number of segments in the repeater chain, for $n \in \{1, 2, \dots\}$. The Monte Carlo algorithm is a randomized algorithm; its presented runtime is the average runtime. The cross ($\times$) indicates that the algorithm is not present.

We can thus construct the sampler from the waiting time for elementary entanglement, T_0, using the inverse of the cumulative distribution function of T_0 as given in eq. (11):

$$s(T_0) = \lceil \log_{(1-p)}(1 - s(U)) \rceil \qquad (27)$$

where U is a random variable which is distributed uniformly at random on $[0, 1]$ and $\lceil . \rceil$ denotes the ceiling function.

For sampling from higher levels, we first note that we can easily transform a sampler $s(X)$ into a sampler $s_{\mathrm{sum}}(X, p)$ from a geometric sum $\sum_{j=1}^{K} X^{(j)}$, where K is geometrically distributed with parameter p. The sampler from the geometric sum probabilistically calls itself:

$$s_{\mathrm{sum}}(X, p) := \begin{cases} s(X) & \text{with prob. } p, \\ s(X) + s_{\mathrm{sum}}(X, p) & \text{with prob. } 1 - p. \end{cases}$$

From the recursive expression for the waiting time T_n in sec. III-A it now follows directly that we can construct a sampler from T_n for $n \geq 1$:

$$s(T_n) \quad = \quad s_{\mathrm{sum}}(M_n, p_{\mathrm{swap}})$$

which, per definition of s_{sum}, makes a call to $s(M_n)$ which is given by

$$s(M_n) \quad = \quad g_{\mathrm{T}}(s(T_{n-1}), s(T_{n-1}))$$

where g_{T} is defined in eq. (10).

Using the Dvoretzky-Kiefer-Wolfowitz inequality [53], we determine how many samples from (T_n, W_n) we need in order to obtain bounds on its cumulative probabilities. It follows from this inequality that if $q(t) := \Pr(T_n \leq t)$ denotes the cumulative probability function of the waiting time T_n and $q_m(t)$ the empirical cumulative probabilities after having drawn m samples, then the difference between q and q_m is bounded as

$$\Pr(|q(t) - q_m(t)| > \epsilon) \leq 2e^{-2m\epsilon^2}$$

for all $t \geq 0$. Thus we can bound the probability that the empirical estimate $q_m(t)$ deviates from $q(t)$ at most ϵ for any value of t by $z = 2e^{-2m\epsilon^2}$ if the number of samples to draw equals

$$m = -\log(z/2)/(2\epsilon^2) \qquad (28)$$

Let us emphasize that this number of samples is independent of any parameters of the repeater chain, for instance the number

of segments, and thus its contribution to the runtime or space usage of the Monte Carlo algorithm is at most a multiplicative constant, independent of any such parameters.

Following sec. III-B, we modify the Monte Carlo algorithm to also compute the Werner parameter of the sampled produced entangled pair (for pseudocode see algorithm 1). First note that the notation $s(X)$ which samples from a random variable X can also be applied to a joint random variable (X, Y), so that $s((X, Y))$ returns a tuple. We will now define a sampler $s((T_n, W_n))$ where (T_n, W_n) is the joint random variable representing waiting time and Werner parameter of a 2^n-segment SWAP-ONLY repeater chain (see sec. III-B). For this, we first need to adapt the sampler of the geometric compound sum s_{sum} to a sampler of the geometric compound forgetting sum (eq. (14)) by defining $\hat{s}_{\mathrm{sum}}((X, Y), p)$ where X and Y are arbitrary random variables and $p \in [0, 1]$ is the parameter of the geometric distribution:

$$\hat{s}_{\mathrm{sum}}((X, Y), p) := \begin{cases} s((X, Y)) & \text{with prob. } p, \\ \pi(s((X, Y))) + \hat{s}_{\mathrm{sum}}((X, Y), p) \\ \qquad\qquad \text{with prob. } 1 - p. \end{cases}$$

where '+' denotes pairwise addition and π is the projector onto the first element of a tuple: $\pi((x, y)) = (x, 0)$ for any numbers x, y.

A recursive definition of the joint sampling function from (T_n, W_n) follows directly from the joint expression for waiting time T_n and Werner parameter W_n in eqs. (15)-(18):

$$\begin{aligned} s((T_0, W_0)) &= (s(T_0), w_0) \\ s((T_n, W_n)) &= \hat{s}_{\mathrm{sum}}((M_n, V_n), p_{\mathrm{swap}}) \\ s((M_n, V_n)) &= g(s(T_{n-1}, W_{n-1}), s(T_{n-1}, W_{n-1})) \quad (29) \end{aligned}$$

where w_0 is the Werner parameter of each single-hop link at the time it is produced (see sec. II-B) and the function g is defined in eq. (17). In this pseudocode for this Monte Carlo algorithm in algorithm 1, the sampler $s(T_n, W_n)$ is denoted by sample_swap.

Since the expression for (T_n, W_n) from sec. III-B assumes that the communication time for the heralding signal after the entanglement swap takes negligible time, it is not included in the Monte Carlo algorithm above. Fortunately, the adaptation to include this communication time as in sec. III-C directly

carries over to the Monte Carlo algorithm by replacing g in eq. (29) with

$$g^n((t_A, w_A), (t_B, w_B)) := (g_{\mathrm{T}}^n(t_A, t_B), g_{\mathrm{W}}^n((t_A, w_A), (t_B, w_B))$$

where g_{T}^n and g_{W}^n are defined in eqs. (21) and (22).

The time complexity of the Monte Carlo algorithm is a random variable since it is a randomized algorithm. Every call to $s(T_{n+1}, W_{n+1})$ performs the auxiliary function $\hat{s}_{\mathrm{sum}}$ on average $1/p_{\mathrm{swap}}$ times, each of which calls $s(M_n, W_n)$ precisely once and thus $s(T_n, W_n)$ exactly twice by eq. (29). Given access to a constant-time sampler from the uniform distribution on $[0, 1]$, a sample from the base level $s(T_0, W_0)$ can be obtained in constant time, so a simple inductive argument shows that a drawing a single sample from (T_n, W_n) has average runtime $\mathcal{O}\left((2/p_{\mathrm{swap}})^n\right)$, which equals

$$\mathcal{O}\left(N^{\log_2(2/p_{\mathrm{swap}})}\right)$$

which is polynomial in the number of segments $N = 2^n$.

Following sec. III-D, we also adjust the Monte Carlo algorithm to determine the waiting time and average Werner parameter in the d-DIST-SWAP repeater chain protocol. We add a recursive function sample_dist in algorithm 1 for sampling from the random variable $T_n^{d'}$ from eq. (24), which represents the waiting time at each level $d' \in \{0, 1, \ldots, d\}$ of the nested distillation scheme. The relation between the random variable tuples $\left(T_n^{d'}, W_n^{d'}\right)$ and $\left(M_n^{d'}, V_n^{d'}\right)$ on the one hand and (T_n, W_n) and (M_n, V_n) on the other is mirrored in their implementations sample_dist and sample_swap, respectively: the function sample_swap calls sample_dist following eq. (26), which subsequently calls itself recursively for d nesting levels following eq. (23) and eq. (24) and calls sample_swap at the lowest level in line with eq. (25). See algorithm 1 for the full pseudocode of the d-dependent sampler sample_swap for the d-DIST-SWAP protocol.

The average runtime of the sampler for the d-DIST-SWAP protocol is upper bounded by $\mathcal{O}\left(4^d \cdot (2/p_{\mathrm{swap}})^n\right)$. In order to derive this, note that the probability that a distillation attempt succeeds (see eq. (6)) is lower bounded by $1/2$ and hence a call to sample_dist(n, d) recursively performs at most $(2/(1/2))^d = 4^d$ calls to sample_swap$(n - 1)$ on average. The average runtime of the full algorithm is the product of this number of calls and the average runtime of the SWAP-ONLY algorithm $\mathcal{O}\left((2/p_{\mathrm{swap}})^n\right)$ since the recurrence distillation scheme is performed at every swapping level.

Let us finish this section with an analysis of the algorithm's space complexity. For generating a single sample of (T_n, W_n) of the SWAP-ONLY protocol, the number of variables that need to be stored grows linearly in the number of segments n. To see this, first note that at level ℓ the algorithm only needs to keep track of two samples of $(T_{\ell-1}, W_{\ell-1})$ at a time, since in the case of a failed swap it may discard the samples after updating the total time used and subsequently reuse the space for storing two fresh samples. In addition, for producing these two samples, only two samples need to be stored at *every* level $< \ell$. The insight here is that at each level the required two

samples can be drawn *in sequence* rather than in parallel[1], so that the space needed to draw the first sample can be reused for the second. Therefore, the algorithm needs to keep track of at most two samples at every level, which implies that the total number of variables it stores is linear in the number of levels and thus in the number of segments n. For the d-DIST-SWAP protocol, the scaling is linear in $n \cdot d$ with d the number of distillation steps per nesting level, which can be shown by an analogous argument.

The number of samples that is required to generate a probability distribution histogram with pre-specified precision is independent of the number of segments (see explanation directly below eq. (28)). For constructing the histogram, we only need to store the waiting times for which at least a single sample was drawn and hence the number of such waiting times is also independent of the number of segments. We conclude that reproducing the probability distribution of (T_n, W_n) using the Monte Carlo algorithm will not exceed polynomial space usage in the number of segments n.

B. Second algorithm: deterministic computation

In this section, we present our full second algorithm, which computes the probability distribution of the waiting time and average Werner parameter up to some pre-specified truncation time t_{trunc}. The algorithm applies to the SWAP-ONLY repeater protocol. In what follows, we first show how to compute the probability distribution of the waiting time T_n of the SWAP-ONLY protocol by recursion (see sec. III-A). After this, we outline how our algorithm performs a modified version of this computation on the finite domain $\{1, 2, \ldots, t_{\mathrm{trunc}}\}$. We finish the section by extending its computation to include the average Werner parameter (sec. III-B).

Let us start by showing how to derive the probability distribution of the waiting time T_n in the SWAP-ONLY protocol. For a single repeater segment ($n = 0$), the waiting time follows the geometric distribution as given in eq. (11). For nesting levels $\ell \in \{0, 1, 2, \ldots, n\}$, the relation between the probability distributions of M_ℓ and T_ℓ follows straightforwardly from eq. (12):

$$\Pr(M_\ell = t) = \Pr(T_\ell \le t)^2 - \Pr(T_\ell \le t - 1)^2. \quad (30)$$

Now we compute the probability distribution of $T_{\ell+1} | K_\ell$, which is the waiting time conditioned on the number of swaps needed that transform 2^ℓ-hop entanglement to the final $2^{\ell+1}$-hop entanglement:

$$\Pr\left(T_{\ell+1} = t | K_\ell = k\right) = \Pr\left(\sum_{j=1}^{k} M_\ell^{(j)} = t\right)$$
$$= \left[\mathop{\scalebox{1.5}{$\ast$}}_{j=1}^{k} m_\ell\right](t) \quad (31)$$

where we have denoted $m_\ell(t) := \Pr(M_\ell = t)$ and $*$ denotes convolution of functions (see sec. II-C). The marginal proba-

---

[1]Note that in our runtime analysis, we already assumed sequentiality since we showed that the average *number of calls* to $s(T_0, W_0)$ is at most polynomial in the number of segments n.

Algorithm 1: Monte-Carlo algorithm sample_swap(n) for producing a single sample of the joint waiting time and Werner parameter (T_n, W_n) for a d-DIST-SWAP quantum repeater chain of 2^n segments as in sec. III-B. Setting $d = 0$ corresponds to the SWAP-ONLY repeater chain protocol.

Input : Success probabilities p_{gen} and p_{swap}, Werner parameter of single-hop links w_0, nesting level n, number of distillation rounds at each level d.

Output: Single sample from (T_n, W_n).

1 **if** $n = 0$ **then**
2     $u \leftarrow$ uniform random sample from $[0, 1]$
3   **return** $(\lceil \log_{(1-p)}(1 - u) \rceil, w_0)$ // (eq. (27))
4 **else if** $n \geq 1$ **then**
5     $(t_A, w_A) \leftarrow$ sample_dist(n, d)
6     $(t_B, w_B) \leftarrow$ sample_dist(n, d)
7     $t, w \leftarrow g((t_A, w_A), (t_B, w_B))$ // (eq. (17))
8     $u \leftarrow$ uniform random sample from $[0, 1]$
9   **if** $u \leq p_{\text{swap}}$ **then**
10   **return** t, w
11   **else**
12       $t_{\text{retry}}, w_{\text{retry}} \leftarrow$ sample_swap(n)
13   **return** $t + t_{\text{retry}}, w_{\text{retry}}$
14   **end**
15 **end**

16 **Auxiliary function** sample_dist(n, d) :
17 **if** $d = 0$ **then**
18   **return** sample_swap$(n - 1)$
19 **else**
20     $(t_A, w_A) \leftarrow$ sample_dist$(n, d - 1)$
21     $(t_B, w_B) \leftarrow$ sample_dist$(n, d - 1)$
22     $t, w \leftarrow g_{\text{D}}((t_A, w_A), (t_B, w_B))$ // eq. (24)
23     $u \leftarrow$ uniform random sample from $[0, 1]$
     // Success probability: eq. (6)
24   **if** $u \leq p_{\text{dist}}(w_A, w_B)$ **then**
25   **return** t, w
26   **else**
27       $t_{\text{retry}}, w_{\text{retry}} \leftarrow$ sample_dist(n, d)
28   **return** $t + t_{\text{retry}}, w_{\text{retry}}$
29   **end**
30 **end**

bility distribution of $T_{\ell+1}$ is calculated from the distribution of the conditional random variable $T_{\ell+1}|K_\ell$ as

$$\Pr(T_{\ell+1} = t) = \sum_{k=1}^{\infty} p_{\text{swap}}(1 - p_{\text{swap}})^{k-1} \Pr(T_{\ell+1} = t | K_\ell = k)$$
(32)

where we used the fact that the number of swaps K_ℓ is geometrically distributed with parameter p_{swap}.

Our algorithm computes the probability distribution of T_n by iterating the procedure in the eqs. (30), (31) and (32) over ℓ from 0 to $n - 1$ and is outlined in algorithm 2. Its implementation follows naturally from the equations above

Algorithm 2: Deterministic algorithm for computing the probability distribution of the waiting time T_n of the SWAP-ONLY protocol at nesting level n. The subroutine fast_convolution_algorithm computes the distribution of the sum of two random variables A and B, each represented by an array of size $t_{\text{trunc}} + 1$ with their probabilities $\Pr(A = t)$ and $\Pr(B = t)$ for $t \in \{0, 1, 2, \ldots, t_{\text{trunc}}\}$.

Input : Success probs. p_{gen} and p_{swap}, nesting level n
Output: Two-dimensional array of size $(n + 1) \times (t_{\text{trunc}} + 1)$ with entries $\Pr(T_\ell = t)$ for $\ell \in \{0, 1, 2, \ldots, n\}$ and $t \in \{0, 1, 2, \ldots, t_{\text{trunc}}\}$.

1 $C \leftarrow$ 3-dim. array of zeros,
2   size $(n + 1) \times (t_{\text{trunc}} + 1) \times (t_{\text{trunc}} + 1)$
3 $T \leftarrow$ 2-dim. array of zeros, size $(n + 1) \times (t_{\text{trunc}} + 1)$
4 $M \leftarrow$ 1-dim. array of zeros, of size $(t_{\text{trunc}} + 1)$
    // Base level probs (eq. (11))
5 **for** $t \in \{0, 1, \ldots, t_{\text{trunc}}\}$ **do**
6     $T[0, t] \leftarrow 1 - (1 - p_{\text{gen}})^t$
7 **end**
    // Probabilities on higher levels
8 **for** $\ell \in \{0, 1, \ldots, n - 1\}$ **do**
     // Maximum of two copies (eq. (30))
9   **for** $t \in \{1, 2, \ldots, t_{\text{trunc}}\}$ **do**
10       $M[t] \leftarrow T[\ell, t]^2 - T[\ell, t - 1]^2$
11   **end**
     // Conditional probs... (eq.(31))
12   **for** $k \in \{1, 2, \ldots, t_{\text{trunc}}\}$ **do**
13   set column $C[\ell, k]$ to output of
       convolve$(C[\ell, k - 1], k, M)$
14   **end**
     // ...and the marginals (eq. (32))
15   **for** $t \in \{1, 2, \ldots, t_{\text{trunc}}\}$ **do**
16   **for** $k \in \{1, 2, \ldots, t_{\text{trunc}}\}$ **do**
17   term $\leftarrow p_{\text{swap}}(1 - p_{\text{swap}})^{k-1} \cdot C[\ell, k, t]$
18   add term to $T[\ell + 1, t]$
19   **end**
20   **end**
     // Convert T to cumulative probs
21   **for** $t \in \{1, 2, \ldots, t_{\text{trunc}}\}$ **do**
22       $T[\ell + 1, t] \leftarrow T[\ell + 1, t] + T[\ell + 1, t - 1]$
23   **end**
24 **end**
25 **return** T

26 **Auxiliary function** convolve(S, k, M) :
27 **if** $k = 1$ **then**
28   **return** M
29 **else**
     // Compute convolution of two arrays using Fast Fourier Transforms
30   array_with_sum_distribution $\leftarrow$
     fast_convolution_algorithm(S, M)
31   **return** array_with_sum_distribution
32 **end**

Algorithm 3: Extension to algorithm 2 for computing W_n, the average Werner parameter of the end-to-end state produced at time $t \in \{0, 1, \ldots, t_{\text{trunc}}\}$ by a 2^n-segment repeater chain with the SWAP-ONLY protocol. The algorithm is an extension to algorithm 2 and contains of several parts that should be inserted into that algorithm.

The following should be inserted directly after line 4
in algorithm 2:

$W \leftarrow$
2-dim. array of zeros, size $(n + 1) \times (t_{\text{trunc}} + 1)$
for $t \in \{0, 1, 2, \ldots, t_{\text{trunc}}\}$ **do**
$\quad$ $W[0, t] \leftarrow w_0$
end

The following should be inserted directly after line 23
within the loop over ℓ in algorithm 2:
for $k \in \{1, 2, \ldots, t_{\text{trunc}}\}$ **do**
$\quad$ **for** $t_A \in \{1, 2, \ldots, t_{\text{trunc}}\}$ **do**
$\quad\quad$ **for** $t_B \in \{1, 2, \ldots, t_{\text{trunc}}\}$ **do**
$\quad\quad\quad$ // The average Werner
$\quad\quad\quad\quad$ parameter (eq. (35))
$\quad\quad\quad$ $w_A \leftarrow W[\ell, t_A]$
$\quad\quad\quad$ $w_B \leftarrow W[\ell, t_B]$
$\quad\quad\quad$ $t, w \leftarrow g((t_A, w_A), (t_B, w_B))$ $\quad$ // (see
$\quad\quad\quad\quad$ eq. (17))
$\quad\quad\quad$ // Add terms to the numerator
$\quad\quad\quad\quad$ of eq. (33)...
$\quad\quad\quad$ $p \leftarrow p_{\text{swap}} (1 - p_{\text{swap}})^{k-1}$
$\quad\quad\quad$ $p \leftarrow p \cdot T[\ell, t_A] \cdot T[\ell, t_B]$
$\quad\quad\quad$ **if** $k = 1$ **then**
$\quad\quad\quad\quad$ // ... in the case of a
$\quad\quad\quad\quad\quad$ single swap (eq. (34))
$\quad\quad\quad\quad$ $W[\ell + 1, t] \leftarrow W[\ell + 1, t] + w \cdot p$
$\quad\quad\quad$ **else**
$\quad\quad\quad\quad$ // ... in the case of
$\quad\quad\quad\quad\quad$ multiple swaps (eq. (36))
$\quad\quad\quad\quad$ **for** $t_{\text{fail}} \in \{0, 1, 2, \ldots, t_{\text{trunc}} - 1\}$ **do**
$\quad\quad\quad\quad\quad$ $t_{\text{deliver}} \leftarrow t + t_{\text{fail}}$
$\quad\quad\quad\quad\quad$ add $w \cdot p \cdot C[\ell + 1, k - 1, t_{\text{fail}}]$ to
$\quad\quad\quad\quad\quad\quad$ $W[\ell + 1, t_{\text{deliver}}]$
$\quad\quad\quad\quad$ **end**
$\quad\quad\quad$ **end**
$\quad\quad$ **end**
$\quad$ **end**
end
for $t \in \{1, 2, \ldots, t_{\text{trunc}}\}$ **do**
$\quad$ // Normalize by dividing by the
$\quad\quad$ denominator of eq. (33)
$\quad$ $W[\ell + 1, t] \leftarrow W[\ell + 1, t] / T[\ell + 1, t]$
end

---

except for the following remarks. First, in the algorithm, the sum in eq. (32) is truncated at the pre-specified truncation time t_{trunc}. That this truncation yields correct probabilities $\Pr(T_{\ell+1} = t)$ for all $t \in \{0, 1, \ldots, t_{\text{trunc}}\}$ follows from the fact that $\Pr(T_{\ell+1} = t | K_\ell > t) = 0$ since the generation of entanglement over any number of hops takes at least a single time step. Second, the convolutions in eq. (31) can be computed iteratively over k by noting that $\Pr(T_{\ell+1} = t | K_\ell = k + 1)$ equals the convolution of $\Pr(T_{\ell+1} = t | K_\ell = k)$ and $m_\ell(t)$. Moreover, for a single convolution we use a well-known algorithm based on Fast Fourier Transforms [54] which we denote by fast_convolution_algorithm in algorithm 2. This subroutine computes the convolution of two arrays of size t_{trunc} in time $\Theta(t_{\text{trunc}} \log t_{\text{trunc}})$.

The time complexity of the deterministic algorithm 2 equals $\Theta(n \cdot t_{\text{trunc}}^2 \log t_{\text{trunc}})$: the iteration over a single level is dominated by the $\Theta(t_{\text{trunc}}^2 \log t_{\text{trunc}})$ runtime of the convolutions in eq. (31) because eqs. (30) and (32) are performed in linear time in t_{trunc} by looping through an array of t_{trunc} elements. In sec. V-B, we give an explicit expression for the truncation time t_{trunc} which ensures that $\Pr(T_n \leq t_{\text{trunc}}) \geq 0.99$. This expression is polynomial in the number of repeater segments, which implies that algorithm 2 runs in polynomial time in the number of segments also.

We extend our deterministic algorithm to also compute the average Werner parameter $W_n(t) := E[W_n | T_n = t]$ of the end-to-end link produced at time t by a 2^n-segment SWAP-ONLY repeater chain (see sec. III-B). The computation of the average Werner parameter at each level from 0 to n is performed after completion of the computation of the waiting time probabilities at the same level.

Let us explain the algorithm here (see algorithm (3) for pseudocode). At the base level the fidelity $W_0(t)$ equals the constant Werner parameter w_0 as in sec. III-B for all $t \in \{1, 2, \ldots, t_{\text{trunc}}\}$. At a higher level, the Werner parameter of a link which is delivered at time t is the output of g_W from eq. (18), averaged over all possible *realizations* of waiting times T_n which yield $T_n = t$. In order to precisely define what we mean by 'realization', note that the waiting time T_n and average Werner parameter W_n as expressed recursively in sec. III-B are a function of K_n copies of (T_{n-1}, W_{n-1}), the waiting time and Werner parameter at one level lower. Regarding (T_n, W_n) as a function with K_n and all such copies of (T_{n-1}, W_{n-1}) as input, we define a 'realization' of (T_n, W_n) as its evaluation on particular instances of these copies.

Using the notion of realization, we obtain the Werner parameter of the 2^ℓ-hop link at levels $\ell \in \{1, 2, \ldots, n\}$, given that it was produced at time t:

$$W_\ell(t) = \frac{\sum\limits_{\substack{r: \\ r \text{ delivers link at } t}} p_\ell(r) \cdot W_\ell^{\text{av}}(r)}{\sum\limits_{\substack{r: \\ r \text{ delivers link at } t}} p_\ell(r)} \qquad (33)$$

where r is a realization of (T_ℓ, W_ℓ) and $W_\ell^{\text{av}}(r)$ denotes the average Werner parameter of the 2^ℓ-hop that realization r delivers with $p_\ell(r)$ its probability of occurrence.

In what follows, we will derive expressions for $p_\ell(r)$ and $W_\ell^{\mathrm{av}}(r)$. This will give us an explicit expression for $W_\ell(t)$ and it is this expression that our algorithm evaluates. We distinguish between two cases of realizations for computing $p_\ell(r)$. In the first case, only a single swap (i.e. $K_\ell = 1$) is needed to produce the 2^ℓ-hop entanglement, i.e. the first swap from level $\ell - 1$ to ℓ is successful. The realizations r that belong to this case can be parameterized by the times t_A and t_B at which the two $2^{\ell-1}$-hop links are generated. The total probability of occurrence of these realizations, each of which delivers a 2^ℓ-hop link at time $g_{\mathrm{T}}(t_A, t_B)$ (see eq. (10)), is given by

$$p_\ell(r) = \Pr(K_\ell = 1)\Pr(T_\ell = t_A)\Pr(T_\ell = t_B) \quad (34)$$

and the average Werner parameter of the produced 2^ℓ-hop entangled link is

$$W_\ell^{\mathrm{av}}(r) = g_{\mathrm{W}}((t_A, W_{\ell-1}(t_A)), (t_B, W_{\ell-1}(t_B))) \quad (35)$$

where g_W is given in eq. 18.

In the second case, at least a single entanglement swap to produce 2^ℓ-hop entanglement fails. Note that the average Werner parameter only depends on the states of the two $2^{\ell-1}$-hops that are produced as input to the *last* swap since the entanglement inputted into the failing swaps is lost. In the case of multiple swaps we can therefore group together the realizations for which the following four quantities are identical: the waiting times t_A and t_B for the production of the last two $2^{\ell-1}$-hop links with in addition the number of swaps k and the time t_{fail} that these failed swaps need. The total probability of occurrence of such a group of realizations equals the product of four probabilities,

$$\begin{aligned} p_\ell(r) &= \Pr(K_\ell = k) \cdot \Pr(T_\ell = t_{\mathrm{fail}}|K_\ell = k-1) \\ &\quad \cdot \Pr(T_\ell = t_A) \cdot \Pr(T_\ell = t_B) \end{aligned} \quad (36)$$

while the average Werner parameter $W_\ell^{\mathrm{av}}(r)$ of the 2^ℓ-hop that is produced by each of these realizations is identical to the first case and is given in eq. (35). Each realization in this group delivers a 2^ℓ-hop link at time $t_{\mathrm{fail}} + g_{\mathrm{T}}(t_A, t_B)$ (see eq. (10)).

Our algorithm loops over each group of realizations, evaluates their probabilities of success in eqs. (34) and (36) and their average Werner parameter in eq. (35) and subsequently computes $W_\ell(t)$ using eq. (33). The domain of the time parameters t_A, t_B and t_{fail} is bounded from above by t_{trunc} since no short-range link that is used to produce a long-range link at time $\le t_{\mathrm{trunc}}$ can take longer than t_{trunc}. Also, the total number of swaps K_n runs up to t_{trunc} since it cannot exceed the time at which the end-to-end link is delivered by the same reasoning as the truncation of the sum in eq. (32), i.e. $\Pr(T_{n+1} = t|K_n > t) = 0$. The pseudocode of the deterministic algorithm for computing the average Werner parameter can be found in algorithm 3.

The time complexity of the Werner-parameter algorithm can be inferred directly from algorithm 3 by the four loops with domain of size $\Theta(t_{\mathrm{trunc}})$, which implies that the full time complexity is $\Theta(n \cdot t_{\mathrm{trunc}}^4)$. This is polynomial in the number of repeater chain segments (see sec. V-B).

C. Possible extensions

In this section, we give examples of possible extensions of the Monte Carlo algorithm and the deterministic algorithm. First, we provide an example of how the two algorithms can be extended to different quantum state and noise models than the Werner states and depolarizing decoherence noise used in this work. We also give an example of an extension to a different network topology than a chain. We finish the section by sketching what is needed to extend the deterministic algorithm to the d-DIST-SWAP protocol in the future.

An example of applying the algorithms to more general quantum states is to track states that are diagonal in the Bell basis, i.e. we assume that the generated single-hop states can be written as

$$\sum_{j\in\{\pm\}} \sum_{k\in\{\pm\}} p_{j,k} |\phi_{j,k}\rangle\langle\phi_{j,k}|$$

where $|\phi_{+\pm}\rangle := (|0\rangle \otimes |0\rangle \pm |1\rangle \otimes |1\rangle)/\sqrt{2}$ and $|\phi_{-\pm}\rangle := (|0\rangle \otimes |1\rangle \pm |1\rangle \otimes |0\rangle)/\sqrt{2}$ are the four Bell states and the Bell coefficients $p_{j,k}$ are probabilities which sum to 1. The implementation of the Monte Carlo method would have the Werner parameter W_n replaced by a joint random variable on the four[2] Bell coefficients $(p_{++}, p_{+-}, p_{-+}, p_{--})$, while the deterministic algorithm would compute the average over each of these four coefficients individually in a fashion similar to the average of the Werner parameter (eq. (33)). Successful entanglement swap and distillation operations both map Bell-diagonal states to Bell-diagonal states [34] and could thus each be formulated as an operation on the four Bell coefficients.

An example of a different model of memory decoherence noise (currently eq. (3)) is the application of the Pauli operator $Z := |0\rangle\langle0| - |1\rangle\langle1|$ on one of the two qubits with probability

$$q(\Delta t) := \frac{1}{2}\left(1 - e^{-\Delta t/T_{\mathrm{coh}}}\right)$$

where Δt is the time that the state has resided in memory and T_{coh} is the joint coherence time of the two memories that hold the two qubits. This probabilistic application of Z acts on the Bell coefficients as

$$p_{j,k} \mapsto (1 - q(\Delta t)) \cdot p_{j,k} + q(\Delta t) \cdot p_{m,\ell}$$

where $p_{m,\ell}$ is the coefficient belonging to $|\phi_{m,\ell}\rangle := (\mathbb{1}_2 \otimes Z)|\phi_{j,k}\rangle$ with $\mathbb{1}_2 := |0\rangle\langle0| + |1\rangle\langle1|$. Lastly, the algorithm could be generalized by modelling the swapping and distillation operations as noisy operations by concatenating the perfect operation with a noise map that can be written as operation on the four Bell coefficients.

In addition to more general state and noise models, both algorithms could also be applied to more general network topologies than a chain. An example is the generation of a Greenberger-Horne-Zeilinger (GHZ) state [55] in a star network, where there is a single central node and each of the other nodes (the leaves) is connected to this single central node only. All leave nodes start by generating an elementary link

---

[2]In fact, tracking only three of these coefficients already completely characterize a Bell-diagonal state since they sum to one.

with the central node, after which the central node performs a local operation to convert these links into a single GHZ state on all the leave nodes, e.g. by a combination of two-qubit controlled-rotation gates and single-qubit measurements [49]. Similar to our model of the swap operation, we could model the local operation that produces the GHZ state as probabilistic, motivated by probabilistic two-qubit operations in linear photonics [56]. In the same spirit as the SWAP-ONLY protocol and our analysis of it in sec. III, the central node waits for all links to have been generated (which corresponds to the maximum of their individual waiting times) while failure of the local operation requires regeneration of the elementary links (which corresponds to the geometric compound sum). Since both maximums and geometric sums of random variables can be treated by the two algorithms, both could be used to sample the produced state and waiting time in the star network.

In general, the Monte Carlo algorithm could be applied if one can express the random variables describing waiting time and fidelity as a function of random variables whose probability distribution is known. For the deterministic algorithm, on the other hand, one should find efficiently computable expressions for the probability distributions of these random variables. These two cases are different: indeed, for the d-DIST-SWAP protocol, we have found a Monte Carlo algorithm but have been unable to formulate a deterministic algorithm. We finish this section by elaborating on this difference for the specific case of the d-DIST-SWAP protocol and sketch an opportunity for extending the deterministic algorithm to it. For the SWAP-ONLY protocol, we were able to design a Monte Carlo sampler of the waiting time and fidelity by expressing the corresponding random variables at level ℓ as a function of those at level $\ell - 1$. In addition, we could express the probability distributions recursively also, which enabled us to formulate the deterministic algorithm. For the d-DIST-SWAP protocol, we succeeded in performing the first step, i.e. find a recursive expression for the random variables (see sec. III-D), but have been unable to express the probability distributions in a recursive fashion. The difficulty lies in the fact that waiting time cannot be analyzed independently of the fidelity of the input states, since the distillation success probability is a function of the produced states contrary to the swapping success probability p_{swap} (see also last paragraph in sec. III-D for more explanation). The resulting two-way dependence between waiting time and fidelity renders eq. (32) incorrect if we simply replaced p_{swap} by the distillation success probability. For this reason, the key to finding a deterministic algorithm for computing waiting time and fidelity for the d-DIST-SWAP protocol is to find an efficiently computable expression for the probability function of (T_ℓ, W_ℓ) as a function of $(M_{\ell-1}, V_{\ell-1})$, i.e. a correct alternative to eq. (32).

V. BOUNDS ON THE MEAN WAITING TIME

In this section, we first show how to use the deterministic algorithm from section IV-B to obtain bounds on the mean of the waiting time T_n, which improve upon a common analytical approximation. Then we give an explicit expression for the

choice of truncation time in the algorithm for which 99% of probability mass of T_n is captured.

A. Numerical mean using the deterministic algorithm

Here, we show how to obtain bounds on the mean of T_n using the deterministic algorithm from section IV-B. Such bounds are interesting since a common approximation to the mean in the regime of small success probabilities p_{gen} and p_{swap}, the *3-over-2-formula* [17], [23], [31], [44]

$$E[T_n] \approx \left(\frac{3}{2p_{\text{swap}}} \right)^n \cdot \frac{1}{p_{\text{gen}}}, \qquad (37)$$

overestimates the waiting time for large success probabilities. For example, it can be seen in [45, fig. 7(a)] (reproduced in this work as fig. 7, top plot) that for $p_{\text{gen}} = p_{\text{swap}} = 1$, the ratio [true mean]/[approximation] of the true mean $E[T_n]$ and the approximation in eq. (37) decreases as a function of the number of segments and equals 0.2 for a chain of 16 segments, i.e. an overestimation by a factor $\frac{1}{0.2} = 5$. In fig. 7 (bottom plot), it can be seen that this overestimation grows to more than a factor $\frac{1}{0.05} = 20$ for a chain of 2048 segments.

The bounds are obtained in two steps. First, we perform the deterministic algorithm to compute the probability distribution of T_n, as described in section IV-B. Since this probability distribution is only computed by the algorithm on the truncated domain $\{0, 1, \ldots, t_{\text{trunc}}\}$, we cannot calculate the mean of T_n. Instead, we compute its *empirical mean*, which we define for random variable X with the nonnegative integers as domain as

$$E[X, t_{\text{trunc}}] := \sum_{t=1}^{t_{\text{trunc}}} \Pr(X \geq t). \qquad (38)$$

Note that the empirical mean reduces to the real mean for $t_{\text{trunc}} \to \infty$ (see section II-C).

In the second step, we quantify how well the empirical mean of T_n approximates its real mean. We need two tools for doing so. As first tool, we introduce the random variable T_n^{upper}, which is identical to T_n except for the fact that the two links required for the entanglement swap are produced sequentially at every level rather than in parallel. We proceed analogously to the first step: we perform a modified version of the deterministic algorithm to compute the probability distribution of T_n^{upper} and we compute its empirical mean (details and formal definition of T_n^{upper} can be found in appendix B). In contrast to T_n, we are able to compute the real mean of T_n^{upper}, which equals $E[T_n^{\text{upper}}] = (2/p_{\text{swap}})^n \cdot 1/p_{\text{gen}}$ (proof in appendix B). The second tool is the following proposition, which states that for T_n the empirical mean converges at least as fast to the real mean with increasing truncation time as for T_n^{upper}.

Proposition V.1. *The difference between the real mean and the empirical mean (eq. (38)) of the waiting time is bounded as*

$$0 \leq E[T_n] - E[T_n, t_{\text{trunc}}] \leq \left(\frac{2}{p_{\text{swap}}} \right)^n \cdot \frac{1}{p_{\text{gen}}} - E[T_n^{\text{upper}}, t_{\text{trunc}}]$$

and the two bounds coincide for $t_{\text{trunc}} \to \infty$. The random variable T_n^{upper} is formally defined in appendix B.

The main tool for proving proposition V.1 is the fact that T_n^{upper} stochastically dominates T_n for every nesting level n, which means that $\Pr(T_n \geq t) \leq \Pr\left(T_n^{\text{upper}} \geq t\right)$ for all $t \in \{0, 1, 2, \dots\}$. We formally prove the proposition and give a more detailed version of the computation of the probability distribution of T_n^{upper} in appendix B.

B. Choosing a truncation time for the deterministic algorithm

The truncation time that is inputted into the deterministic algorithm determines how much probability mass will be captured by the algorithm. The captured probability mass can be bounded from above using Markov's inequality:

$$\Pr(T_n \geq t_{\text{trunc}}) \leq E\left[T_n\right] / t_{\text{trunc}}. \tag{39}$$

We upper bound the mean of T_n in eq. (39) by invoking proposition V.1 with $t_{\text{trunc}} = 0$. The latter reduces to $E\left[T_n\right] \leq (2/p_{\text{swap}})^n \cdot 1/p_{\text{gen}}$ and thus implies

$$\Pr(T_n \geq t_{\text{trunc}}) \leq \left(\frac{2}{p_{\text{swap}}}\right)^n \cdot \frac{1}{p_{\text{gen}} \cdot t_{\text{trunc}}}.$$

Consequently, setting

$$t_{\text{trunc}} = \left(\frac{2}{p_{\text{swap}}}\right)^n \cdot \frac{1}{p_{\text{gen}}} \cdot \frac{1}{1 - 0.99} \tag{40}$$

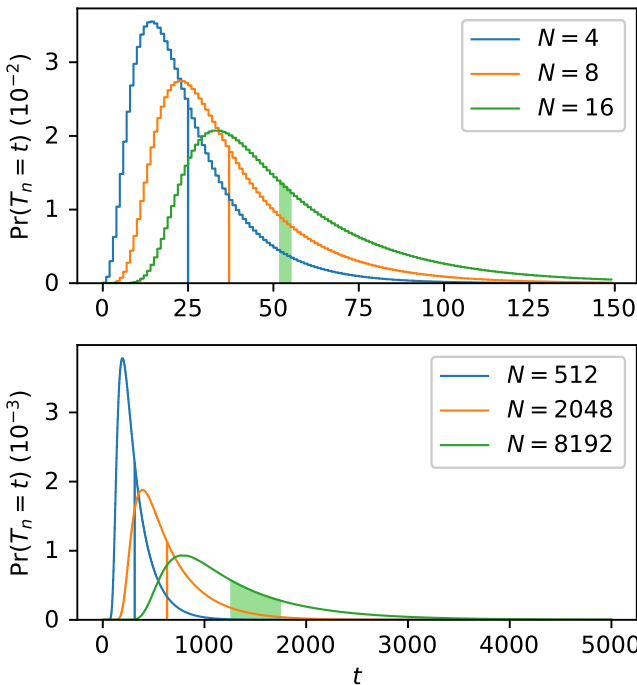ensures that an end-to-end link will be produced with probability $\Pr(T_n < t) = 99\%$.

Fig. 2: (Color online) Probability distributions of the waiting time and bounds on the mean (vertical shaded areas, see V-A) calculated by the deterministic algorithm for the SWAP-ONLY protocol. The repeater chain parameters are $p_{\text{gen}} = 0.1$, $p_{\text{swap}} = 0.9$, and the number of repeater segments is given by N. The top plot recovers the results from Shchukin et al. [45, Fig.10(a)]. Computation time ≈ 5 seconds for $N = 8192$.

VI. NUMERICAL RESULTS

In this section we investigate different repeater chain protocols with the help of our two algorithms. We start with the SWAP-ONLY protocol, first considering the waiting time distribution of the first produced end-to-end link and subsequently also its average fidelity. We also show how fidelity and waiting time are affected by the d-DIST-SWAP protocol. Finally we consider the effect of including the communication time in swap operations.

Our proof-of-principle implementation can be found in [57]. The reported computation times have been obtained from single-threaded computations on commodity hardware (specifically: a single logical processor of an Intel i7-4770K CPU @ 3.85 GHz). In the plot captions in this section, we state the computation time for the largest number of repeater segments because computing the distribution of (T_n, W_n) requires finding the distribution of (T_{n-1}, W_{n-1}) first (see sec. III).

First we consider the waiting time in the SWAP-ONLY protocol. Our algorithms are able to recover the results from Shchukin et al. [45], both the full distribution of waiting time exactly (fig. 2, top plot) as well as its mean up to arbitrary precision (fig. 7, top plot), and extend these results from 16 to 8192 and to 2048 repeater segments, respectively (figs. 2, 3 and 7). In fig. 3 we compare results from both our Monte Carlo and deterministic algorithm and find that there is good agreement between the two. For high swapping success probability p_{swap} the deterministic algorithm can compute probability distributions up to thousands of nodes, as illustrated in fig. 2. For small p_{swap}, we have found that the number of repeater segments $N = 2^n$ that we can simulate is limited in practice. This is a consequence of the fact that t_{trunc} grows fast in N for small p_{swap} if we want the guarantee that 99% of the probability distribution is captured (see eq. (40)), and the polynomial scaling in t_{trunc} of the algorithm's runtime.

Secondly, we consider the average fidelity of the SWAP-ONLY and d-DIST-SWAP protocols. We investigate the SWAP-ONLY protocol with a small number of segments ($N = 1, 2, 4$), see fig. 4. We observe that fidelity stabilizes as the waiting time increases, and it stabilizes at values for which the state remains entangled in spite of the absence of distillation. Again, the deterministic and Monte Carlo algorithms show good agreement. Adding the calculation of fidelity increases the time complexity of the deterministic algorithm, which reduced the maximum number of segments that we could simulate. We found that the Monte Carlo algorithm is able to simulate a larger number of segments, as its computational complexity is unchanged when also tracking the fidelity.

Third, we consider the d-DIST-SWAP protocol. In fig. 5, we study the effects of distillation in a repeater chain of 4 segments comparing one distillation round ($d = 1$) against no distillation rounds ($d = 0$) for two different memory coherence times. We first observe the increase in the waiting times caused by the generation of the additional links necessary for distillation. An increase in waiting time is accompanied by an increase in memory decoherence, which implies that the degree to which distillation is beneficial depends on the
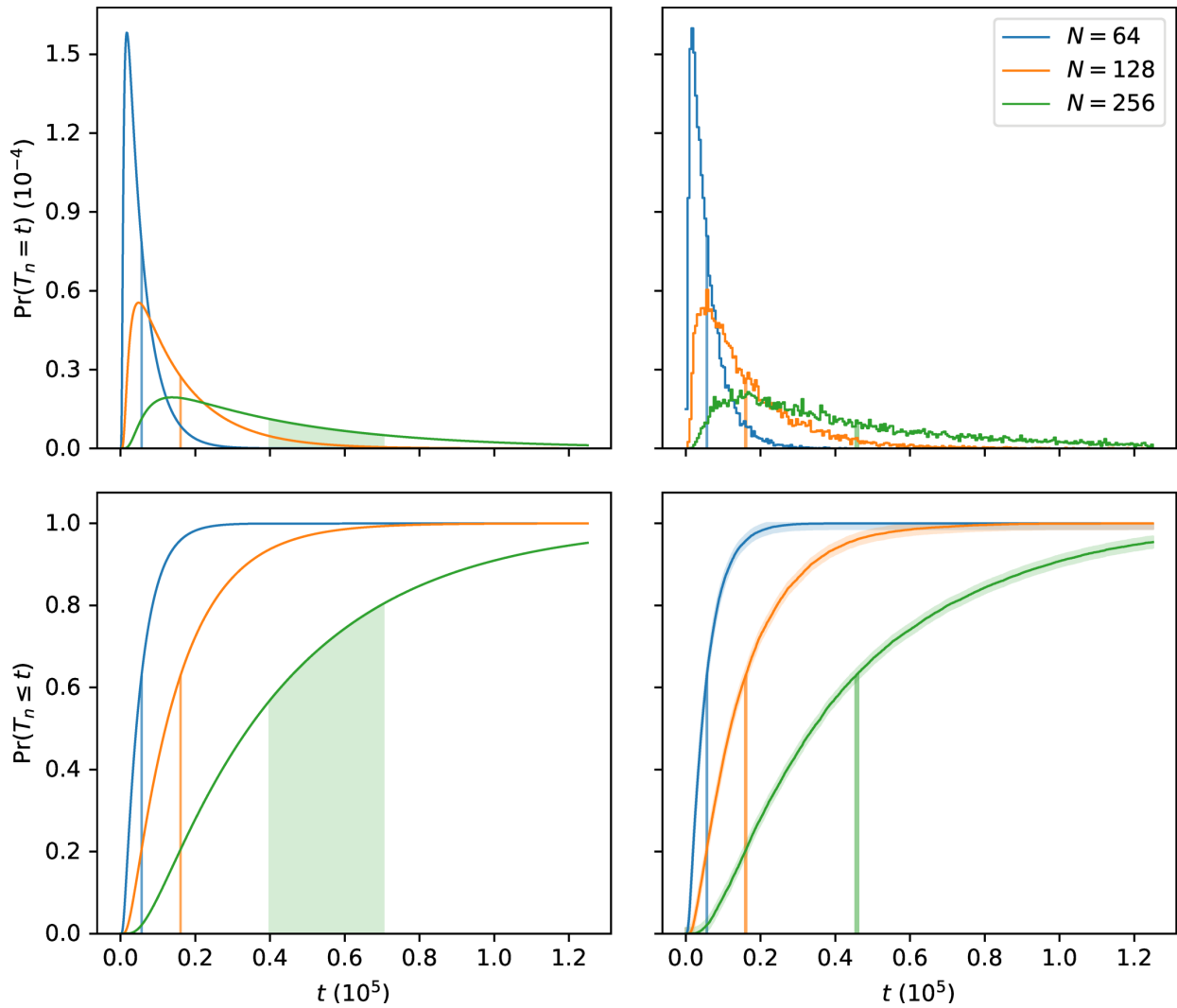
Fig. 3: (Color online) The probability distributions of waiting time for repeater chains with varying number of segments N for the SWAP-ONLY protocol, computed using the deterministic algorithm (left plots) and generated from 15,000 samples from the Monte Carlo simulation (right plots). Vertical shaded areas indicate bounds on the mean in the left plots (see sec. V-A) and the sample mean $\pm$ one standard error in the right plot. In the bottom right plot the shaded bands are confidence bands for a probability $z = 0.01$ that the actual distribution lies outside of these bands, obtained from the Dvoretzky-Kiefer-Wolfowitz inequality [53] discussed in sec. IV-A. The repeater chain parameters are $p_{\mathrm{gen}} = 0.1$, $p_{\mathrm{swap}} = 0.5$. The computation time for the deterministic algorithm ≈ 9 minutes, while for the Monte Carlo algorithm ≈ 30 minutes.

memory coherence time. The values for the coherence time we chose allow to show both types of behavior.

Finally, we incorporate the communication time for the entanglement swap into our model following sec. IV-A. Fig. 6 shows how the output probability distributions change when we include this communication time. We confirm that, as stated by Brask and Sørensen [17], omitting this communication time gives a good approximation for small p_{gen}, but not for larger p_{gen}.

In order to get a rough analytical understanding of the probability distributions for the waiting time that our algorithms have computed, we fitted a generalized extreme value (GEV) distribution to them, which has cumulative distribution function

$$\Pr(X \leq t) = \exp\left(-(1 + \xi s)^{-1/\xi}\right) \qquad (41)$$

where X is a random variable following the GEV distribution, $s = (t - \mu)/\sigma$, and $\xi > 0$, $\sigma > 0$ and $\mu \in \mathbb{R}$ are the free parameters [58]. Fig. 8 shows a typical result of such a fit. We find that the fit seems rather close to the computed distribution, although the difference in the means indicates that the fit should only be used to make approximate statements.

A good fit could provide a speedup for the deterministic algorithm, since the algorithm computes the distribution at each level from the distribution at the previous level. To be precise, the algorithm's runtime can be reduced by starting the computation at the fitted distribution instead of computing the distribution at level n, and subsequently using this distribution to have the algorithm compute the distribution at level $n + 1$. In order to ensure that the distribution at the final level $> n$ still approximates the real distribution, careful analysis
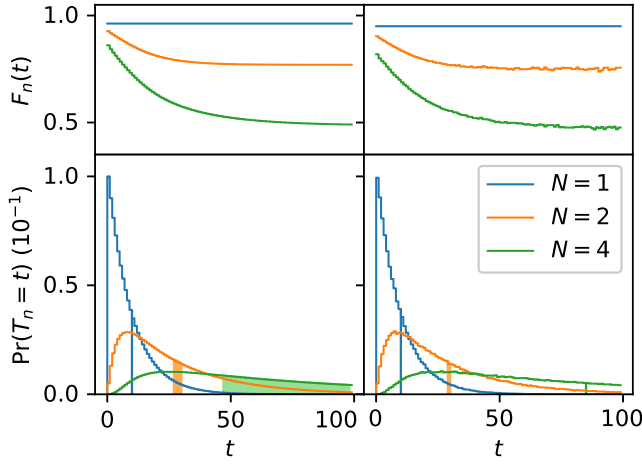
Fig. 4: (Color online) The average fidelity of links delivered at time t by an N-segment SWAP-ONLY repeater chain (top row), and the corresponding probability distributions (bottom row), from both the deterministic algorithm (left column) and the Monte Carlo algorithm (right column) using 250,000 samples. For the deterministic figures the vertical shaded areas indicate numerical bounds on the mean (see sec. V-A), and for the Monte Carlo figures these indicate the sample mean $\pm$ one standard error. The repeater chain parameters for SWAP-ONLY protocol are $p_{gen} = 0.1$, $p_{swap} = 0.5$, $T_{coh} = 50$ time steps and the fidelity of the elementary links equals $F_0 = 0.95$, which corresponds to Werner parameter $w_0 = (4 \cdot 0.95 - 1)/3 \approx 0.93$ following eq. (2). The computation time for the deterministic algorithm ≈ 15 minutes, while for the Monte Carlo algorithm ≈ 20 seconds.

of the acquired error of the distribution at higher levels is required. We leave such error accumulation analysis, based on a distribution that forms a phenomenological fit, for future work.

VII. DISCUSSION

Quantum networks enable the implementation of communication tasks with qualitative advantages with respect to classical networks. A key ingredient is the delivery of entanglement between the relevant parties. In this work, we provide two algorithms for computing the probability that an entangled pair is produced by a quantum repeater chain at any given time and also show how to compute the pair's Bell-state fidelity. The first algorithm is a probabilistic Monte Carlo algorithm whose precision can be rigorously estimated using standard techniques. The second one is deterministic and exact up to a chosen truncation time.

Both algorithms run in time polynomial in the number of nodes, which is faster than the exponential runtime of previous algorithms. The workhorse behind the improved complexity is a formal recursive definition of the waiting time and the state produced by the chain. We developed an open source proof-of-principle implementation in [57], which allows to analyze repeater chains with several thousands of segments for some parameter regimes.
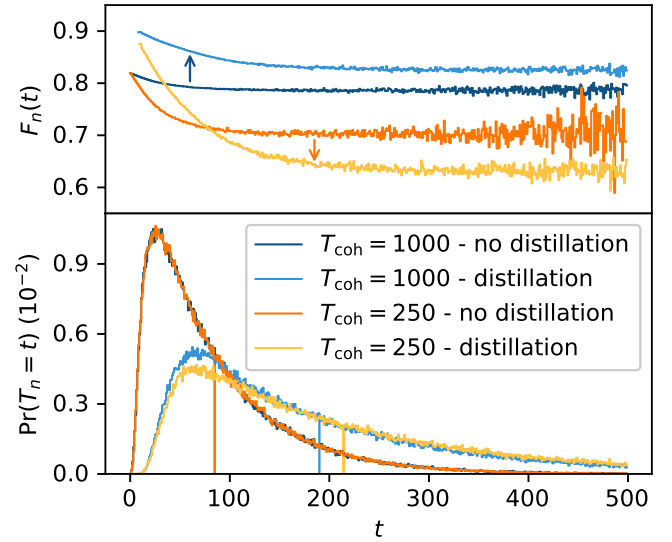
Fig. 5: (Color online) Comparison between the d-DIST-SWAP protocol with a single distillation round on every level ($d = 1$) and the SWAP-ONLY protocol (no distillation) for a repeater chain with $N = 4$ segments, for longer and shorter memory coherence times T_{coh}. The additional repeater chain parameters are $p_{gen} = 0.1$, $p_{swap} = 0.5$, and $F_0 = 0.95$, which corresponds to $w_0 \approx 0.93$. While the goal of distillation is to improve the fidelity of delivered links, when the coherence time is too short compared to the time needed to deliver a link, adding distillation actually decreases the fidelity (orange arrow). For longer coherence times adding distillation does improve the fidelity (blue arrow). In both cases the waiting time increases because entanglement distillation requires more links to be generated. For the SWAP-ONLY protocol, the waiting time is independent of the memory coherence time (in contrast to fidelity), which can be observed from the identical waiting times in the bottom plot. Each curve has been generated from 250,000 Monte Carlo algorithm samples. Computation time ≈ 15 seconds without distillation, ≈ 100 seconds with distillation.

The deterministic algorithm is the fastest of the two for a large set of success probabilities for generating single-hop entanglement p_{gen} and entanglement swapping p_{swap}. The Monte Carlo algorithm could be sped up to a factor proportional to the number of samples by parallelization. A second option to reduce the runtime would be to construct estimates of the random variables at each level of a repeater chain and sample from the estimates to estimate the following level. A careful analysis would be necessary to ensure that the speed up does not vanish when taking the accumulated precision error into account.

We have been able to adapt our algorithms to several protocols for repeater chains. More concretely, we have studied the SWAP-ONLY protocol and two generalizations. The first one includes distillation d-DIST-SWAP, the second one takes into account the communication time in the swap operations. We believe that the tools we have developed here can be
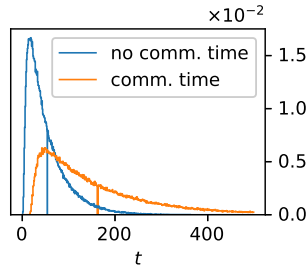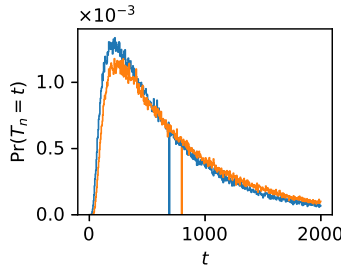
Fig. 6: (Color online) Waiting time distributions with and without communication time for entanglement swapping for the SWAP-ONLY protocol, with entanglement generation success probabilities $p_{\text{gen}} = 0.1$ (left) and $p_{\text{gen}} = 0.9$ (right), generated from 250,000 samples from the Monte Carlo algorithm. The vertical bars indicate the mean $\pm$ one standard error. As stated by Brask and Sørensen [17] omitting this communication time gives a good approximation when p_{gen} is small. The repeater chain has $N = 16$ segments and $p_{\text{swap}} = 0.5$. Computation time ≈ 5 minutes per curve.

extended to several other protocols without losing the polynomial runtime. Some examples which we leave for further work include tracking the full density matrix, variations of d-DIST-SWAP with unequal spacing of the nodes or with a number of segments different than a power of two, and the investigation of more general network topologies. Inspired by hardware, it would also be interesting to model decaying memory efficiency and nodes that can not generate entanglement concurrently with both adjacent neighbors.

In summary, we have proposed two efficient algorithms to characterize the behavior of repeater chain protocols. We expect our algorithms to find use in the study and analysis of future quantum networks. Moreover, the existence of protocols capable of efficiently characterizing the state produced opens the door to real-time decision taking at the nodes based on this knowledge.

APPENDIX A
DISTILLATION ON WERNER STATES

In this appendix, we find the state after successful entanglement distillation on two Werner states. Performing entanglement distillation on two Werner states with Bell-state fidelities F_A and F_B yields a state with Bell-state fidelity [34]

$$\frac{\left(F_A F_B + \frac{1}{9}\bar{F}_A \bar{F}_B\right)}{p_{\text{dist}}} \tag{42}$$

where the probability of success p_{dist} is given by

$$F_A F_B + \frac{1}{3}F_A \bar{F}_B + \frac{1}{3}\bar{F}_A F_B + \frac{5}{9}\bar{F}_A \bar{F}_B \tag{43}$$

where we have denoted $\bar{F} = 1 - F$. Although the output state is not a Werner state, it is always possible to transform it into a Werner state with the same Bell-state fidelity by local operations. We rewrite eqs. (42) and (43) as function of the Werner parameters w_A and w_B of the input states rather than their fidelities F_A and F_B using eq. (2), which yields eqs. (5) and (6).
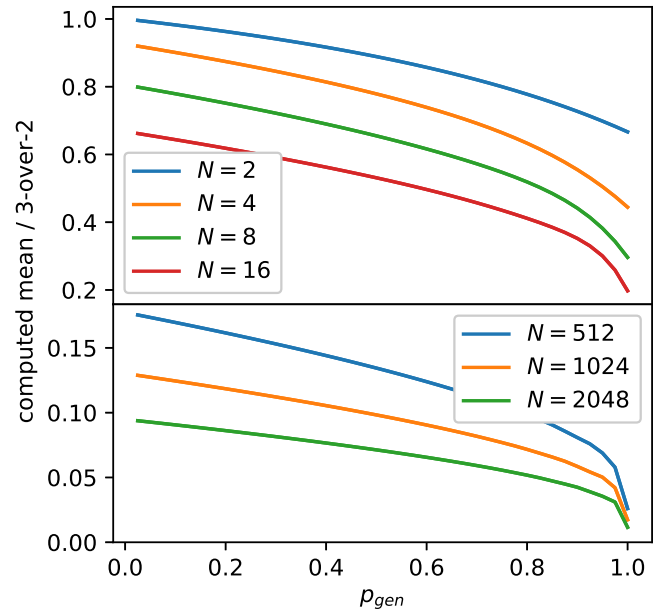
Fig. 7: (Color online) Ratio between the bound on the mean computed by the deterministic algorithm and the 3-over-2 approximation eq. (37) as a function of entanglement generation success probability p_{gen} with deterministic swapping ($p_{\text{swap}} = 1$). For each number of segments N, the figures show two lines: one for the upper and lower bound on the mean (see sec. V-A). The fact that for each N only a single thick line rather than two lines can be seen indicates that the bounds on the mean almost coincide. The top figure recovers work by Shchukin et al. [45, fig. 7(a)], whose exponential-time algorithm based on Markov chains is able to compute the mean exactly while our algorithms can get arbitrarily tight bounds on the mean (deterministic algorithm) or approximate the mean with arbitrary precision (Monte Carlo algorithm) at the benefit of polynomial runtime. The runtime improvement over the Markov-chain approach allows us to extend the results of Shchukin et al. to more than 2000 segments (bottom figure). Each curve was generated by running the deterministic algorithm for 40 different values of p_{gen} ($0.025, 0.05, \ldots, 1$) and the truncation time was set to $t_{\text{trunc}} = 1000$. Computation time for each curve $\lesssim 2$ seconds.

APPENDIX B
COMPUTATION OF T_n^{upper}

In this appendix, we first prove proposition V.1 and subsequently show how a modified version of the deterministic algorithm from section IV-B computes the empirical mean $E\left[T_n^{\text{upper}}, t_{\text{trunc}}\right]$ from eq. (38).

A. Proof of proposition V.1

The random variable T_n^{upper} for $n \in \{0, 1, 2, \ldots\}$ is recursively defined as

$$T_0^{\text{upper}} = T_0 \tag{44}$$

$$M_{n+1}^{\text{upper}} = (T_n^{\text{upper}})^{(A)} + (T_n^{\text{upper}})^{(B)} \tag{45}$$

$$T_{n+1}^{\text{upper}} = \sum_{k=1}^{K_n} (M_n^{\text{upper}})^{(k)} \tag{46}$$
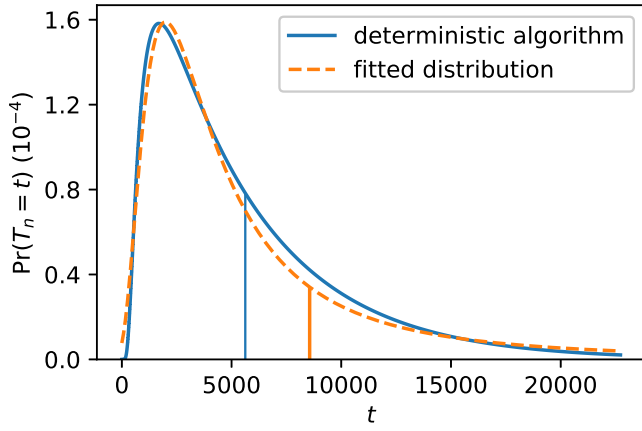
Fig. 8: (Color online) Waiting time distribution (64 segments, $p_{\text{gen}} = 0.1$, $p_{\text{swap}} = 0.5$), computed with the deterministic algorithm (sec. IV-B), and a fit to the same distribution using the generalized extreme value (GEV) distribution (see eq. (41); fitting parameters: $\xi \approx -0.5997$, $\mu \approx 3092.7$, $\sigma \approx 2694.9$). Although the two distributions seem to coincide fairly well by eye, the difference between the means (vertical lines) is relatively large (a factor ≈ 1.5). For different number of segments and success probabilities p_{gen} and p_{swap}, the difference between fitted and computed distribution is similar.

where T_0 is defined in section III-A and K_n is geometrically distributed with parameter p_{gen} for all n.

For random variables X and Y, both defined on a subset $\mathcal{D}$ of the nonnegative integers, we say that the random variable Y stochastically dominates the random variable X, denoted by $X \leq_{\text{st}} Y$, if $\Pr(X \leq x) \geq \Pr(Y \leq x)$ for all $x \in \mathcal{D}$. We prove that T_n^{upper} stochastically dominates T_n for all $n \geq 0$, for which we need the following lemma.

Lemma B.1. *Let X, Y, A and B each be discrete random variables taking values in the nonnegative integers and let X' (Y') denote an i.i.d. copy of X (Y). Then the following hold:*

(a) *If $X \leq_{\text{st}} Y$, then $\max\{X, X'\} \leq_{\text{st}} Y + Y'$.*

(b) *If $X \leq_{\text{st}} Y$, then $A + X \leq_{\text{st}} A + Y$.*

(c) *If $X \leq_{\text{st}} Y$ and $A \leq_{\text{st}} B$, then $A + X \leq_{\text{st}} B + Y$.*

(d) *If $m \in \{1, 2, \dots\}$ and $X \leq_{\text{st}} Y$, then $\sum_{j=1}^{m} X^{(j)} \leq_{\text{st}} \sum_{j=1}^{m} Y^{(j)}$.*

(e) *If K and K' are i.i.d. geometric random variables with parameter p and $X \leq_{\text{st}} Y$, then $\sum_{j=1}^{K} X^{(j)} \leq_{\text{st}} \sum_{j=1}^{K'} Y^{(j)}$*

where we use the notation $X^{(\cdot)}$ to denote an i.i.d. copy of X, following sec. II-C.

Proof. For statement (a), we explicitly use that Y only takes nonnegative values so that we can write

$$\Pr(Y + Y' \leq y) = \sum_{z=0}^{y} \Pr(Y \leq y - z) \Pr(Y' = z).$$

which is, by the fact that any cumulative distribution function is monotone increasing, smaller than

$$
\begin{aligned}
\sum_{z=0}^{y} \Pr(Y \leq y) \Pr(Y' = z) &= \Pr(Y \leq y)^2 \\
&\leq \Pr(X \leq y)^2 \\
&= \Pr(\max\{X, X'\} \leq y)
\end{aligned}
$$

where the inequality is immediate by $X \leq_{\text{st}} Y$. Statement (b) is proven as

$$
\begin{aligned}
\Pr(A + X \leq z) &= \sum_{a=0}^{\infty} \Pr(A = a) \Pr(X \leq z - a) \\
&\geq \sum_{a=0}^{\infty} \Pr(A = a) \Pr(Y \leq z - a) \\
&= \Pr(A + Y \leq z)
\end{aligned}
$$

and statement (c) follows by repeated application of (b):

$$A + X \leq_{\text{st}} A + Y = Y + A \leq_{\text{st}} Y + B = B + Y.$$

Statement (d) can be proven using the fact that $\sum_{j=1}^{m} X^{(j)} = X^{(m)} + \sum_{j=1}^{m-1} X^{(j)}$ and statement (c) by induction on m. For proving statement (e), first note that $\sum_{k=1}^{K} X^{(k)}$ where K is geometrically distributed with parameter p has cumulative distribution function

$$\Pr\left(\sum_{k=1}^{K} X^{(k)} \leq x\right) = p \cdot \sum_{k=1}^{\infty} (1-p)^k \cdot \Pr\left(\sum_{j=1}^{k} X^{(j)} \leq x\right)$$

which is a linear combination of the functions $f_k^X : x \mapsto \Pr\left(\sum_{j=1}^{k} X^{(j)} \leq x\right)$. Positivity of the weights $p \cdot (1-p)^k$ together with the fact that the $f_m^X(x) \geq f_m^Y(x)$ for all $m \in \{1, 2, \dots\}$ and all $x \in \{0, 1, \dots\}$ (see (d)) imply (e). $\qquad\square$

Using lemma B.1, it is straightforward to prove that T_n^{upper} stochastically dominates T_n.

Proposition B.1. *It holds that $T_n \leq_{\text{st}} T_n^{\text{upper}}$ for all $n \geq 0$.*

Proof. We use induction on n. The base case $n = 0$ is immediate since $T_0^{\text{upper}} = T_0$ (eq. (45)). Now suppose that $T_n \leq_{\text{st}} T_n^{\text{upper}}$ for some $n \geq 0$. It follows directly from lemma B.1(a) that $M_n \leq_{\text{st}} M_n^{\text{upper}}$, where M_n is given in eq. (9) and M_n^{upper} in eq. (45). Using lemma B.1(e), we find that the dominance $M_n \leq_{\text{st}} M_n^{\text{upper}}$ implies $T_{n+1} \leq_{\text{st}} T_{n+1}^{\text{upper}}$ where T_{n+1} and T_{n+1}^{upper} are defined in eqs. (8) and (46), respectively. This concludes our proof. $\qquad\square$

Using this stochastic dominance on the waiting time on each nesting level, we are now ready to prove proposition V.1. First, notice that, in contrast to T_n, the mean of T_n^{upper} can be computed analytically.

Lemma B.2.

$$E\left[T_n^{\text{upper}}\right] = \left(\frac{2}{p_{\text{swap}}}\right)^n \cdot \frac{1}{p_{\text{gen}}}$$

Proof. We use induction on n. Since T_0^{upper} equals T_0, which is geometrically distributed with parameter p_{gen}, we have

$E\left[T_0^{\text{upper}}\right] = 1/p_{\text{gen}}$. For the induction step, first note that by linearity of the mean, we have

$$E\left[M_n^{\text{upper}}\right] = E\left[(T_n^{\text{upper}})^{(A)}\right] + E\left[(T_n^{\text{upper}})^{(B)}\right] = 2E\left[T_n^{\text{upper}}\right].$$

The last step is given by Wald's identity [59], which states that the mean of a compound sum equals the product of the mean of the summand and the random variable that is the summation upper bound:

$$E\left[T_{n+1}^{\text{upper}}\right] = E\left[K_n\right] \cdot E\left[M_n^{\text{upper}}\right] = \frac{1}{p_{\text{swap}}} \cdot 2E\left[T_n^{\text{upper}}\right].$$

Closing the recursion relation on $E\left[T_n^{\text{upper}}\right]$ yields the expression in the lemma. $\square$

The following lemma, which states that stochastic domination implies domination of the empirical mean from eq. (38) provides the last step in proving proposition V.1.

Lemma B.3. *Let X and Y be discrete random variables, both defined on the nonnegative integers. If $X \leq_{\text{st}} Y$, then*

$$0 \leq E\left[X\right] - E\left[X, t_{\text{trunc}}\right] \leq E\left[Y\right] - E\left[Y, t_{\text{trunc}}\right]$$

for each $t_{\text{trunc}} \in \{0, 1, \dots\}$. In particular, for $t_{\text{trunc}} = 0$ it follows that $E\left[X\right] \leq E\left[Y\right]$.

Proof. The lower bound is an immediate consequence of positivity of probabilities and

$$E\left[X\right] - E\left[X, t_{\text{trunc}}\right] = \sum_{t=t_{\text{trunc}}+1}^{\infty} \Pr(X \geq t). \qquad (47)$$

The upper bound follows from eq. (47) and the definition of stochastic dominance: $\Pr(X \geq t) \leq \Pr(Y \geq t)$ for all $t \in \{0, 1, \dots\}$. $\square$

Proposition V.1 follows from lemma B.3 by replacing X by T_n and Y by T_n^{upper}, and substituting $E\left[T_n^{\text{upper}}\right]$ by the expression in lemma B.2.

B. Computing the empirical mean of T_n^{upper}

Here, we outline how the deterministic algorithm computes $E\left[T_n^{\text{upper}}, t_{\text{trunc}}\right]$, which is needed for determining a bound on the mean of T_n using proposition V.1. First note that T_n^{upper} and T_n are identical except for the difference between M_n in eq. (9), which equals the maximum of two copies of the waiting time, and M_n^{upper} in eq. (45), which equals their sum. We modify the algorithm to account for this difference by replacing the computation of the probability distribution of M_n in eq. (30) by the convolution

$$\Pr(M_n^{\text{upper}} = t) = \sum_{j=0}^{t} \Pr\left(T_{n-1}^{\text{upper}} = j\right) \Pr\left(T_{n-1}^{\text{upper}} = t - j\right).$$

In order to determine $E\left[T_n^{\text{upper}}, t_{\text{trunc}}\right]$, we first run the modified algorithm to compute the cumulative probability distribution $\Pr\left(T_n^{\text{upper}} \leq t\right)$ for $t \in \{0, 1, \dots, t_{\text{trunc}}\}$, after which we calculate

$$E\left[T_n^{\text{upper}}, t_{\text{trunc}}\right] = \sum_{t=1}^{t_{\text{trunc}}} \left[1 - \Pr(T_n^{\text{upper}} \leq t - 1)\right].$$

REFERENCES

[1] H. J. Kimble, "The quantum internet," *Nature*, vol. 453, no. 7198, p. 1023, 2008.

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

[3] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, vol. 175, 1984.

[4] A. K. Ekert, "Quantum cryptography based on Bell's theorem," *Phys. Rev. Lett.*, vol. 67, pp. 661–663, Aug 1991. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.67.661

[5] R. Jozsa, D. S. Abrams, J. P. Dowling, and C. P. Williams, "Quantum clock synchronization based on shared prior entanglement," *Phys. Rev. Lett.*, vol. 85, pp. 2010–2013, Aug 2000. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.85.2010

[6] W. Ge, K. Jacobs, Z. Eldredge, A. V. Gorshkov, and M. Foss-Feig, "Distributed quantum metrology with linear networks and separable inputs," *Phys. Rev. Lett.*, vol. 121, p. 043604, Jul 2018. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.121.043604

[7] Q. Zhuang, Z. Zhang, and J. H. Shapiro, "Distributed quantum sensing using continuous-variable multipartite entanglement," *Phys. Rev. A*, vol. 97, p. 032329, Mar 2018. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.97.032329

[8] A. M. Childs, "Secure assisted quantum computation," *Quantum Info. Comput.*, vol. 5, no. 6, pp. 456–466, Sep. 2005. [Online]. Available: http://dl.acm.org/citation.cfm?id=2011670.2011674

[9] A. Kellerer, "Quantum telescopes," *Astronomy & Geophysics*, vol. 55, no. 3, pp. 3.28–3.32, 06 2014. [Online]. Available: https://doi.org/10.1093/astrogeo/atu126

[10] W. K. Wootters and W. H. Zurek, "A single quantum cannot be cloned," *Nature*, vol. 299, no. 5886, p. 802, 1982.

[11] 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.*, vol. 81, pp. 5932–5935, Dec 1998. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.81.5932

[12] K. Azuma, K. Tamaki, and H.-K. Lo, "All-photonic quantum repeaters," *Nature communications*, vol. 6, p. 6787, 2015.

[13] N. K. Bernardes, L. Praxmeyer, and P. van Loock, "Rate analysis for a hybrid quantum repeater," *Phys. Rev. A*, vol. 83, p. 012323, Jan 2011. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.83.012323

[14] J. Borregaard, P. Kómár, E. M. Kessler, M. D. Lukin, and A. S. Sørensen, "Long-distance entanglement distribution using individual atoms in optical cavities," *Physical Review A*, vol. 92, no. 1, p. 012307, 2015.

[15] J. Borregaard, M. Zugenmaier, J. M. Petersen, H. Shen, G. Vasilakis, K. Jensen, E. S. Polzik, and A. S. Sørensen, "Scalable photonic network architecture based on motional averaging in room temperature gas," *Nature communications*, vol. 7, p. 11356, 2016.

[16] J. Borregaard, A. S. Sørensen, and P. Lodahl, "Quantum networks with deterministic spin–photon interfaces," *Advanced Quantum Technologies*, p. 1800091, 2019.

[17] J. B. Brask and A. S. Sørensen, "Memory imperfections in atomic-ensemble-based quantum repeaters," *Phys. Rev. A*, vol. 78, p. 012350, Jul 2008. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.78.012350

[18] D. Buterakos, E. Barnes, and S. E. Economou, "Deterministic generation of all-photonic quantum repeaters from solid-state emitters," *Physical Review X*, vol. 7, no. 4, p. 041023, 2017.

[19] O. A. Collins, S. D. Jenkins, A. Kuzmich, and T. A. B. Kennedy, "Multiplexed memory-insensitive quantum repeaters," *Phys. Rev. Lett.*, vol. 98, p. 060502, Feb 2007. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.98.060502

[20] L.-M. Duan, M. D. Lukin, J. I. Cirac, and P. Zoller, "Long-distance quantum communication with atomic ensembles and linear optics," *Nature*, vol. 414, pp. 413 EP –, Nov 2001, article. [Online]. Available: https://doi.org/10.1038/35106500

[21] A. G. Fowler, D. S. Wang, C. D. Hill, T. D. Ladd, R. Van Meter, and L. C. Hollenberg, "Surface code quantum communication," *Physical review letters*, vol. 104, no. 18, p. 180503, 2010.

[22] S. Guha, H. Krovi, C. A. Fuchs, Z. Dutton, J. A. Slater, C. Simon, and W. Tittel, "Rate-loss analysis of an efficient quantum repeater architecture," *Phys. Rev. A*, vol. 92, p. 022357, Aug 2015. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.92.022357

[23] L. Jiang, J. M. Taylor, and M. D. Lukin, "Fast and robust approach to long-distance quantum communication with atomic ensembles," *Phys. Rev. A*, vol. 76, p. 012301, Jul 2007. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.76.012301

[24] L. Jiang, J. M. Taylor, K. Nemoto, W. J. Munro, R. Van Meter, and M. D. Lukin, "Quantum repeater with encoding," *Physical Review A*, vol. 79, no. 3, p. 032325, 2009.

[25] W. Munro, K. Harrison, A. Stephens, S. Devitt, and K. Nemoto, "From quantum multiplexing to high-performance quantum networking," *Nature Photonics*, vol. 4, no. 11, p. 792, 2010.

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

[27] S. Muralidharan, L. Li, J. Kim, N. Lütkenhaus, M. D. Lukin, and L. Jiang, "Optimal architectures for long distance quantum communication," *Scientific reports*, vol. 6, p. 20463, 2016.

[28] S. Muralidharan, C.-L. Zou, L. Li, J. Wen, and L. Jiang, "Overcoming erasure errors with multilevel systems," *New Journal of Physics*, vol. 19, no. 1, p. 013026, 2017.

[29] M. Pant, H. Krovi, D. Englund, and S. Guha, "Rate-distance tradeoff and resource costs for all-optical quantum repeaters," *Physical Review A*, vol. 95, no. 1, p. 012304, 2017.

[30] S. Santra, L. Jiang, and V. S. Malinovsky, "Quantum repeater architecture with hierarchically optimized memory buffer times," *Quantum Science and Technology*, vol. 4, no. 2, p. 025010, mar 2019. [Online]. Available: https://doi.org/10.1088%2F2058-9565%2Fab0bc2

[31] C. Simon, H. de Riedmatten, M. Afzelius, N. Sangouard, H. Zbinden, and N. Gisin, "Quantum repeaters with photon pair sources and multimode memories," *Phys. Rev. Lett.*, vol. 98, p. 190503, May 2007. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.98.190503

[32] R. Van Meter, T. D. Ladd, W. Munro, and K. Nemoto, "System design for a long-line quantum repeater," *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 3, pp. 1002–1013, 2009.

[33] W. J. Munro, K. Azuma, K. Tamaki, and K. Nemoto, "Inside quantum repeaters," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 21, no. 3, pp. 78–90, may 2015. [Online]. Available: https://doi.org/10.1109%2Fjstqe.2015.2392076

[34] W. Dür and H. J. Briegel, "Entanglement purification and quantum error correction," *Reports on Progress in Physics*, vol. 70, no. 8, p. 1381, 2007. [Online]. Available: http://stacks.iop.org/0034-4885/70/i=8/a=R03

[35] N. Gisin, "Hidden quantum nonlocality revealed by local filters," *Physics Letters A*, vol. 210, no. 3, pp. 151–156, 1996.

[36] C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, and W. K. Wootters, "Purification of noisy entanglement and faithful teleportation via noisy channels," *Phys. Rev. Lett.*, vol. 76, pp. 722–725, Jan 1996. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.76.722

[37] D. Deutsch, A. Ekert, R. Jozsa, C. Macchiavello, S. Popescu, and A. Sanpera, "Quantum privacy amplification and the security of quantum cryptography over noisy channels," *Phys. Rev. Lett.*, vol. 77, pp. 2818–2821, Sep 1996. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.77.2818

[38] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, "Mixed-state entanglement and quantum error correction," *Physical Review A*, vol. 54, no. 5, p. 3824, 1996.

[39] E. T. Campbell and S. C. Benjamin, "Measurement-based entanglement under conditions of extreme photon loss," *Physical review letters*, vol. 101, no. 13, p. 130502, 2008.

[40] F. Rozpędek, T. Schiet, D. Elkouss, A. C. Doherty, S. Wehner *et al.*, "Optimizing practical entanglement distillation," *Physical Review A*, vol. 97, no. 6, p. 062333, 2018.

[41] K. Fang, X. Wang, M. Tomamichel, and R. Duan, "Non-asymptotic entanglement distillation," *IEEE Transactions on Information Theory*, 2019.

[42] S. Abruzzo, S. Bratzik, N. K. Bernardes, H. Kampermann, P. van Loock, and D. Bruß, "Quantum repeaters and quantum key distribution: Analysis of secret-key rates," *Phys. Rev. A*, vol. 87, p. 052315, May 2013. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.87.052315

[43] S. Khatri, C. T. Matyas, A. U. Siddiqui, and J. P. Dowling, "Practical figures of merit and thresholds for entanglement distribution in large-scale quantum repeater networks," *arXiv:1905.06881*, 2019. [Online]. Available: http://www.arxiv.org/abs/1905.06881

[44] N. Sangouard, C. Simon, H. de Riedmatten, and N. Gisin, "Quantum repeaters based on atomic ensembles and linear optics," *Rev. Mod. Phys.*, vol. 83, pp. 33–80, Mar 2011. [Online]. Available: https://link.aps.org/doi/10.1103/RevModPhys.83.33

[45] E. Shchukin, F. Schmidt, and P. van Loock, "On the waiting time in quantum repeaters with probabilistic entanglement swapping," *arXiv:1710.06214v4, version 4*, 2017.

[46] S. E. Vinay and P. Kok, "Statistical analysis of quantum-entangled-network generation," *Phys. Rev. A*, vol. 99, p. 042313, Apr 2019. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.99.042313

[47] F. Rozpędek, 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 Science and Technology*, vol. 3, no. 3, p. 034002, 2018.

[48] F. Rozpędek, R. Yehia, K. Goodenough, M. Ruf, P. C. Humphreys, R. Hanson, S. Wehner, and D. Elkouss, "Near-term quantum repeater experiments with nv centers: overcoming the limitations of direct transmission," *Physical Review A*, vol. 99, no. 5, p. 052330, 2019.

[49] J. Cirac, A. Ekert, S. Huelga, and C. Macchiavello, "Distributed quantum computation over noisy channels," *Physical Review A*, vol. 59, no. 6, p. 4249, 1999.

[50] 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.*, vol. 70, pp. 1895–1899, Mar 1993. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.70.1895

[51] R. F. Werner, "Quantum states with einstein-podolsky-rosen correlations admitting a hidden-variable model," *Phys. Rev. A*, vol. 40, pp. 4277–4281, Oct 1989. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.40.4277

[52] W. Feller, *An introduction to probability theory and its applications*. New York: John Wiley and Sons, 1957, vol. I.

[53] A. Dvoretzky, J. Kiefer, and J. Wolfowitz, "Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator," *Ann. Math. Statist.*, vol. 27, no. 3, pp. 642–669, 09 1956. [Online]. Available: https://doi.org/10.1214/aoms/1177728174

[54] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of Computation*, vol. 19, no. 90, p. 297, apr 1965. [Online]. Available: https://doi.org/10.2307%2F2003354

[55] D. M. Greenberger, M. A. Horne, and A. Zeilinger, "Going beyond Bell's theorem," *arXiv:0712.0921*, 2007.

[56] M. A. Nielsen, "Optical quantum computation using cluster states," *Phys. Rev. Lett.*, vol. 93, p. 040503, Jul 2004. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.93.040503

[57] "Waiting time and fidelity in quantum repeater chains," https://github.com/sebastiaanbrand/waiting-time-quantum-repeater-chains, 2019.

[58] M. Charras-Garrido and P. Lezaud, "Extreme value analysis: an introduction," *Journal de la Société Française de Statistique*, vol. 154, no. 2, pp. pp–66, 2013.

[59] A. Wald, *Sequential Analysis*. Dover, New York: Courier Corporation, 1947.

Sebastiaan Brand received a B.S. degree in Computer Science and Engineering from the Technical University of Delft, the Netherlands, in 2016 and an M.S degree in Computer Science from Leiden University, the Netherlands, in 2019. Currently he works as a research and education assistant at Leiden University, working on the simulation of quantum circuits.

Tim Coopmans received B.S. degrees in Mathematics & Physics from Utrecht University, the Netherlands, in 2014 and an M.S. degree in Logic from the University of Amsterdam, the Netherlands, in 2017. As member of QuTech, he is currently a PhD student at the Technical University of Delft, the Netherlands, working on simulations of quantum networks.

David Elkouss (S'09-M'12) received M.S. degrees from the Polytechnic University of Madrid, Spain, and from ParisTech, France, in 2006 and a PhD degree from the Polytechnic University of Madrid in 2012. He was a postdoctoral researcher at the Complutense University of Madrid and at the Technical University of Delft, the Netherlands. Since 2017 he is an assistant professor at the Technical University of Delft and a member of QuTech.