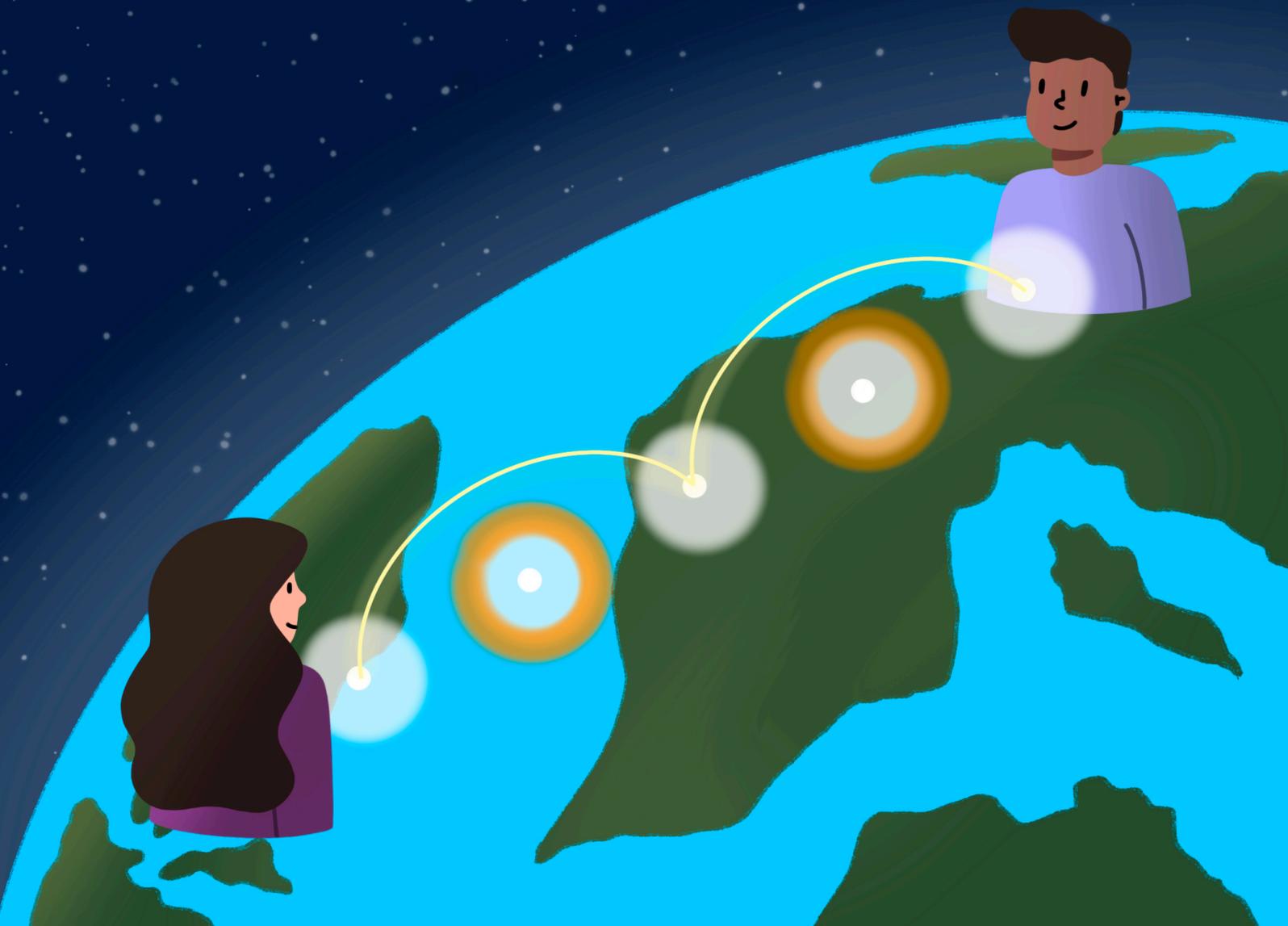


Scalable and Implementable Entanglement Distribution Policies in Homogeneous Repeater Chains with Cutoffs

Héctor Calero Mas



QuTech



Scalable and Implementable Entanglement Distribution Policies in Homogeneous Repeater Chains with Cutoffs

by

Héctor Calero Mas

Student number: 5844460
Project Duration: September 1, 2023 - April 14, 2024
Thesis Committee: Dr. G. Vardoyan, TU Delft, supervisor
Dr. M. Blaauboer, TU Delft
Dr. S. Feld, TU Delft
Daily supervisor: PhD candidate Á.G. Iñesta, TU Delft

Cover: Illustration by Jorge Sarrias



QuTech

Abstract

Entanglement can be used as a resource to support a wide range of quantum applications. However, the scarcity of efficient entanglement distribution protocols poses a significant challenge for the deployment of large-scale quantum networks. Losses in the media prevent the direct transmission of quantum states over large distances, but the use of quantum repeaters presents a possible alternative for long-distance quantum communication. Here, we focus on homogeneous quantum repeater chains and provide some guidelines based on heuristic methods that allow the design of entanglement swapping policies. For instance, delaying simultaneous swaps on adjacent nodes can reduce the probability of losing entanglement. Whereas previous work mainly focused on chains with few nodes only, we present three different policies that are easy to implement and scalable to longer chains. We evaluate these policies using Monte Carlo simulations, comparing their performance to the well-known swap-asap policy. When classical communication time is neglected, our policies provide lower delivery time than swap-asap for probabilistic swaps and large entanglement generation probability. When classical communication time is large, only one of our policies is in most cases faster than swap-asap for both probabilistic and deterministic swaps.

*Physics is very muddled again at the moment;
it is much too hard for me anyway,
and I wish I were a movie comedian or something like that
and had never heard anything about physics..*

—Wolfgang Pauli

Acknowledgments

Firstly, I would like to thank my supervisor Dr. Gayane Vardoyan, and my daily supervisor, PhD candidate Álvaro Gómez Iñesta, for allowing me to be part of this project and providing me with invaluable guidance throughout the past seven months. I deeply appreciate your involvement and dedication, which made this work possible. Álvaro, I am very grateful for your wise advice, which has saved me a lot of time and effort. I have also learned a great deal from your feedback and mathematical rigor. Many thanks to Dr. Blaauboer and Dr. Feld as well, for kindly accepting the invitation to join my thesis committee.

I would also like to thank all the “qoffee” people for welcoming me, accompanying me during lunches and breaks, and making my stay in QuTech much more enjoyable. Special thanks to Jeroen for our insightful discussions and help in the project. Of course, I want to give a big shout-out to all my friends and colleagues in the Master room: Alex, Caroline, Cees, Constantijn, Dani, Elvis, Joan, Kees, Kenji, Leo, Marit, Pepijn, Stijn, Sugeiva, Timo, Tobi and Victor. B104 has been the source of countless memes, board games, group therapy sessions, discussions, and sometimes even work. I am very grateful for the time I have spent with all of you. To my IITD boys Praveen, Kishore, and Sibasish: thank you for hosting me, and joining me in my existential crises and workouts.

Quiero agradecer especialmente a mi familia el haberme permitido siempre estudiar donde quisiera. Gracias también por darme ánimos y soportar mis constantes quejas sobre el clima. A mis amigos de España, gracias por existir. Deambular por el mundo es mucho más fácil sabiendo que estaréis cuando regrese.

*Héctor Calero Mas
Delft, April 2024*

Contents

1	Introduction	1
2	Background	4
2.1	Quantum networks	5
2.1.1	Entanglement generation	5
2.1.2	Entanglement swapping	6
2.2	Quantum repeater chain model	8
2.3	Measures of performance	10
3	Heuristic strategy	12
3.1	Basic principles	12
3.1.1	Scalability	13
3.1.2	Robustness	13
3.1.3	Determinism	13
3.1.4	Monotonicity	14
3.1.5	Symmetry	14
3.2	Communication time	14
3.2.1	The effect of communication	15
3.2.2	Local, global, and minimum-knowledge policies	17
3.3	The swap-asap policy	19
4	Policy Candidates	26
4.1	The nonadjacent policy	26
4.1.1	Connected component	27
4.1.2	Number of allowed swaps	28
4.1.3	Distribution of waiting nodes	28
4.2	The nonadjacent-middle-last policy	30
4.3	The nested policy	30
4.3.1	Nested algorithm	32
5	Results	36
5.1	One-time entanglement distribution	36
5.1.1	Deterministic swaps	36
5.1.2	Probabilistic swaps	38
5.2	Continuous entanglement distribution	38
5.2.1	Deterministic swaps	39
5.2.2	Probabilistic Swaps	40
5.3	Non-negligible communication time	41
5.4	Remarks	44
6	Conclusion	46

Bibliography	47
A Link post-swapping age	51
B End-to-end delivery time of middle-last	52
B.1 Odd n	52
B.2 Even n	53
C Nested vector for ten nodes	55
D Simulation methods	57

Introduction

Quantum networks are distributed and interconnected systems that enable the transmission, processing, and storage of quantum information. By utilizing the principles of quantum mechanics, they offer the possibility of establishing secure communication, performing quantum computing in the cloud, as well as building sensor networks or conducting fundamental scientific tests [1]. For most of these applications, bipartite [2] or multipartite [3] entanglement is required, so great efforts have been put in the past few decades to devise reliable methods to distribute entanglement in quantum networks. Even though a fully functional global-scale quantum network is still far from becoming a reality, recent experimental demonstrations of long-distance entanglement distribution and entanglement purification [4, 5, 6] are paving the way towards building scalable quantum networks and ultimately the so-called quantum internet [7].

The main challenge of long-distance quantum communication is that losses scale exponentially with distance. This is critical for quantum communication protocols, where information is often encoded into single photons or entangled photon pairs. Moreover, the no-cloning theorem [8] prevents creating copies of an arbitrary quantum state, so classical amplification techniques are generally not suitable for quantum applications. Given the high scattering and absorption by particles in the atmosphere, sending photons through free space is not practical for large distances, so reducing the attenuation of the physical medium has been one of the priorities in the telecommunications industry. In the last five decades, optical fibers have attained a drastic reduction of their losses, practically reaching their fundamental limit [9]. However, even for the current lowest achieved losses (0.142 dB/km at 1560 nm [10]), transmission of single photons is not feasible for long distances. For instance, the time required for transmitting one entangled photon pair over 2000 km, with a 1 GHz source exceeds the age of the universe [11].

Two main approaches have been proposed to enable the transmission of quantum states over long distances. The first, involving the use of satellite platforms and space-based links, has motivated highly sophisticated experiments in which teleportation of quantum states over distances up to 1400 km was reported [12]. The second approach is to effectively break down the transmission distance into shorter, more manageable segments through the use

of quantum repeaters [13]. Instead of directly attempting entanglement generation between two distant parties, shorter-distance entangled pairs are distributed between neighbouring nodes. Then, entanglement can be extended to longer distances employing swaps [14], which are a set of local operations that allow the transfer of entanglement between distant parties without direct interaction between them. A mathematical description of this phenomenon is described in Section 2.1. In practice, entanglement distribution through quantum repeater chains is only possible if the rate of entanglement generation between nodes exceeds the states' decoherence rates. Surpassing this threshold [15] enabled the first realization of a three-node quantum network [16], showing promising perspectives for future repeater-based quantum communications. Whereas satellite communication experiments have successfully demonstrated long-distance entanglement distribution, quantum repeaters seem a better alternative for achieving high entanglement distribution rates in metropolitan-scale (1 km– 100 km) quantum networks. Future global-scale quantum networks may also combine these two elements, for example, by employing satellites as quantum repeaters [17] or by connecting satellites through ground repeater stations [11].

In this thesis, we are mainly interested in studying bipartite entanglement between qubits. An equivalent term to describe the existence of entanglement between two particles is to say that they share an entangled link. We define an *entanglement distribution policy* π as the set of rules that determines which actions should be performed by the nodes given the existing entangled links in the chain. An example is the swap-asap policy, in which swaps are attempted *as soon as* a node shares two entangled links. Our goal is to shed light on how to design these entanglement distribution policies for an arbitrary number of nodes using heuristic methods. Depending on the particular policy, the classical communication costs may vary. Whereas in local-knowledge policies nodes only communicate with their nearest neighbours, in global-knowledge policies all-to-all communication is necessary. In this work, we focus on chains of equidistant and identical nodes, which in our model are parametrized by the number of nodes n , the probability of generating entanglement between neighbouring nodes p , and the probability of successfully swapping two links into a longer one p_s . For simplicity, we assume that each repeater holds two memories, one to be used towards each side of the chain. A representation of a repeater chain with five nodes is shown in Fig. 1.1. We benchmark our policies using two performance measures: the expected *end-to-end delivery time*, and the *end-to-end fidelity*. The first accounts for the average time it takes to generate entanglement between the two end nodes, and the second is the average fidelity of the end links. It is expected that both metrics for all policies should be monotonic on the parameters p and p_s , which means that if the hardware specifications of the devices are improved, the performance of the policy should increase.

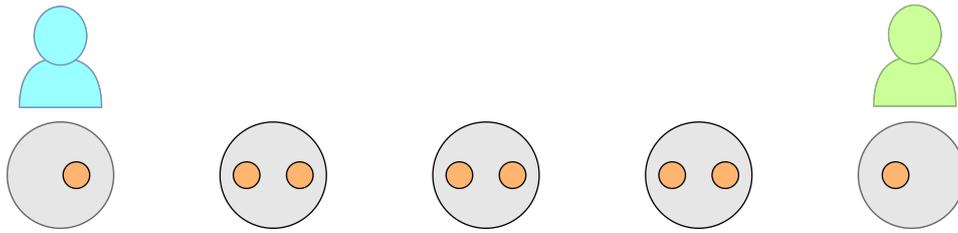


Figure 1.1: Illustration of a quantum repeater chain. In our model, end nodes store a single memory qubit (represented by orange circles), while intermediate nodes store two.

Our main contributions are the following:

- We consider classical communication between nodes and show that fully local-knowledge policies do not exist for chains with five or more nodes.
- We study the monotonicity of the end-to-end delivery time and end-to-end fidelity with respect to the entanglement generation probability p and the amount of classical information required in the swap-asap policy. From this, we extract heuristics for designing new policies.
- We propose three entanglement distribution policies, and benchmark their expected entanglement delivery time and end-to-end fidelity with respect to the swap-asap policy.

Our main findings are the following:

- The proposed policies reduce the end-to-end delivery time for probabilistic swaps and large p , up to a factor of ~ 6 (for $n = 10$, $p = 1$ and $p_s = 0.5$) compared to swap-asap, when classical communication times are neglected. For very large classical communication time, our nested policy is faster than swap-asap for $p_s = 0.5$, and for $p_s = 1$ when p is small. Improvements over swap-asap in the end-to-end fidelity are not observed.
- The swap-asap policy requires global-knowledge, and its end-to-end delivery time exhibits a non-monotonic behaviour for some range of values of p when swaps are probabilistic.
- Avoiding swapping *many* adjacent links simultaneously can reduce the expected delivery time when swaps are probabilistic.
- Enforcing the last swap to be performed at the center of the chain can reduce the classical communication time.

This thesis is structured in six chapters. In Chapter 2, we introduce the theoretical basis of quantum networks and present our quantum repeater chain model. We also present the measures of performance used to benchmark our policies. In Chapter 3, we define the basic principles on which we base our search for entanglement distribution policies and extract heuristics from the study of the swap-asap policy. In Chapter 4, we propose our three candidate policies and show our results in Chapter 5. Finally, in Chapter 6, we conclude the dissertation by summarizing our main findings and suggesting future lines of research.

2

Background

In this thesis, we approach quantum repeater chains from a theoretical point of view, providing a general framework applicable to different platforms but without delving into the particular properties of each one. Nevertheless, to design policies that can be implemented in a real setup, certain fundamental aspects of the hardware and protocols must be considered. In Section 2.1, we distinguish between *communication* and *flying* qubits, and we depict the basic characteristics and limitations of entanglement generation and entanglement swapping protocols. Then, we discuss how depolarizing noise affects the fidelity of quantum states and introduce our quantum repeater chain model in Section 2.2. Finally, in Section 2.3, we introduce the four different performance measures which will be used to compare our entanglement distribution policies, distinguishing between “one-time” and “continuous” end-to-end entanglement delivery.

2.1. Quantum networks

The building blocks of quantum networks are quantum nodes, which are physical devices capable of generating, manipulating, storing, and transmitting quantum states. Nodes typically have two types of qubits: communication qubits which can be used to generate entanglement with their neighbours and memory qubits that store quantum information. The physical platforms for each type of qubit can vary, even within the same node. Next, we present the main protocols for entanglement generation and entanglement swapping. These are the key functionalities that allow long-distance entanglement distribution in quantum networks.

2.1.1. Entanglement generation

Entanglement generation can be defined as the process of creating entangled states between quantum systems. If two or more particles are entangled, the quantum state of any of the particles cannot be described independently from the state of the others, even if they are spatially separated [18]. To become entangled, particles need to physically interact, so entanglement generation between distant parties cannot be achieved only by local operations.

For quantum applications, it is necessary to be able to perform logical quantum operations across the network. This means that the physical qubits must be ideally easily addressable and robust to decoherence. In most physical implementations, it is not feasible to use the same qubits to perform quantum gates and to transmit quantum information. Instead, there are generally two types of qubits involved: local operations are generally performed on the so-called “stationary” qubits, while information is sent through “flying” qubits. In order to generate entanglement between two distant locations, nodes should be able to interconvert between stationary and flying qubits [19]. Photons are great entanglement mediators (flying qubits), as they can transmit quantum states at high velocities. Furthermore, they do not require low temperatures to preserve their quantum properties, so they can be transmitted through the existing low-loss optical fibers. However, qubits with long coherence time must have a weak interaction with the environment and consequently, it is difficult to transfer their quantum state into photons. For this reason, quantum networks commonly employ two different classes of stationary qubits: communication qubits, which must have an efficient photonic interface to transfer their quantum states into photons, and memory qubits, which are more resilient against decoherence.

The interaction between stationary and flying qubits has been extensively studied, and several experiments have demonstrated remote entanglement generation using different platforms such as trapped ions [20], color centers in diamond [21, 15], quantum dots [22, 23] and neutral atoms [24, 25]. The implementation of entanglement generation is highly dependent on the specific platform, but it is still possible to find common features between different protocols [26]. In the context of quantum communications, it is particularly

relevant to consider *heralded* entanglement generation protocols. That is, entanglement generation is conditional on the detection of a specific event. Consequently, given a particular measurement outcome, it is possible to determine if two qubits are entangled or not. Note that this does not imply that entanglement generation is always deterministic, since the events that herald entanglement can occur with a certain probability. This probability is limited by the specific entanglement generation protocol and photon losses. Essentially, the most common bipartite entanglement generation protocols can be summarized in three steps:

- Within each node, stationary and flying qubits interact selectively (through an optical transition), which generates entanglement between them. This interaction can be interpreted as a C-NOT gate with control on the stationary qubits and target on the flying qubits.
- Nodes send their flying qubits (photons) to a measurement device (typically a midpoint station). For certain measurement outcomes, the joint state of the stationary qubits will be maximally entangled. In the rest of the cases, the state of the stationary qubits will be fully separable.
- Nodes receive a flagged signal communicating the measurement outcome.

2.1.2. Entanglement swapping

Short-distance entangled links can be fused into longer-distance ones via entanglement swapping. This operation has some rather counterintuitive implications, as particles that may have never directly interacted can become entangled [14]. Let us consider two parties, Alice and Bob, that want to share an entangled state. Each of them shares a maximally entangled link with one of the qubits of Charlie, who is in an intermediate station. Figure 2.1 shows the quantum circuit representation of one of the possible implementations of this scheme. We will assume that the joint states between Alice and Charlie and Bob and Charlie are the following:

$$|\psi\rangle_{AC_1} = |\Phi^+\rangle_{AC_1} = \frac{1}{\sqrt{2}} (|0_A 0_{C_1}\rangle + |1_A 1_{C_1}\rangle). \quad (2.1)$$

$$|\psi\rangle_{C_2B} = |\Phi^+\rangle_{C_2B} = \frac{1}{\sqrt{2}} (|0_{C_2} 0_B\rangle + |1_{C_2} 1_B\rangle). \quad (2.2)$$

Where the subscripts A , B , and C_i represent Alice, Bob, and each of Charlie's qubits, respectively. The full state of the system is given by:

$$\begin{aligned} |\psi\rangle_{AC_1C_2B} &= |\Phi^+\rangle_{AC_1} \otimes |\Phi^+\rangle_{C_2B} = \\ &= \frac{1}{2} (|0_A 0_{C_1} 0_{C_2} 0_B\rangle + |0_A 0_{C_1} 1_{C_2} 1_B\rangle + |1_A 1_{C_1} 0_{C_2} 0_B\rangle + |1_A 1_{C_1} 1_{C_2} 1_B\rangle). \end{aligned} \quad (2.3)$$

In this situation, Charlie can perform a Bell state measurement, which is an operation that reveals in which of the four Bell states two qubits are [27]. A possible implementation consists of applying a C-NOT gate between the two

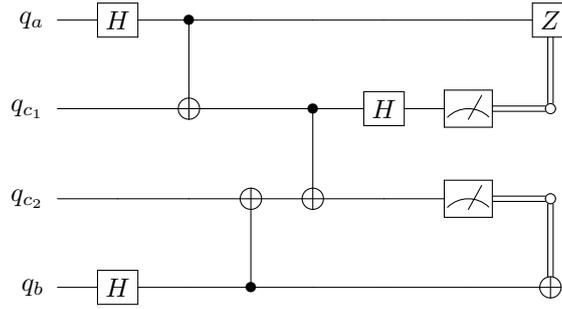


Figure 2.1: Entanglement swapping circuit diagram. Entanglement is first generated between Alice and Charlie and between Bob and Charlie. Then, Charlie performs a Bell state measurement on his qubits and classically communicates the measurement outcome to Alice and Bob, who perform local operations on their qubits based on Charlie’s measurement outcomes. At the end of the protocol, Alice and Bob share an entangled pair despite having never directly interacted.

qubits (it is irrelevant which of the qubits is the control and which the target) followed by a Hadamard gate on the control qubit and the measurement of both qubits. Before Charlie’s qubits are measured, the state of the system is the following:

$$|\psi\rangle_{AC_1C_2B} = \frac{1}{2\sqrt{2}} (|0_A0_{C_1}0_{C_2}0_B\rangle + |0_A1_{C_1}0_{C_2}0_B\rangle + |0_A0_{C_1}1_{C_2}1_B\rangle + |0_A1_{C_1}1_{C_2}1_B\rangle \\ + |1_A0_{C_1}1_{C_2}0_B\rangle - |1_A1_{C_1}0_{C_2}0_B\rangle + |1_A0_{C_1}0_{C_2}1_B\rangle - |1_A1_{C_1}1_{C_2}1_B\rangle). \quad (2.4)$$

The measurement of Charlie’s qubits projects Alice and Bob’s qubits state into one of the four possible Bell states, depending on the measurement outcome:

$$|\Phi^\pm\rangle = \frac{1}{\sqrt{2}} (|00\rangle \pm |11\rangle), \quad |\Psi^\pm\rangle = \frac{1}{\sqrt{2}} (|01\rangle \pm |10\rangle). \quad (2.5)$$

Then, Charlie communicates classically the outcome of his measurements so Alice and Bob can perform local operations to ensure that they share the desired Bell state. At the end of the protocol, Alice and Bob share the maximally entangled state $|\Phi^+\rangle$:

$$|\psi\rangle_{AB} = |\Phi^+\rangle_{AB} = \frac{1}{\sqrt{2}} (|0_A0_B\rangle + |1_A1_B\rangle). \quad (2.6)$$

As we had anticipated, Alice and Bob’s qubits are now entangled despite never having directly interacted. In the described entanglement swapping protocol, it is assumed that the same qubits are used for generating entanglement and for performing the entanglement swap. In a more realistic scenario, entanglement would be first realized between communication qubits, and then swapped to memories. This way, quantum states can be stored for longer periods without decohering. If all quantum gates are ideal, Alice and Bob always share a maximally entangled pair at the end of the protocol. This is achieved

at the cost of an additional gate overhead, which translates into longer operation times and more complex experimental setups. On the other hand, linear optic setups provide a simpler approach and achieve higher speeds, but are typically limited by a 50% success probability [28]. This limit can also be exceeded by introducing ancillary qubits [29]; if N ancilla qubits are used, the maximum success probability for linear optics setups scales as $1 - \frac{1}{N+2}$ [30].

2.2. Quantum repeater chain model

In this section, we present our quantum repeater chain model, which is adapted from [31]. The first physical property that defines the nodes is the entanglement generation probability p , defined as the probability of creating an entangled link between two neighbouring nodes. Newly generated links will be called elementary links. State-of-the-art quantum devices are far from being ideal due to several factors such as the photonic interface efficiency of communication qubits and photon losses. This causes extremely low values for the probability of a successful entanglement generation attempt (e.g., 10^{-5} for trapped ions over a distance of 520 m [32]). However, it is expected that improvements in the hardware, and exploration of new qubit platforms, jointly with the use of multiplexed memories can significantly increase this probability in the future. In addition, if multiple entanglement generation attempts per time step are considered, the entanglement generation probability can remarkably grow, at the expense of a longer time step duration. In this project, we will consider values of p ranging from 0.1 to 1.

The second parameter of interest is the entanglement swapping probability p_s , which is the probability that a swap succeeds, generating a longer-distance entangled link from two shorter-distance ones. After a swap, both shorter-distance links are consumed, irrespective of the success or failure of the operation. We will consider either $p_s = 1$ or 0.5, which is in agreement with current experimental demonstrations.

Memory qubits should be stable to achieve large coherence times. Although techniques such as dynamical decoupling can extend the lifetime of qubits by several orders of magnitude [33, 34], current quantum memories are still noisy. We consider a worst-case noise model [35] and describe the quantum states of memory qubits as Werner states [36], which are subject to a depolarizing channel at every instant and whose fidelity with respect to a maximally entangled state (e.g., $|\Phi^+\rangle$) evolves as follows [31]:

$$F(t) = \frac{1}{4} + \left(F(t - \Delta t) - \frac{1}{4} \right) e^{-\frac{\Delta t}{\tau}}. \quad (2.7)$$

Here Δt represents an arbitrary time interval and τ is the decay rate caused by noisy memories. We would ideally want to guarantee that the fidelity of the entangled links shared between the two end nodes in the repeater chain, i.e., the *end-to-end fidelity*, satisfies the requirements for a particular application: $F_{e2e} \geq F_{\min}$. One way of ensuring this is by applying hard cutoffs. That is, all

links used to generate an end-to-end link must be generated within a time window of fixed size. In practice, this is done by tracking the ages of links (time passed after they have been generated). Then, entangled links are deterministically removed if they exceed a certain age t_{cut} . In [31], authors take the post-swapping age of a link to be the age of the oldest link involved in the operation. They show the end-to-end fidelity always exceeds the threshold F_{min} for cutoffs that satisfy

$$t_{\text{cut}} \leq -\tau \ln \left(\frac{3}{4F_0 - 1} \left(\frac{4F_{\text{min}} - 1}{3} \right)^{\frac{1}{n-1}} \right), \quad (2.8)$$

where n is the number of nodes in the chain, F_0 is the fidelity of elementary links and τ is the decay rate. The age of a newly generated link is set to zero. At the end of each time step, the ages of all existing links are increased by one. If a link is the result of one or more swaps, its age will be the age of the oldest link that was involved in the swap. Note that this rule does not accurately describe the decrease in fidelity caused by the swaps, but it is a useful approach to define the cutoff time. In Appendix A, a more precise rule for the post-swapping ages will be introduced.

Next, we list all the additional assumptions of our model:

- All nodes in the chain have identical physical properties. This means that every node has the same entanglement swapping probability p_s and that all pairs of neighbouring nodes generate elementary links with equal probability p . Moreover, all states decohere at the same rate τ .
- All nodes are arranged in a one-dimensional array, separated by a constant distance of l meters.
- Each node possesses one communication qubit and one memory qubit for each side of the chain. This assumption is consistent with the expectations for the early-stage quantum networks [16, 37].
- All nodes are capable of: (i) generating entanglement with their neighbours with probability p ; (ii) successfully performing entanglement swapping with probability p_s ; (iii) communicating and processing classical information; (iv) deterministically discarding entangled links older than t_{cut} .

Time in our model is discretized into time steps. The length of a time step depends on the speed of the operations and the time needed to communicate classical information. An in-depth discussion about the time step length can be found in Section 3.2.1. Within each time step, nodes sequentially perform the following operations (Fig 2.2):

1. Attempt entanglement generation with neighbouring nodes. Neighbouring nodes can only attempt entanglement generation if they have free qubits. A heralding signal is classically transmitted to communicate whether entanglement has been generated.

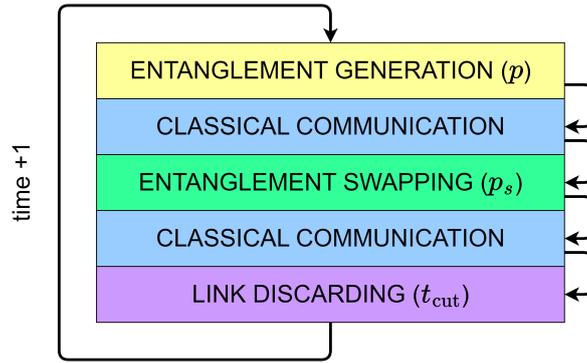


Figure 2.2: Diagram of the sequence of operations within each time step. At the beginning of each time step, neighbouring nodes attempt entanglement with success probability p . Next, classical communication can be transferred between the nodes to communicate the (partial) state of the system. Then, nodes can perform entanglement swaps with probability p_s , after which classical communication is again transmitted. All links with an age larger than t_{cut} are then removed. Finally, at the end of each time step, the ages of all existing links are increased by one. Both classical communication rounds and entanglement swapping are policy-dependent, while the rest of the operations are not.

2. Classical information can be transmitted between nodes, depending on the policy π , to communicate the (partial) state of the system.
3. If both memory qubits of a node hold entangled links, the node can perform a swap, freeing both memory qubits and generating a longer link with probability p_s .
4. The success or failure of the swaps and the post-swapping age of the links are communicated classically between nodes. Which nodes are allowed to communicate depends on the specific policy π .
5. Remove existing links with an age larger than t_{cut} .

Under the conditions described above, every quantum repeater chain in our framework can be characterized by the parameters p , p_s , n and t_{cut} . The latter also depends on the noise of the model, given by F_0 and τ , and the specific threshold fidelity for our application F_{min} .

2.3. Measures of performance

In classical communication systems, there exist some extensively studied performance metrics used to assess the efficiency of communication, such as the bit error rate (BER) or the signal-to-noise ratio (SNR). However, current general-purpose communication protocols and infrastructure are already capable of significantly mitigating errors caused by factors like noise or interference. At present, the main efforts in general-purpose classical communication are put to increase its transmission rates to satisfy the demands of a growing number of users. In quantum networks, high communication speeds are not only desirable from the users' perspective, but for some quantum applications, a large entanglement distribution rate is compulsory [38]. For this reason, the primary measure of performance that will be studied is the ex-

pected end-to-end delivery time, which determines the average time needed to generate end-to-end entanglement. In this work, we consider both “one-time” and “continuous” entanglement delivery. In the first case, two users attempt to generate a single entangled pair starting from an empty chain and stop after it is generated. In the latter, users demand a new end-to-end entangled link immediately after the previous one has been generated. The “continuous” scenario reflects more faithfully a situation in which two parties must share entanglement for a specific quantum application, where many copies of a quantum state are needed. The “one-time” case provides an estimate of the transient state until the first end-to-end link is generated. It is also more commonly used in literature as it simplifies the description of the system in terms of a Markov decision process and it sets a fixed starting state for every reinforcement learning episode.

Current state-of-the-art quantum networks are at a very primitive stage compared to classical communication networks, as quantum devices are subject to high losses and depolarizing errors, among others. It is then expected that in the near term, the utility of quantum applications will be limited not only by the rate but also by the quality of the generated entanglement. In our repeater chain model, strict cutoffs ensure a minimum fidelity of the end-to-end links. Still, a higher fidelity translates into more reliable and secure communication, so the average fidelity of the end links will be another quantity of interest in our simulations. Similarly to the previous case, we will consider “one-time” and “continuous” entanglement delivery.

Another alternative to compare the performance of different policies could have been to consider a combined measure such as the secret key rate, which implicitly included the entanglement delivery time and fidelity. Nonetheless, we opted to study these two quantities separately, as it enables us to understand how different heuristics individually affect each measure of performance.

Heuristic strategy

In multi-objective optimization of quantum networks, there exist trade-offs between different measures of performance (e.g., end-to-end delivery time and end-to-end fidelity), so policy optimality is often described through “Pareto efficiency” [39]. This refers to the cases in which it is not possible to improve the performance over one metric without negatively affecting the others. Policies that are not on the Pareto front are called suboptimal. A policy can also be optimal or suboptimal with respect to a single metric, depending on whether it provides the maximum achievable performance for that metric or not. The authors of [31] observed that finding optimal policies is an infeasible task when the chain length or the cutoff time becomes very large. This motivates the search for heuristic methods for finding new policies. These heuristics, from Ancient Greek *heuriskō* (to discover), do not guarantee to produce optimal solutions to our problem, but they can be used as a shortcut to reach sufficiently good policies without exploring the whole search space.

In this chapter, we outline our approach towards generating close to optimal entanglement swapping policies. In Section 3.1, the general premises that were used to design candidate policies are listed. Then, a general discussion about communication times is presented in Section 3.2, where we conclude that fully local-knowledge policies cannot exist in our model. Finally, in Section 3.3, we provide an in-depth analysis of the swap-asap policy to extract heuristics that can be used to improve some of its drawbacks. For example, the non-monotonic behaviour of its delivery time with respect to the entanglement generation probability.

3.1. Basic principles

The number of states in our repeater chain model of n nodes and cutoff time t_{cut} grows at least as $\Omega((t_{\text{cut}})^{n-2})$ [31], so the number of possible policies quickly becomes intractable as the length of the repeater chain increases. It is then convenient to narrow down the search to a subset of solutions that contains the potentially more relevant cases. Here we present the main five principles that were used to that end.

3.1.1. Scalability

In the near future, physical imperfections of devices may restrict the size of practical quantum networks to only a few nodes. Nevertheless, the development of entanglement distribution policies for networks with several repeaters will become necessary to build a global quantum internet. The most common current approach for finding entanglement distribution policies is to formulate the problem as a Markov decision process (MDP), in which the future state of the system depends exclusively on the current state and the actions taken by an agent [40]. Previous work has focused on finding exact solutions [41, 31] or using deep reinforcement learning techniques [42, 43, 44] to obtain optimal or close-to-optimal entanglement distribution policies. Such approaches are only feasible for a small number of nodes in the network, due to the exponential growth of the state space. In contrast, we aim to design policies deployable on a repeater chain with an arbitrary number of nodes.

Our approach consists of extracting heuristics from optimal policies calculated for a small number of nodes and extrapolating them to generate policies that are sufficiently good for larger chains. Similarly, analyzing the features of far-from-optimal policies can help us prevent undesired behaviour of the system.

3.1.2. Robustness

Entanglement swapping policies should in principle be tailored for a given set of parameters (p, p_s, n , etc.). However, even small fluctuations in the physical properties of the devices can significantly decrease the performance of the system, as will be discussed in Section 3.3. In order to overcome this, the parameters of the system should be regularly calibrated and policies should be adjusted accordingly. However, these efforts can be relaxed by designing policies that are valid for wide ranges of parameter values, rather than for unique points in the parameter space.

3.1.3. Determinism

In Section 3.2, we will discuss how policies can be classified according to the amount of information accessible to the nodes. For scenarios in which nodes do not have complete knowledge about the state of the system, the problem can be formulated as a partially observable Markov decision process (POMDP) [45]. In such situations, agents typically encounter the exploration-exploitation dilemma, in which choosing the best-known option based on past experiences may not be the optimal strategy. Instead, probabilistic policies, which include assigning probabilities to different actions given the current belief state (the agent's subjective probability distribution over the possible states of the system), are shown to enable better payoffs [46].

Nonetheless, describing the state of the system as a belief state involves keeping track of the probability distribution of possible states and updating the probabilities of the agents' actions consistently, yielding large computation

and memory costs. For this reason, we chose to limit our search to those policies in which actions are taken deterministically.

3.1.4. Monotonicity

The development of quantum repeaters is currently in an early stage. No experiments have shown successful performance of quantum repeaters over large distances to date. On the other hand, recent experimental progress in many of the proposed platforms instills optimism regarding the viability of constructing large-scale quantum networks.

It could be anticipated that by gradually improving the specifications of quantum devices, faster and more efficient communication will be achieved. That is, we expect the performance of our policies to increase monotonically with respect to improvements in the hardware (higher p and p_s). Despite the apparent triviality of this argument, in Section 3.2, we show that the swap-asap policy does not satisfy this condition. We then use this observation to devise better-performing policies.

3.1.5. Symmetry

For any state of the system s , we can find the mirrored version of that state s^\perp by relabeling the nodes in opposite order. In our model, all nodes are considered to be homogeneous and equidistant from each other, so there is no physical difference between a state and its mirrored version. The action of a policy π in a given state s is equivalent to the action of a mirrored version of that policy, π^\perp , in the state s^\perp : $\pi(s) \equiv \pi^\perp(s^\perp)$ (we can arbitrarily label the nodes from right to left instead of from left to right). Since the repeater chain is symmetric, it is expected that policies are invariant with respect to mirror inversion. That is, the actions of a policy π in the states s and s^\perp should be the same if labeled in opposite order.

$$\pi(s) = (\pi(s^\perp))^\perp. \quad (3.1)$$

Despite symmetry not being a necessary condition for optimality, it can effectively reduce the search space.

3.2. Communication time

Classical communication between nodes is often disregarded in performance analysis of entanglement distribution policies [31, 43]. Here we acknowledge that in large-scale quantum networks, the time allocated for classical communication may be non-negligible. We propose a classification of policies based on the maximum distance that classical information needs to travel in each of the two communication rounds for every time step.

3.2.1. The effect of communication

In our model, as discussed in Section 2.2, time is discretized into non-overlapping time steps. During each time step, nodes can sequentially: attempt entanglement generation with their neighbouring nodes, perform entanglement swapping, and discard all links with age $t > t_{\text{cut}}$. The time required for these operations is denoted by T_{gen} , T_{swap} , and T_{cut} , respectively. These values depend on the physical implementation of the quantum repeaters, but they are independent of the chosen policy.

Entanglement swapping and link discarding are local operations that do not depend on the distance between nodes. On the other hand, heralded entanglement generation typically entails generating and distributing entangled particles between the involved nodes. Also, a flagged signal must be received by the nodes to herald if entanglement has been generated or not. Therefore, the time required for this operation depends on the speed at which entangled particles and classical signals are sent (which we assume to be the speed of light in an optical fiber $c \sim 2 \cdot 10^8$ m/s), and on the distance l between nodes. Some time is also required to perform local operations. It is important to note that the entanglement generation time depends on the number of sequential attempts performed at each time step. Fewer attempts can reduce T_{gen} , at the expense of reducing the entanglement generation probability p . Furthermore, there exists a trade-off between number of attempts and fidelity of the new links, as the memory qubits experience decoherence. Overall:

$$T_{\text{gen}} = N_A \left(\frac{l}{c} + C_{\text{gen}} \right), \quad (3.2)$$

where N_A is the number of sequential generation attempts in each time step and C_{gen} is the time required for local operations (e.g., measurements) in each attempt. Here we assume that nodes send their heralding photons to a mid-point station. Therefore, each attempt requires a time of $\frac{l}{2c}$ to send the heralding photon and the same amount to receive the flagged signal back (plus local operations). Note that this analysis can be extended to other protocols by simply introducing a different prefactor in the term $\frac{l}{c}$.

In addition, time must be reserved for the two classical communication rounds in each time step. During the first round, nodes can share the information regarding successfully generated links. In the second round, nodes can communicate the success or failure of their swaps, the age of the links they hold, and signal that end-to-end entanglement has been achieved. Communication time is only determined by the maximum distance that classical information can travel within a time step in each of the communication rounds: d_{cc1}^π (classical communication round one) and d_{cc2}^π (classical communication round two), which are dependent on the policy π :

$$T_{\text{comm}}^\pi = \frac{d_{\text{cc1}}^\pi + d_{\text{cc2}}^\pi}{c}. \quad (3.3)$$

In general, d_{cc1}^π and d_{cc2}^π are proportional to the chain length and can be written as $d_{cc_i}^\pi = C_i^\pi \cdot (n - 1) \cdot l$, where C_i^π is the fraction of the total distance that classical information can travel in the policy π during communication round i . The duration of a time step can then be split into two terms: the operations time $T_{\text{op}} \equiv T_{\text{gen}} + T_{\text{swap}} + T_{\text{cut}}$, and communication time T_{comm}^π . The first term is policy-independent while the second one is not.

$$T_{\text{step}}^\pi = T_{\text{gen}} + T_{\text{swap}} + T_{\text{cut}} + T_{\text{comm}}^\pi = T_{\text{op}} + T_{\text{comm}}^\pi. \quad (3.4)$$

The ratio between operations and communication time is given by

$$\frac{T_{\text{op}}}{T_{\text{comm}}^\pi} = \frac{(N_A C_{\text{gen}} + T_{\text{swap}} + T_{\text{cut}})c}{(C_1^\pi + C_2^\pi)(n - 1)l} + \frac{N_A}{(C_1^\pi + C_2^\pi)(n - 1)}. \quad (3.5)$$

From (3.5) we can distinguish two different regimes of parameters:

- If the distance between nodes is sufficiently small, the communication distance becomes negligible and consequently $T_{\text{op}} \gg T_{\text{comm}}^\pi$.
- If l is sufficiently large, the distance-dependent contribution of the entanglement generation time becomes significantly larger than the rest of the operations: $T_{\text{op}} \approx T_{\text{gen}} \approx N_A \frac{l}{c}$. Then, from (3.5):

$$T_{\text{comm}}^\pi \approx \frac{(C_1^\pi + C_2^\pi)(n - 1)}{N_A} T_{\text{op}} \quad \text{for } l \gg 1. \quad (3.6)$$

Let us consider the realization of a three-node quantum network [16] as an example. In this work, nitrogen-vacancy center (NV) electronic spin is used as a communication qubit, and carbon-13 nuclear spin as a memory qubit. Remote entanglement generation is attempted between nodes separated 30 m and 2 m. The resulting experimental times for entanglement swapping and link discarding (memory reset) are 1 ms and 0.6 ms, respectively. Replicating the authors' choice of limiting the number of entanglement generation attempts (N_A) to 450 to avoid dephasing errors, the resulting entanglement generation time is approximately 2.6 ms. Given the short distances of the setup, it can be assumed that the heralded signals' contribution to the total operations time is negligible and thus $T_{\text{gen}} = N_A C_{\text{gen}} \approx 2.6$ ms. Altogether, the total operations time is $T_{\text{op}} \approx 4$ ms. The distance between nodes for which the time contribution of all heralding signals becomes of the order of local operations is $l \approx \frac{T_{\text{op}} c}{N_A} \approx 10^3$ m.

From the previous example it is clear that for small l and large N_A , communication time can be neglected. However, we are ultimately interested in studying the role of classical communication in metropolitan and global-scale quantum networks, where the distances between nodes are of the order of kilometers. Moreover, multiplexed quantum memories and parallel entanglement generation protocols [47], jointly with an improvement in the efficiency of repeaters, can reduce the number of (sequential) attempts in every time step N_A . Overall, it is expected that as the number of nodes grows, the communication time will have an increasingly more significant impact on the networks' performance.

3.2.2. Local, global, and minimum-knowledge policies

Entanglement distribution policies can be classified according to the maximum distance at which nodes are allowed to communicate classically within each communication round, d_{cci}^π . This distance could, in principle, vary from node to node, as will be illustrated in Section 4.3, or even be modified in between time steps. However, for simplicity, d_{cci}^π will be considered constant in time and equal for all nodes unless explicitly stated otherwise for the rest of the manuscript.

Let us consider the two extreme cases: policies in which nodes do not communicate with any other node (heralding entanglement generation enforces at least communication between adjacent nodes, but that contribution is included in the operations time T_{gen}) and policies in which all-to-all communication within one time step is allowed. We will refer to the first type as local-knowledge policies and to the latter as global-knowledge.

Definition 3.2.1 An entanglement distribution policy π is a *local-knowledge (LK)* policy in a given classical communication round i if it satisfies $d_{cci}^\pi = 0$.

Definition 3.2.2 An entanglement distribution policy π is a *global-knowledge (GK)* policy in a given classical communication round i if it satisfies $d_{cci}^\pi = (n - 2)l$.

Communication time is an important factor in the design of entanglement swapping policies, but ultimately the main quantity of interest is the expected end-to-end delivery time. For a given policy π , this is calculated as the product of the average number of time steps required to generate end-to-end entanglement N^π , and the duration of each time step:

$$T^\pi = N^\pi T_{\text{step}}^\pi. \quad (3.7)$$

GK policies consider the whole state of the system and thereby can take actions that result in better payoffs in terms of N^π , at the expense of longer time steps. On the contrary, in LK policies time steps are shorter, but they potentially require a larger number of them to generate end-to-end entanglement. Another design option can be found midway between these two alternatives, allowing *some* but not *all* communication between nodes. As discussed below, a certain amount of classical information in the second communication round is always needed to satisfy the constraints of our model:

- Firstly, the adopted cutoff policy, consisting of removing all links that were generated before a time window of length t_{cut} , presumes that both nodes sharing a link agree on discarding it. This means that nodes should know the post-swapping ages of their links to perform cutoffs. Otherwise, undesired situations in which nodes mistakenly believe they share entanglement may arise.
- Secondly, most quantum communication applications where entanglement is needed (e.g., quantum teleportation [48], E91 [2]) require that the involved parties know that they share an entangled pair before the

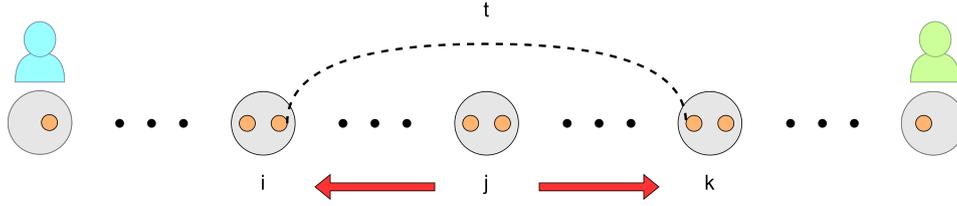


Figure 3.1: Illustration of classical information transfer in a repeater chain. If nodes i and k share an entangled link with a node j , which performs an entanglement swap. Information needs to be transmitted from node j to nodes i and k to communicate the success or failure of the operation and the post-swapping age of the link.

rest of the protocol is executed. In our case, both of the end nodes must receive a confirmation that end-to-end entanglement has been achieved.

For these reasons, classical information must be transmitted at some point between remote nodes and therefore fully local-knowledge policies (LK in both communication rounds) are not implementable in our quantum repeater chain model. To determine the minimum communication distance d_{cc2}^π that satisfies the restrictions of our model, let us consider an entangled link between two nodes i and k ($i < k$). Let us assume that such a link is the result of a swap in node j ($i < j < k$), as shown in Fig. 3.1. If the age of the link is larger than t_{cut} , it must be removed. As a consequence, nodes i and k must have information regarding the existence of this link and its post-swapping age, which are conditional on the success of the swap in node j . Therefore, communication between node j and nodes i and k is necessary. The most extreme scenario occurs when i and k are the end nodes ($i = 0, k = n - 1$) that are waiting for a confirmation that end-to-end entanglement has been achieved. In this case, the maximum distance that information needs to travel during the second communication round depends on the position of node j :

$$d_{cc2}^\pi = \max(n - 1 - j, j)l. \quad (3.8)$$

(3.8) can be used to set a lower bound on d_{cc2}^π , by setting $j = \frac{n}{2}$ or $\frac{n-1}{2}$ (depending on whether n is even or odd). We will denote policies with minimum d_{cc2}^π as *minimum-knowledge (MK)* in communication round two. By extension, the policies that also satisfy the minimum value of (3.8) for d_{cc1}^π , will also be called *minimum-knowledge (in the communication round one)*. Note that in this case d_{cc1}^π is not the minimum possible since policies can be LK in the first communication round. A necessary condition for MK policies is that they must enforce that the last swap before achieving end-to-end entanglement occurs exactly in the middle of the chain (or one of the middle nodes if the number of nodes is even).

Definition 3.2.3 An entanglement distribution policy π is a *minimum-knowledge (MK)* policy in a given classical communication round i if it satisfies:

$$d_{cci}^\pi = \begin{cases} \frac{n-1}{2}l & \text{if } n \text{ odd} \\ \frac{n}{2}l & \text{if } n \text{ even.} \end{cases} \quad (3.9)$$

Note that the same policy can have different communication distances for each communication round, so we use the notation AK-BK to describe the total communication time required within a time step. A and B define the type of policy (local, global, minimum, etc.) during the first and second communication rounds, respectively. For example, if a policy is local-knowledge during the first communication round and global-knowledge in the second, it will be denoted LK-GK, and GK-GK if it requires global information in both rounds.

The policies with the smallest classical communication distance are LK-MK (local-knowledge in the first communication round and minimum-knowledge in the second). Therefore, the minimum value for a time step is given by:

$$T_{\text{step}}^{\pi} \geq T_{\text{step}}^{\text{LK-MK}} = T_{\text{op}} \left(1 + \frac{d_{\text{cc}2}^{\text{MK}}}{T_{\text{op}}c} \right). \quad (3.10)$$

Here $d_{\text{cc}2}^{\text{MK}}$ is the corresponding communication distance of an arbitrary MK policy in the second communication round. If policies allow the last swap to be performed in one of the nodes adjacent to the extremes ($j = 1$ or $j = n - 2$), $d_{\text{cc}2}^{\pi}$ will be maximum and we recover the expression for GK policies ($d_{\text{cc}i}^{\pi} = (n - 2)l$). For repeater chains with three and four nodes, there is no difference between GK and MK policies, so all policies are global-knowledge during the second round of communication. Note that the physical rate at which memory qubits decohere is the same in all settings. However, having shorter time steps implies that cutoff times can be larger, as links need more time steps to reach the same fidelity threshold.

3.3. The swap-asap policy

The swap-asap policy is the simplest entanglement distribution policy. It performs entanglement swaps *as soon as possible*. That is, nodes perform swaps whenever they share two entangled links, one towards each end of the chain (see Fig. 3.2). Due to its simplicity, this policy has been used as a baseline to design cutoff policies using deep reinforcement learning [42], and is commonly used as a benchmark to evaluate the performance of other entanglement distribution policies [43, 31]. In addition, in the case of low hardware quality, swap-asap has been shown to produce higher end-to-end fidelity than the “nested-with-distill” protocol considered in [49], and faster entanglement rates than entanglement purification protocols [50].

Swapping links *as soon as possible* presents certain advantages. Firstly, performing swaps in a node always frees up its qubits, irrespective of the success or failure of the operation. Those qubits can then be used to resume entanglement generation in the following time steps. Secondly, by minimizing the waiting time of the existing links, high fidelity end-to-end links can be achieved. This allows to employ swap-asap even when the end-to-end fidelity requirements are very strict (i.e., when F_{min} is close to 1). The main drawback of the swap-asap policy emerges when swaps are probabilistic. Two swaps

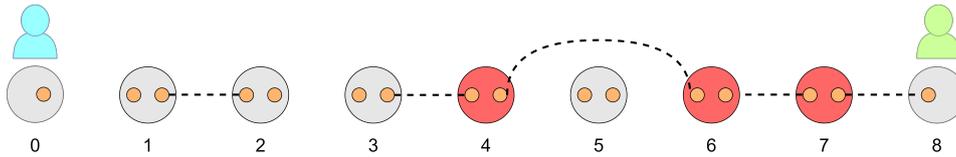


Figure 3.2: Example of swap-asap actions. Nodes perform swaps whenever they share two entangled links, one towards each side of the chain. Nodes that perform a swap are represented in color red.

are adjacent if they share a common link. In case simultaneous adjacent swaps are attempted, a new longer link will only be generated if all swaps are successful. That is, the probability of producing a new link is p_s^x , being x the number of adjacent nodes simultaneously attempting swaps. With probability $1 - p_s^x$, all entanglement will be lost.

In the swap-asap policy, nodes do not need to know any information about the system besides that they share elementary links with their neighbours (which is always known from the heralding signals in the entanglement generation step). Consequently, swap-asap is local during the first communication round. Nevertheless, as discussed in Section 3.2.2, the maximum distance that information can travel during the second communication round depends on the position of the last swap before generating end-to-end entanglement. In the case of swap-asap, there are no restrictions on the order of swaps, so the last swap could happen neighbouring to either end node. Therefore $d_{cc2}^{\text{swap-asap}} = (n - 2)l$ and we can conclude that swap-asap is a LK-GK policy. Incorporating global-knowledge of the state of the system usually entails better decision-making. Nonetheless, swapping actions taken by swap-asap are independent of the full state of the system, which implies that this policy is in many cases not optimal.

A deeper insight into the swap-asap policy can be obtained through simulation results. Figure 3.3 shows the one-time delivery time for a repeater chain of eight nodes as a function of the entanglement generation probability p for probabilistic swaps ($p_s = 0.5$). In most cases, increasing the value of p corresponds to a decrease in the delivery time. However, beyond a certain value of p (approximately 0.8 in our example), the delivery time does not decrease further. In fact, it starts increasing again when p approaches one.

The reason of this counterintuitive behaviour for probabilistic swaps is the number of adjacent swaps that are attempted simultaneously. As the entanglement generation probability p grows, it becomes more likely that adjacent nodes swap in the same time step, losing all entanglement with a high probability. This increases the expected end-to-end delivery time, especially in larger chains, where more adjacent swaps can occur simultaneously. We show in Theorem 1 that if the end-to-end delivery time of a policy is not monotonically decreasing in p for a given interval $p \in [p_1, p_2]$, that policy is suboptimal in terms of delivery time in that interval. The basic idea behind

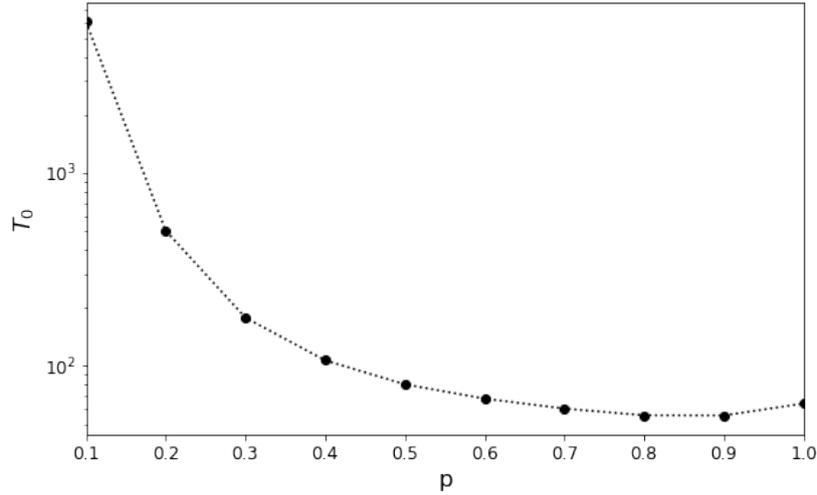


Figure 3.3: The end-to-end delivery time for the swap-asap policy is non-monotonic with respect to p . Expected one-time delivery time T_0 (measured in number of time steps) versus entanglement generation probability p for the swap-asap policy in a chain of 8 nodes with swapping probability $p_s = 0.5$. Data points correspond to averaging over 100000 shots in a Monte Carlo simulation. Error bars are not displayed since they are smaller than the symbol size.

this statement can be elucidated by considering a simple example. Given two entanglement generation probabilities p_1 and p_2 ($p_1 < p_2$), let us assume that a certain policy π shows a better performance for p_1 than for p_2 . If this is the case, a naive strategy to increase the performance of the policy at p_2 would be to artificially decrease the entanglement generation probability to p_1 . This could be done by systematically removing a fraction of the generated links immediately after being generated. We will denote the new policy as π^λ , where π refers to the entanglement swapping policy, and $1 - \lambda$ is the fraction of links that are discarded after being generated.

Definition 3.3.1 The *preserved link fraction* (λ) is the probability of newly generated elementary links being kept. With probability $(1 - \lambda)$, they will be immediately discarded after generation.

In the current definition of time step, discarding links immediately after being generated is not considered. However, note that this is equivalent to defining an “effective” entanglement generation probability $p^{\text{eff}} \equiv \lambda p$ (Fig. 3.4). In general, it is not desirable to reduce the entanglement generation probability of quantum repeaters in this manner, but the preserved link fraction is a concept that will be helpful in the following discussion. Unless otherwise stated, it will always be assumed $\lambda = 1$, so all links are kept after being generated.

Lemma 1 *Given two entanglement distribution policies with an identical entanglement swapping policy π and preserved link fractions λ_1 and λ_2 ($\lambda_1 \leq \lambda_2$), the following holds:*

$$T_s^{\pi^{\lambda_1}}(n, p, p_s, t_{\text{cut}}) = T_s^{\pi^{\lambda_2}}\left(n, \frac{\lambda_1}{\lambda_2}p, p_s, t_{\text{cut}}\right) \quad \forall s \in \mathcal{S}, \quad (3.11)$$

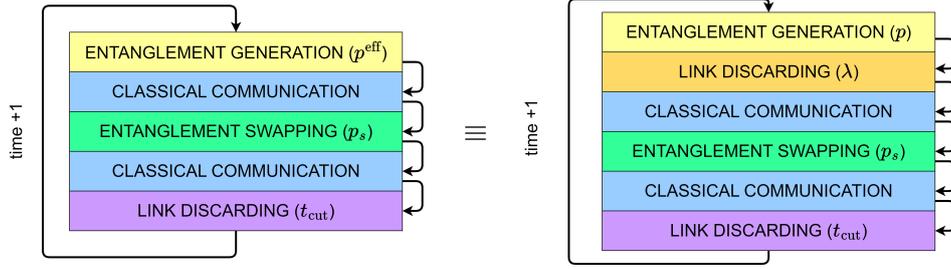


Figure 3.4: Introducing a link discarding stage in the time step is equivalent to considering an effective entanglement generation probability p^{eff} .

where $T_s^\pi(n, p^{\text{eff}}, p_s, t_{\text{cut}})$ is the average time required to reach end-to-end entanglement from the state s using the policy π , for a chain of n nodes, effective entanglement generation probability p^{eff} , swapping probabilities p_s , and cutoff time t_{cut} .

Proof. Generating a link and immediately discarding is equivalent to failing to generate it in the first place, so two systems with the same effective entanglement generation probability and all other parameters equal are identical. The effective entanglement generation is $p^{\text{eff}} = \lambda p$. Therefore:

$$T_s^{\pi^{\lambda_1}}(n, p, p_s, t_{\text{cut}}) = T_s^\pi(n, \lambda_1 p, p_s, t_{\text{cut}}) = T_s^{\pi^{\lambda_2}}\left(n, \frac{\lambda_1}{\lambda_2} p, p_s, t_{\text{cut}}\right). \quad \blacksquare \quad (3.12)$$

Theorem 1 Given two entanglement distribution policies π^λ and $\pi^{\lambda'}$, if $\exists p, p'$ s.t. $p \geq \lambda' p'$ and $T_0^{\pi^{\lambda'}}(n, p', p_s, t_{\text{cut}}) < T_0^{\pi^\lambda}(n, p, p_s, t_{\text{cut}})$, then, π^λ is suboptimal for the set of parameters $(n, p, p_s, t_{\text{cut}})$.

Proof. Consider the entanglement distribution policy $\pi^{\tilde{\lambda}}$, where $\tilde{\lambda} = \frac{\lambda' p'}{p} \leq 1$. Using Lemma 1

$$T_0^{\pi^{\tilde{\lambda}}}(n, p, p_s, t_{\text{cut}}) = T_0^{\pi^{\lambda'}}(n, p', p_s, t_{\text{cut}}). \quad (3.13)$$

Then, by assumption, $T_0^{\pi^{\lambda'}}(n, p', p_s, t_{\text{cut}}) < T_0^{\pi^\lambda}(n, p, p_s, t_{\text{cut}})$, so

$$T_0^{\pi^{\tilde{\lambda}}}(n, p, p_s, t_{\text{cut}}) < T_0^{\pi^\lambda}(n, p, p_s, t_{\text{cut}}), \quad (3.14)$$

and finally we conclude that π^λ is suboptimal in terms of end-to-end delivery time for the given set of parameters, as there exists at least one policy ($\pi^{\tilde{\lambda}}$) with better performance: $\exists \pi^{\tilde{\lambda}}$ s.t. $T_0^{\pi^{\tilde{\lambda}}} < T_0^{\pi^\lambda}$, for the parameters $(n, p, p_s, t_{\text{cut}})$. \blacksquare

In particular, Theorem 1 holds when $\pi^\lambda = \pi^{\lambda'}$, and therefore any entanglement distribution policy with an end-to-end delivery time that is not monotonically decreasing with respect to p in an interval $p \in [p_1, p_2]$, is automatically suboptimal (in terms of end-to-end delivery time) in that interval.

Resuming the discussion on the non-monotonicity of swap-asap, we had speculated that a possible cause for this effect could be that this policy allows

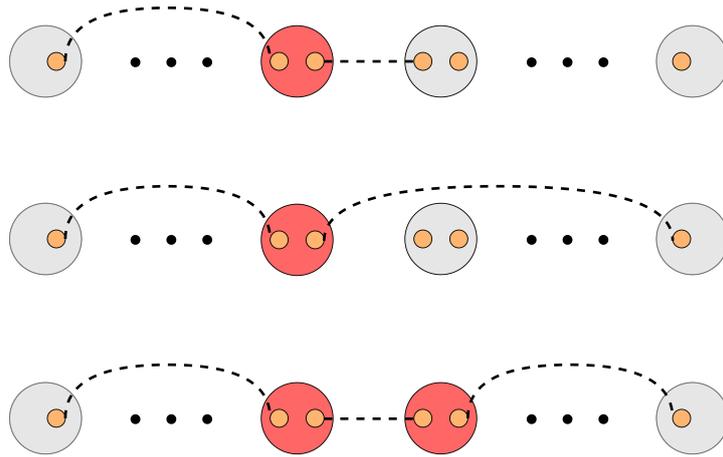


Figure 3.5: Illustration of the actions taken by the central nodes in the middle-last policy for a chain with an even number of nodes. Figure represents the three types of states in which at least one central node performs a swap. A central node performs a swap (represented in color red) only if it shares a link with its closest end node and any other link towards the other end of the chain. Both central nodes swap simultaneously only if they both share an entangled link with their respective closest end node (bottom state).

multiple adjacent swaps in each time step. To test this hypothesis, we can study if preventing some of these swaps at every time step has a positive or negative impact on the average end-to-end delivery time. In particular, we will use a naive policy in which all nodes except the central one attempt swaps *as soon as possible*. On the other hand, the central node, in the case of an odd number of nodes, will wait until it shares an entangled link with each of the end nodes to perform the swap. In the case of an even number of nodes in the chain, both of the central nodes will wait until they share entanglement with their respective closer end node (shown in Fig. 3.5). In either scenario (even or odd number of nodes), the central repeater(s) must wait until the rest of the swaps in the chain have been successful, consequently limiting the maximum number of adjacent swaps that can happen in a single time step to $\frac{n-3}{2}$ (odd n) or $\frac{n-4}{2}$ (even n). This policy will be denoted “middle-last” (it enforces that the final swap happens in the middle of the chain).

One notable attribute of swap-asap and middle-last is that the number of possible states of the system does not grow with n under the condition of deterministic entanglement generation, $p = 1$ (and infinite cutoff time in the case of middle-last). This allows us to solve the Bellman equations (3.15) analytically to find a closed-form expression for the end-to-end delivery time. Let us first consider the swap-asap case.

The Bellman equations describe the relation between the delivery times of different states for a given policy π [31]:

$$T_s^\pi = 1 + \sum_{s' \in \mathcal{S}} P(s'|s, \pi) T_{s'}^\pi, \quad (3.15)$$

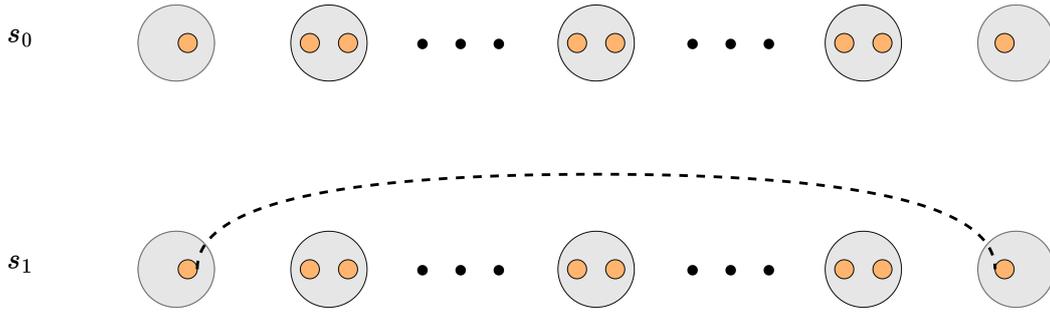


Figure 3.6: Possible states in the swap-asap policy when entanglement generation is deterministic ($p = 1$). The state s_0 represents the empty chain (no entanglement links) and in the state s_1 , the two end nodes share an entangled link.

where T_s^π is the average number of time steps needed to achieve end-to-end entanglement starting from state s under policy π , \mathcal{S} is the state space, and $P(s'|s, \pi)$ is the transition probability of reaching state s from state s' when following policy π .

If entanglement generation is deterministic, there are only two possible states for a chain under the swap-asap policy after each time step (see Fig. 3.6): the empty chain (s_0), and the state with an end-to-end entangled link with age 0 (s_1). The probability of transitioning from s_0 to s_1 is the probability that all $n - 2$ swaps simultaneously succeed: $P(s_0|s_1, \pi^{\text{swap-asap}}) = p_s^{n-2}$. With probability $1 - p_s^{n-2}$, the system will go back to s_0 . Substituting in (3.15) we find the expression for the expected delivery time of the swap-asap policy from the empty state, $T^{\text{swap-asap}}(s_0)$:

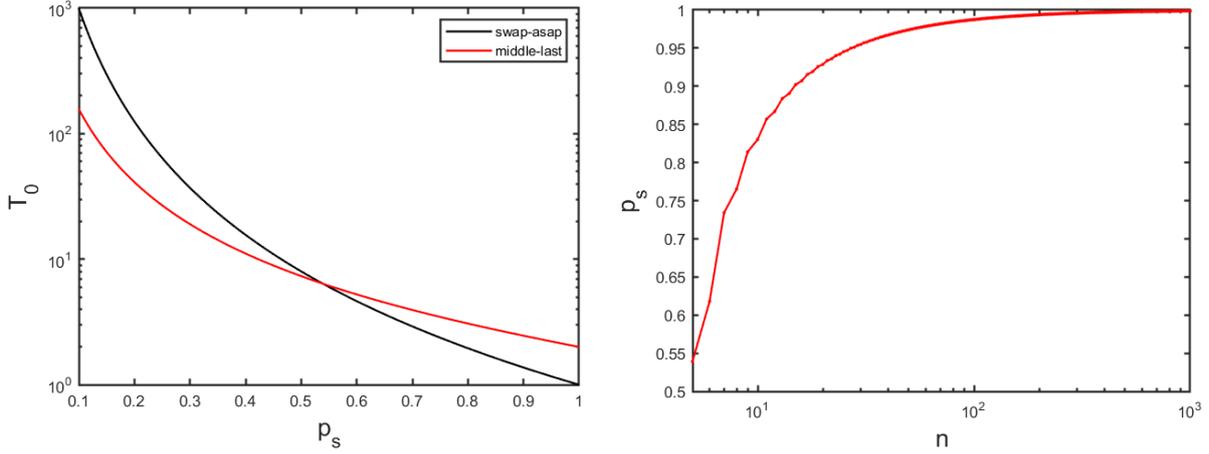
$$T_{s_0}^{\text{swap-asap}} = 1 + p_s^{n-2} T_{s_1}^{\text{swap-asap}} + (1 - p_s^{n-2}) T_{s_0}^{\text{swap-asap}} \quad \text{if } p = 1. \quad (3.16)$$

Solving (3.16) using that s_1 is already an end-to-end entangled state and therefore $T_{s_1}^{\text{swap-asap}} = 0$ yields:

$$T_{s_0}^{\text{swap-asap}} = \frac{1}{p_s^{n-2}} \quad \text{if } p = 1. \quad (3.17)$$

The previous expression is consistent with the probability of generating end-to-end entanglement in one time step (p_s^{n-2}), as it is needed, on average, to perform $\frac{1}{p_s^{n-2}}$ attempts before achieving end-to-end entanglement. Note also how this result is independent of the cutoff time t_{cut} . From (3.17), it is evident that small changes in swapping probability can produce significant variations in the delivery time. This makes the swap-asap policy particularly susceptible to fluctuations in the swapping probability, which is in general not desirable (as explained in Section 3.1.2).

Using the same procedure, similar expressions can be found for the average end-to-end delivery time for the middle-last policy, $T^{\text{middle}}(s_0)$. The full

(a) Middle-last is faster than swap-asap for low p_s .

(b) Intersection between swap-asap and middle-last.

Figure 3.7: Comparison between swap-asap and middle-last when entanglement generation is deterministic. Figure (a) shows the one-time end-to-end delivery time (measured in number of time steps) versus the swapping probability p_s for the middle-last and swap-asap policies in a chain of five nodes. Communication time is neglected, so the time step length of both policies is assumed to be equal. Figure (b) shows the value of p_s at which the intersection between the two policies occurs, as a function of the number of nodes.

derivation is provided in Appendix B:

$$T_{s_0}^{\text{middle}} = \frac{3 - p_s^{n-3}}{2p_s^{\frac{n-1}{2}} - p_s^{n-2}} \quad \text{if } p = 1, t_{\text{cut}} = \infty \text{ and } n \text{ odd.} \quad (3.18)$$

$$T_{s_0}^{\text{middle}} = \frac{1 + 2p_s + p_s^{\frac{n-4}{2}} - p_s^{\frac{n-2}{2}} - p_s^{n-4} + p_s^{3n-11} - p_s^{\frac{3n-12}{2}}}{2p_s^{\frac{n}{2}} - p_s^{n-2} - p_s^{\frac{3n-4}{2}} + p_s^{\frac{3n-2}{2}}} \quad \text{if } p = 1, t_{\text{cut}} = \infty \text{ and } n \text{ even.} \quad (3.19)$$

Figure 3.7a shows the expected delivery time for the middle-last and swap-asap policies when the entanglement generation is deterministic in a chain of five nodes. As expected, swap-asap shows a better performance for high entanglement swapping probabilities, but it is slower for low p_s . Note that in the limit of ideal parameter values ($p = p_s = 1$), $T_{s_0}^{\text{swap-asap}} = 1$ while $T_{s_0}^{\text{middle}} = 2$. That is, swap-asap is the optimal policy for chains of any length when $p = p_s = 1$. However, the minimum value of p_s for which swap-asap outperforms middle-last is dependent on the chain length (see Fig. 3.7b). In particular, for chains with a large number of nodes, middle-last produces a lower average end-to-end delivery time for almost every value of $p_s < 1$. It can then be concluded that for non-deterministic swaps ($p_s \neq 1$) and sufficiently large chains, preventing too many adjacent simultaneous swaps can decrease the average end-to-end delivery time with respect to swap-asap. This idea, jointly with minimizing the communication time between nodes, will be the main elements of design for our proposed policies in the next chapter.

Policy Candidates

In previous chapters, we have explored the main challenges for designing entanglement distribution policies for an arbitrary number of nodes in a repeater chain. In particular, the state space of our repeater chain model increases exponentially with the number of nodes [31], which renders an increase in the number of design choices for our policies. To narrow down our search we follow the principles described in Section 3.1: scalability, robustness, determinism, monotonicity, and symmetry. We will also incorporate the strategies found in Section 3.3 to avoid some of the undesirable features of the swap-asap policy, such as the non-monotonicity of its end-to-end delivery time with respect to p . Moreover, we are interested in reducing the classical communication time in our policies by ensuring that the last swap before achieving end-to-end entanglement occurs in one of the central nodes in the chain.

Depending on the specific application of interest and hardware available, it may be convenient to focus on one or more of the principles mentioned above and relax some of the other constraints, yielding distinct policies suitable for different ranges of parameters. In this chapter, we propose three policy candidates: the “nonadjacent”, the “nonadjacent-middle-last”, and the “nested” policies. We will also describe the main differences and similarities between them and consider particular examples of their implementation. The performance of these policies will be discussed in Chapter 5.

4.1. The nonadjacent policy

Our first candidate is the “nonadjacent” policy. In this policy, all nodes perform their swaps as soon as possible, unless there is a “large enough” group of adjacent links in the chain. In that case, some of the nodes holding those links will wait instead. This prevents the loss of all entanglement within the group in case of a failed swap. In this policy, nodes must know the full state of the system before swapping, and there is no fixed position for the last swap before generating end-to-end entanglement. Therefore, the nonadjacent policy is GK-GK. In principle, better performing GK-GK policies can be found using dynamic programming or deep reinforcement learning techniques, but computing them becomes exponentially more expensive as the number of nodes in the chain grows. Moreover, the simplicity of this policy enables an easier implementation in real experimental setups.

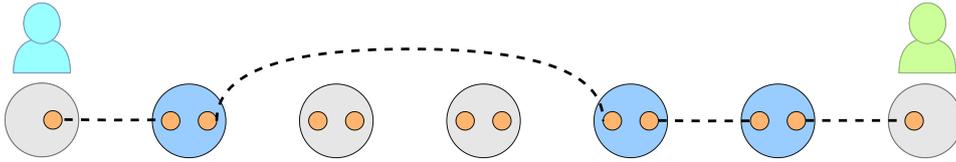


Figure 4.1: Diagram of a connected component. Nodes colored in blue are part of the same connected component, despite not being physical neighbours. To know the size of their connected component, classical information needs to be transmitted between all intermediate nodes.

Note that in this policy there are a few additional parameters that should be considered, such as the maximum number of adjacent swaps that are allowed within one time step and how the waiting nodes are distributed along the chain. We will discuss our choice of these parameters in Sections 4.1.2 and 4.1.3 respectively. Before that, we will define exactly what a connected component and adjacent swaps mean in our context.

4.1.1. Connected component

The concept of connected component is widely used in graph theory. For every pair of vertices in a connected component, there must exist a path that connects them [51]. We adapt this definition to describe the sets of nodes that are connected by entangled links. Formally:

Definition 4.1.1 A *connected component* is a group of nodes in which every pair of nodes can share an entangled link through a sequence of successful swaps. In addition, every node in the connected component must share two entangled links (one towards each side of the chain). The length of the connected component is determined by its number of nodes.

Note that according to this definition, nodes can be part of a connected component despite not being physical neighbours of any of the other elements in the chain (see Fig. 4.1). In particular, any pair of nodes can in principle share an entangled link, becoming part of the same connected component. In our model, nodes can only share one entangled link with each extreme of the chain, and therefore any given node belongs, at most, to one connected component. Given a connected component of size $k \leq n - 2$, we can label the nodes with indices ranging from 0 to $k - 1$ (from leftmost to rightmost). To avoid confusion, we will denote “relative” index to the position of a node within a connected component and “absolute” index to the position of a node within the full chain. Adjacent swaps were previously introduced in Section 3.3, but can also be defined as follows:

Definition 4.1.2 A set of *adjacent swaps* is a collection of entanglement swapping attempts performed by two or more nodes with consecutive relative indices within the same connected component.

In the nonadjacent policy, at every time step, nodes need to be aware of the size of their connected component before attempting swaps. This involves

communication during the first communication round between nodes in the same connected component, as illustrated in Figure 4.1. The maximum separation between two nodes in a connected component is $(n-3)l$. This does not exactly correspond to the communication distance of a GK policy, $d_{cc}^{\text{GK}} = (n-2)l$. However, since this difference does not have a major impact in the performance of the policy, for simplicity we define this policy as GK in the first communication round. The last swap before achieving end-to-end entanglement can happen in any node of the chain. Hence, the nonadjacent policy falls into the category of GK-GK policies.

4.1.2. Number of allowed swaps

The maximum number of allowed adjacent swaps can critically affect the performance of the nonadjacent policy. For instance, if $n-2$ adjacent swaps are allowed, we recover the swap-asap policy. The optimal value of this parameter may depend on the entanglement swapping probability p_s , the chain length n and the entanglement generation probability p . However, for simplicity, we opted to keep the maximum number of allowed swaps as a constant. After analyzing the optimal solutions found by using the dynamic programming methods described in [31], we decided to set the maximum number of allowed adjacent swaps to two, as the nonadjacent policy, in this case, is shown to be optimal for chains of length three, four and five nodes for deterministic entanglement generation ($p=1$) and $p_s=0.5$. For the rest of the discussion, it will be always assumed that the maximum number of allowed adjacent swaps is two. However, the implementation of the nonadjacent policy is equivalent to other choices of this parameter.

4.1.3. Distribution of waiting nodes

For a given connected component of size k , there are multiple choices for placing the waiting nodes in a way such that the maximum number of adjacent swaps does not exceed a certain quantity. There are two degrees of freedom for this choice: the number of waiting nodes and their position in the connected component. We opted to use the minimum possible number of waiting nodes, since adding “unnecessary” waiting actions can reduce the performance of the policy in terms of end-to-end delivery time and end-to-end fidelity. For a connected component of length k and a maximum number of allowed adjacent swaps $M \leq n-2$, the minimum required number of waiting nodes n_w is given by:

$$n_w = \left\lceil \frac{k}{M+1} \right\rceil. \quad (4.1)$$

To increase the probability of preserving entanglement after swapping, we also found appropriate to minimize the number of times exactly M adjacent swaps occur simultaneously. This can be done by evenly spacing the waiting nodes (separating their relative indices by a distance of $M+1$) and placing them symmetrically within their connected component.

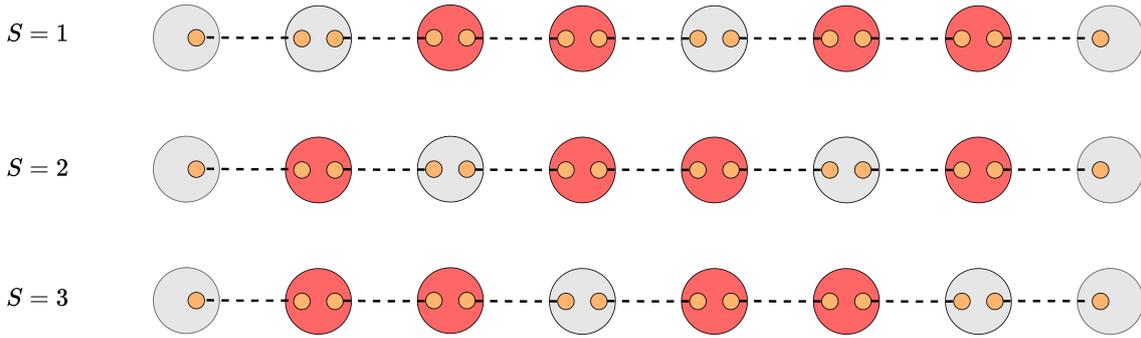


Figure 4.2: Position of waiting nodes in a connected component. We consider a connected component of length six, with $M = 2$, $n_w = 2$ and $d_w = 3$. Each row corresponds to the first waiting node having index 1, 2 and 3, respectively. Nodes that perform a swap are colored in red. Whereas in the first and last case, there exist two groups of two adjacent swaps occurring simultaneously, in the case with $S = 2$, there is only one group of two simultaneous adjacent swaps. Note that in this case waiting nodes are placed symmetrically with respect to the connected component.

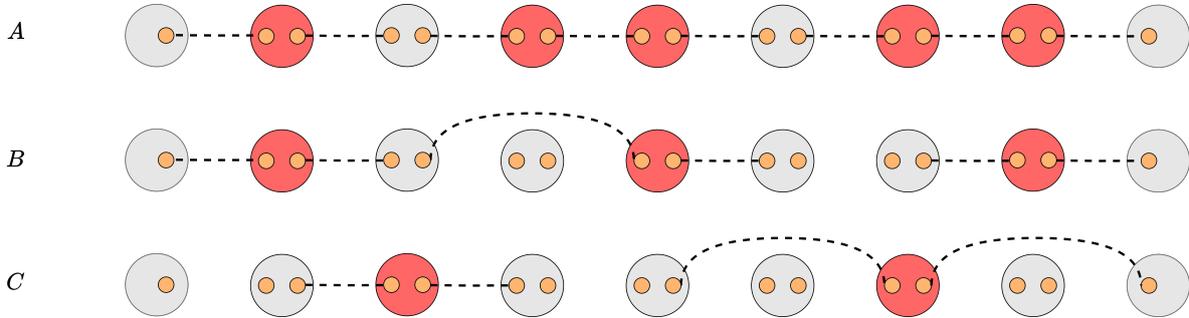


Figure 4.3: Examples of nonadjacent policy actions in a chain of nine nodes. Nodes perform swaps (represented in color red) *as soon as possible*, except when there is a connected component with length larger than $M = 2$. In that case, waiting nodes are placed according to Eqs. (4.1), (4.2) and (4.3).

The distance d_w between the first and last waiting nodes that minimizes the number of times M simultaneous adjacent swaps happen is given by:

$$d_w = (M + 1)(n_w - 1). \quad (4.2)$$

Figure 4.2 illustrates different choices of the position of the waiting nodes in the connected component. To impose symmetry, we enforce that the number of nodes of the connected component placed to the left of the first waiting node is equal (or as close as possible) to the number of nodes to the right of the last waiting node. Then we chose the following relative index of the first waiting node within the connected component S :

$$S = \left\lfloor \frac{k - d_w - 1}{2} \right\rfloor. \quad (4.3)$$

Figure 4.3 shows the actions taken by the nodes for a maximum of two allowed adjacent swaps; the relative index of the first waiting node is given by (4.3). Note that for two connected components of the same size, the actions of the nodes with equal relative index will be identical in both cases, irrespective of the state of the rest of the chain or its length.

4.2. The nonadjacent-middle-last policy

The second proposed candidate is the nonadjacent-middle-last (nml) policy, which can be considered as a mixture of the nonadjacent policy and the middle-last policy described in Section 3.3. Similarly as in the nonadjacent policy, there is a maximum number of allowed adjacent swaps. But in this case, the last swap before generating end-to-end entanglement must occur in the middle node of the chain (or one of the two middle nodes if the chain length is even). This entails that the middle nodes wait until they receive confirmation that they share an entangled pair with both end nodes (or the closest end node for an even number of nodes in the chain).

This seemingly minimal change with respect to the nonadjacent policy can have deep implications on the end-to-end delivery time and fidelity, as well as on the required communication time. Firstly, since the middle node(s) have to wait until the rest of the swaps have been performed, no entangled links can be shared between nodes at different sides of the chain (to the left or right of the middle node(s)). Therefore all connected components must be located at either side of the chain. This means that no classical information needs to cross from the left to the right of the middle node(s) in any communication round, so nml is a MK-MK policy. In addition, enforcing the last swap to happen in the middle of the chain has some further implications. Intuitively, including an extra waiting step may decrease the end-to-end fidelity and should increase the average end-to-end delivery time. On the other hand, the “value” of an entangled link depends on its length (and on its age to a lesser extent) as longer links require more time and resources to be generated. For this reason, losing a long-distance entangled link typically involves a larger increase in delivery time than losing elementary links. In the nml policy, middle nodes only swap simultaneously if the chain length is even and both achieve entanglement with their respective closest end nodes in the same time step. In all other cases, only a single swap is attempted in the last time step, minimizing the risk of losing long entangled links. Some examples of the actions taken by this policy for different states of the system are illustrated in Figure 4.4.

4.3. The nested policy

Our last candidate for entanglement distribution policy is the nested policy. In particular, this policy can be understood as a generalization for an arbitrary number of nodes of the “doubling” policies considered in [44] and originally proposed in [13, 52] in the context of entanglement purification. Unlike the nonadjacent policy, actions of the nodes are exclusively determined by their absolute position in the chain. For each node in the chain, the nested policy assigns two quantities: the node distance and the node type, which ultimately define the actions of the policy.

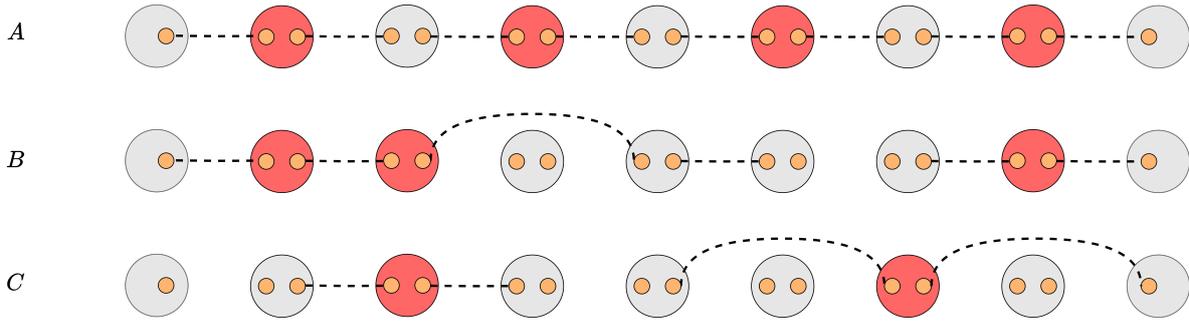


Figure 4.4: Examples of nonadjacent-middle-last policy actions in a chain of nine nodes. States A, B, and C represent possible configurations of the system. The central node must wait until it shares an entangled link with each end node before performing a swap (represented in color red), the rest of the nodes perform the same actions as in the nonadjacent policy, provided that all connected components are located at one particular side of the chain (left or right of the central node).

Definition 4.3.1 In a chain with n nodes and entanglement policy π , the *node distance* of a node with absolute index i is defined as $d_i^\pi = |i - k|$, k being the absolute index of the most distant node with which node i can communicate classically within a single time step.

For a chain with n nodes separated by a distance l and a policy π , the relation between d_i^π and the maximum communication distance d_{cc2}^π defined in section 3.2.2 is the following:

$$d_{cc2}^\pi = \max(d_{i=0,1,\dots,n-1}^\pi) \cdot l. \quad (4.4)$$

In the nested policies, we can distinguish five different types of nodes.

- Type E (end nodes): cannot perform any swap operation.
- Type S (swap-asap nodes): perform swaps as soon as possible.
- Type W (wait-symmetric nodes): perform swaps if and only if they share entangled links with two nodes at distance d_i^π .
- Type L (wait-left nodes): perform swaps if and only if they share their left link with a node at distance $d_i^\pi - 1$.
- Type R (wait-right nodes): perform swaps if and only if they share their right link with a node at distance $d_i^\pi - 1$.

Types W , L , and R nodes will be denoted as waiting nodes in contrast to type S or swap-asap nodes. The actions of a node i are completely defined by its type and communication distance. The actions of the whole chain can therefore be described using the nested vector $\vec{\nu}(n)$, which depends on the chain length. We will use the following notation for each component.

$$\nu_i(n) = A_i d_i^\pi, \quad \text{for } 0 \leq i \leq n - 1, \quad (4.5)$$

where $A_i \in \{E, S, W, L \text{ or } R\}$ classifies the node with absolute index i . Note that the actions of the nodes in the nested policy depend on the success of previous swapping operations, but not on the generation of new links (except

for swap-asap nodes, which by definition do not require any classical information during the first communication round). Then, classical information only needs to be transmitted during the second classical communication round. Since the last swap before achieving end-to-end entanglement always occurs in the central node(s), the nested policy is LK-MK, so it has the shortest possible time step length. Next, we will specify the algorithm to compute the nested vector ν for a chain of arbitrary length n .

4.3.1. Nested algorithm

Our nested algorithm uses a recursive relation to generate the nested vector $\vec{\nu}(n)$ from the nested vectors $\vec{\nu}(\frac{n}{2})$ or $\vec{\nu}(\frac{n}{2} - 1)$ for even and odd n , respectively. For the smallest possible repeater chains, with three and four nodes, the nested vectors need to be determined *a priori*. In three and four-node repeater chains, GK and MK policies have the same maximum communication distance, so swap-asap, being LK-GK, yields the minimum possible time step length. In those cases, swap-asap is also close to being optimal in terms of delivery time (it is optimal for the three-node chain) and therefore we opted to set all middle nodes as type S. For example, in a chain of length three, nodes with index 0 and 2 are end nodes, and the node with index 1 is a swap-asap node. In all cases, the node distance is 1. The complete expression of the nested vector for chains of three and four nodes can be found in Eqs. (4.6) and (4.7), respectively.

$$\vec{\nu}(3) = [E1, S1, E1]. \quad (4.6)$$

$$\vec{\nu}(4) = [E2, S2, S2, E2]. \quad (4.7)$$

We can now define the algorithm to describe the nested vector of a chain with an arbitrary number of nodes n . The cases n even and odd will be considered separately.

Odd n

1. Define the label of the central node $m = \frac{n-1}{2}$.
2. Set $\nu_m(n) = Wm$.
3. Set $\nu_0(n) = \nu_{n-1}(n) = Em$.
4. Calculate $\vec{\nu}(m+1)$.
5. Set $\nu_i(n) = \nu_{n-i-1}(n) = \nu_i(m+1)$ for $i = 1, \dots, m-1$.

Even n

1. Define the label of the central nodes $m_1 = \frac{n}{2} - 1$, $m_2 = \frac{n}{2}$.
2. Set $\nu_{m_1}(n) = Lm_2$, $\nu_{m_2}(n) = Rm_2$.
3. Set $\nu_0(n) = \nu_{n-1}(n) = Em_2$.
4. Calculate $\vec{\nu}(m_2)$.
5. Set $\nu_i(n) = \nu_{n-i-1}(n) = \nu_i(m_2)$ for $i = 1, \dots, m_1 - 1$.

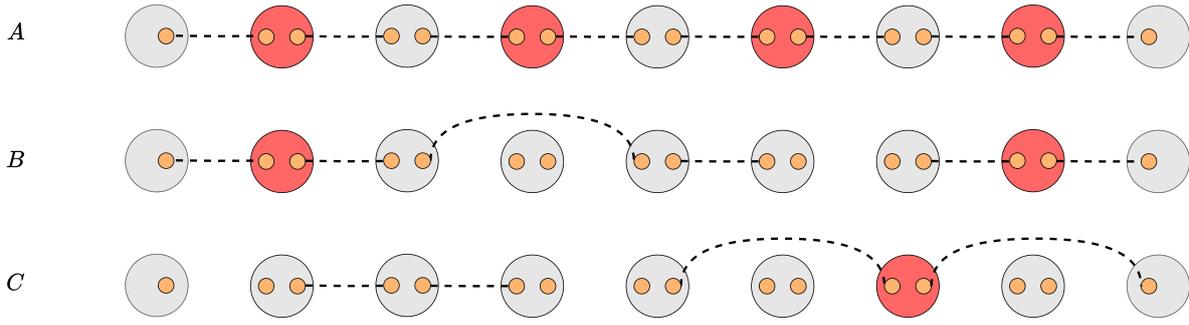


Figure 4.5: Examples of nested policy actions in a chain of nine nodes. States A, B, and C represent possible configurations of the system. Swaps are represented in color red. Nodes perform their actions based on the nested vector, in this case given by Eq. 4.12. For example, the node with index two must wait until it shares links with nodes with indexes zero and four before performing a swap.

Equations below show the explicit expression of the nested vectors for chain lengths ranging from five to ten. A step-by-step derivation of $\vec{v}(10)$ is presented as an example in Appendix C.

$$\vec{v}(5) = [E2, S1, W2, S1, E2] \quad (4.8)$$

$$\vec{v}(6) = [E3, S1, L3, R3, S1, E3] \quad (4.9)$$

$$\vec{v}(7) = [E3, S2, S2, W3, S2, S2, E3] \quad (4.10)$$

$$\vec{v}(8) = [E4, S2, S2, L4, R4, S2, S2, E4] \quad (4.11)$$

$$\vec{v}(9) = [E5, S1, W2, S1, W5, S1, W2, S1, E5] \quad (4.12)$$

$$\vec{v}(10) = [E5, S1, W2, S1, L5, R5, S1, W2, S1, E5] \quad (4.13)$$

The intuition behind nested policies is to recursively split the chain into regions of swap-asap nodes separated by nodes of type W , L , or R . These waiting nodes act as a barrier that prevents more than two adjacent entanglement swapping attempts in the same time step. To a certain extent, the nested policy can be considered as a less flexible version of nml , since it fixates the position of the waiting and swap-asap nodes along the chain. This causes that in some scenarios like the one described in Figure 4.5, the nested policy decides to wait even when there is no risk of performing multiple simultaneous adjacent swaps. In general, including these “unnecessary” waiting stages has a negative impact on both fidelity and delivery time, as will be shown in Chapter 5. However, in the nested policy, nodes can be separated into layered segments with different communication distances. This becomes an interesting feature especially when communication time overheads are relevant.

In our nested policy, every waiting node is surrounded by swap-asap nodes with a smaller node distance, as can be corroborated in the expressions for $\vec{v}(n)$. Since the actions of the waiting nodes are conditional on the success of the swaps of their neighbouring nodes, there is a predetermined order of the entanglement swapping operations. Figure 4.6 shows the dependency graph of the performed swaps for a chain of nine nodes. We can divide the nodes

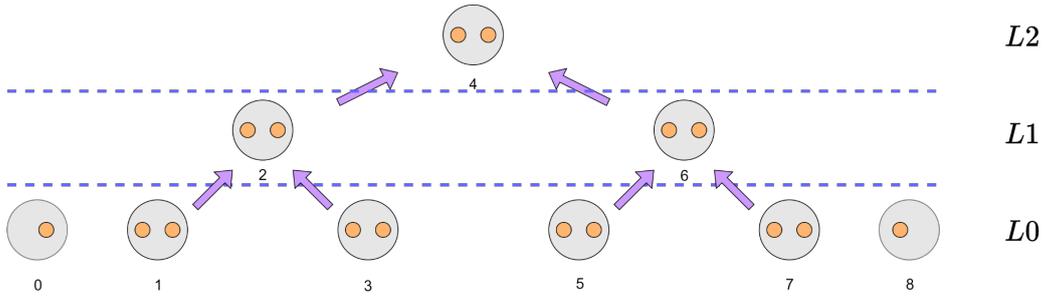


Figure 4.6: Dependency of the swaps in the nonadjacent policy in a nine-node chain. Nodes 2 and 6 (layer $L1$) can only perform swaps after the swapping operations of their respective neighbouring nodes in layer $L0$ are successful. Similarly, node 4 (layer $L2$) performs a swap after nodes 2 and 6 confirm that their operations were successful.

in the repeater chain into different layers according to their dependency relations. Actions of nodes in the bottom layer ($L0$) do not depend on any other previous action, nodes in the immediately upper layer ($L1$) only depend on actions of nodes in $L0$ and so on. The actions of each node exclusively depend on the success of the swaps from the immediately lower layer (note that the action of a node does not necessarily depend on *all* the swaps of the lower layer, but only on a group of *relevant* ones). Also, all the nodes within one layer have the same node distance. Then, it is possible to define different communication distances and therefore different time step lengths for the nodes in each layer. Nevertheless, discarding links requires some agreement between nodes that do not necessarily belong to the same layer and therefore setting variable time step lengths may give rise to synchronization issues.

We propose an alternative approach that overcomes such obstacles and is still able to reduce the time step duration below the limit for constant and global time steps given by (3.10). Instead of equal time step lengths, waiting nodes now have two independent and dynamic time step durations, one for every communication qubit. Initially, the time step duration of all nodes is set to τ_0 , which is the time corresponding to the communication distance of the layer $L0$. Before attempting any swap, waiting nodes shall receive confirmation that all the swaps on which its action depends on have been successful. When all the *relevant* swaps at one side of the chain have been successful, the time step of the communication qubit in that part of the chain is changed to τ_i ($\tau_0 < \tau_1 < \tau_2 \dots$), corresponding to the communication distance of the layer of the waiting node. After performing entanglement swapping, the length of the time step is again switched to τ_0 . This way, waiting nodes can synchronously generate entanglement and discard links with nodes of lower layers, and communicate the success or failure of the swapping operation with the nodes of higher layers. Note that a node cannot perform entanglement swap and discard a link within the same time step.

Separating the nodes in the repeater chain in layered segments does not only achieve a much higher efficiency in terms of classical communication time,

but it also paves the way for considering entanglement distribution policies in which nodes can have different hardware specifications, and optimize over those. For instance, it may be advisable to impose stricter constraints on the entanglement generation probability for nodes in the lowest layers. In the top layers, where failed swaps involve a much higher risk, a larger swapping probability could be preferred, at the expense of reducing the generation probability. These questions are however left open for future research.

5

Results

The main goal of this work is to compare our heuristic policy candidates with the swap-asap policy, which we use as a baseline. From the data collection and analysis employing the methods described in Appendix D, we present in this chapter the key results of our study. We will focus on one-time entanglement distribution and continuous entanglement distribution in Sections 5.1 and 5.2, respectively. In both cases, we will examine repeater chains with deterministic and probabilistic swaps, while varying the length of the chain from five to ten nodes. We will first assume that classical communication time is negligible and hence the time step length will be equal for all policies. If the number of nodes in the chain is equal to or less than eight, the nml and nested policies yield the same actions for all states and consequently we will only consider one of them. From the observed data, if cutoffs are sufficiently large (so that all policies have a finite end-to-end delivery time), the precise value of the cutoff does not have a significant impact on any of the measures of performance and therefore the cutoff time will be arbitrarily set to ten time steps.

Motivated by the study of future global-scale quantum networks, in which distances between nodes can be very large, we discuss in Section 5.3 how classical communication can affect the end-to-end delivery time and end-to-end fidelity of the swap-asap (LK-GK), nonadjacent (GK-GK), nml (MK-MK) and nested (LK-MK) policies. Finally, in Section 5.4, we comment on the general results obtained and propose how to further improve the performance of our policies.

5.1. One-time entanglement distribution

5.1.1. Deterministic swaps

We first consider one-time entanglement distribution assuming that swaps are deterministic. Figure 5.1 shows how the average end-to-end delivery time varies with respect to p for chains of five, seven, and ten nodes. As expected, the expected delivery time decreases for increasing probability p for all policies, and increases as the number of nodes becomes larger. For low p , the probability of the system encountering states with more than two adjacent links becomes very low, and therefore swap-asap and nonadjacent show similar performance. However, whereas the swap-asap policy requires a minimum of

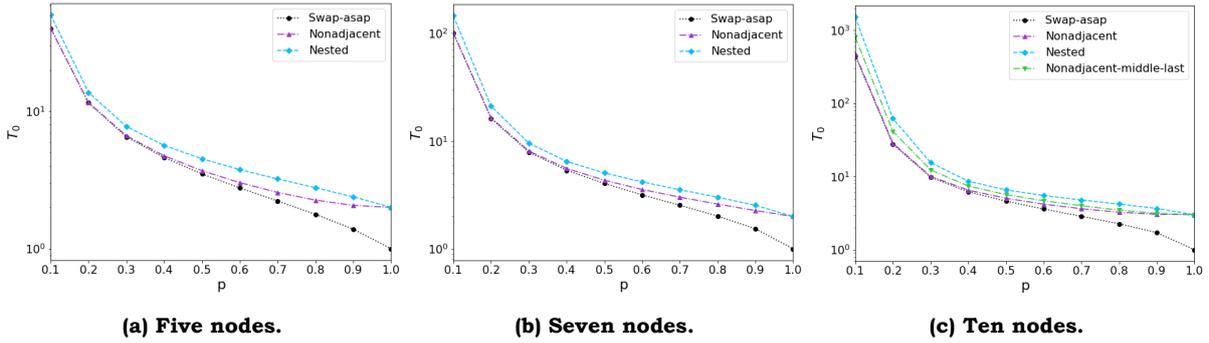


Figure 5.1: The delivery time of swap-asap is lower than that of the other policies when $p_s = 1$. End-to-end delivery time T_0 (measured in time steps) versus entanglement generation probability p when swaps are deterministic and communication time is negligible. We simulate one-time entanglement generation for repeater chains of five (a), seven (b), and ten (c) nodes. Each data point was computed by averaging 10^5 Monte Carlo shots. Error bars are not displayed since they are smaller than the symbol size.

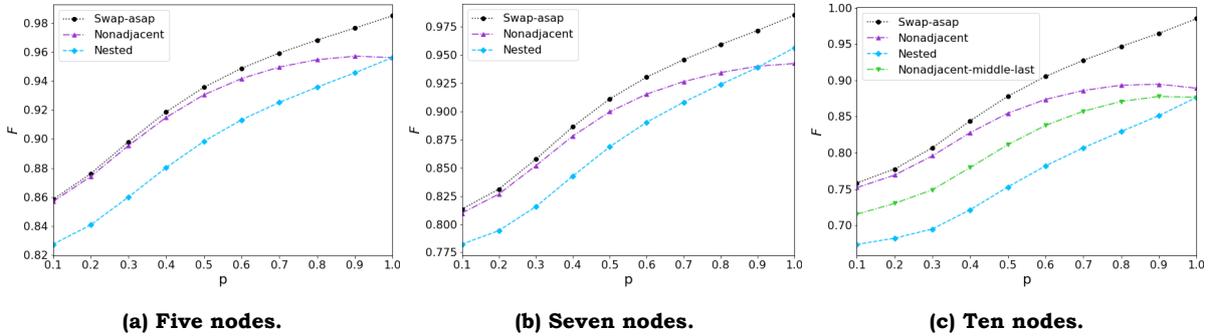


Figure 5.2: The fidelity of swap-asap is higher than that of the other policies when $p_s = 1$. End-to-end fidelity F versus entanglement generation probability p when swaps are deterministic and communication time is negligible. We simulate one-time entanglement generation for repeater chains of five (a), seven (b), and ten (c) nodes with a decoherence rate $\tau = 50$. Each data point was computed by averaging 10^5 Monte Carlo shots. Error bars are not displayed since they are smaller than the symbol size.

one time step to achieve end-to-end entanglement, all of our proposed policies need at least two. Thus, in the ideal scenario $p = p_s = 1$, swap-asap is two times faster than the rest of our policies. In particular, none of our policies outperform swap-asap for any value of p .

The average end-to-end fidelity increases for higher values of p in the swap-asap and nested policies, as shown in Figure 5.2. However, the fidelity of the nonadjacent policy (and nml for ten nodes) stagnates, and even decreases in some cases, when the p approaches one. The intuition behind this rather strange phenomenon is the following: for low p the nonadjacent policy is almost equivalent to swap-asap, and nodes rarely wait instead of swapping. As p increases, elementary links are generated more frequently. Consequently, the average waiting time before links are swapped decreases, in principle resulting in higher fidelity. Nevertheless, if p becomes too large, the system will often encounter states with large connected components. This will cause some nodes to wait to prevent many adjacent swaps, thus decreasing the av-

erage fidelity. In any case, we observe that under deterministic swaps, swap-asap always performs better than our proposed policies.

5.1.2. Probabilistic swaps

We now consider the case with probabilistic swaps ($p_s = 0.5$). From Figure 5.3, we observe that the situation is radically different from deterministic swaps. To start with, the nonadjacent policy now yields a lower end-to-end delivery time than swap-asap for all values of p . Moreover, the nested and nml policies also show better performance than swap-asap when p is sufficiently large. Notably, the difference between the proposed policies and swap-asap becomes even more evident for a larger number of nodes. For the extreme case of $n = 10$ and $p = 1$, all the proposed policies yield an end-to-end delivery time almost six times smaller than the swap-asap policy. It is also remarkable how whereas the swap-asap policy shows a non-monotonic behaviour on p when n is seven or larger, none of the proposed policies exhibit that feature, as we intended. Also interestingly, nested and nml are slower than nonadjacent for low p , but this tendency is reversed for high p . This is because, in the nonadjacent policy, it is more common that the last action before reaching end-to-end entanglement is to perform swaps in two adjacent nodes. This yields a failure probability of $1 - p_s^2 = 0.75$. However, nml and nested reduce the frequency at which two adjacent swaps are performed in the time step previous to reaching end-to-end entanglement. This effect is particularly relevant for repeater chains with an odd number of nodes, in which there exists only a single central node.

On the other hand, the average end-to-end fidelity does not present any noteworthy difference with respect to the deterministic swaps scenario. Swap-asap still provides significantly higher fidelity than any of the other policies. This is consistent with our expectations, since imposing that some nodes wait before they swap inherently increases the average time that states need to be subject to depolarizing noise while in storage.

5.2. Continuous entanglement distribution

One may a priori guess that the continuous and one-time cases should be strongly correlated, and that optimal policies in the first scenario should perform equally well in the second. However, some additional considerations make this a non-trivial question. If we consider the rate at which two parties can share entanglement, it becomes relevant to consider the “remaining links” (links are still in the chain after the end-to-end link is consumed), since these can also be used to generate the following end-to-end link and so on. In principle, being able to reuse those links should imply faster rates, but we will show that this is not always the case. Moreover, the end-to-end fidelity can be negatively affected as well.

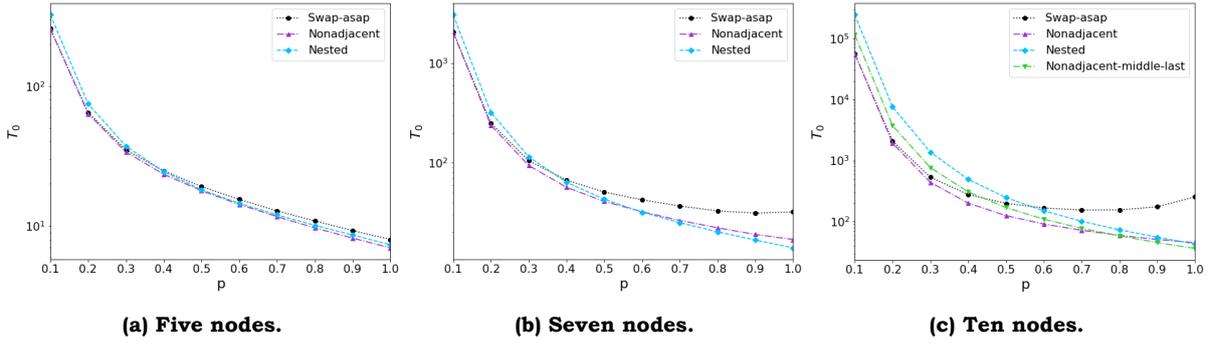


Figure 5.3: Our policies improve the delivery time of swap-asap when swaps are probabilistic and p is large. End-to-end delivery time T_0 (measured in number of time steps) versus entanglement generation probability p when swaps are probabilistic ($p_s = 0.5$) and communication time is negligible. We simulate one-time entanglement generation for repeater chains of 5 (a), 7 (b), and 10 (c) nodes. Each data point was computed by averaging 10^5 Monte Carlo shots. Error bars are not displayed since they are smaller than the symbol size.

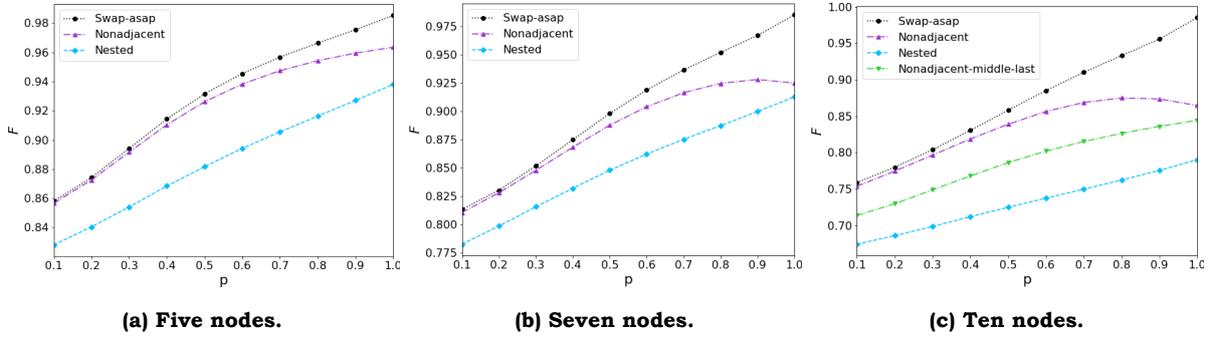


Figure 5.4: The fidelity of swap-asap is higher than that of the other policies when $p_s = 0.5$. End-to-end fidelity F versus entanglement generation probability p when swaps are probabilistic ($p_s = 0.5$) and communication time is negligible. We simulate one-time entanglement generation for repeater chains of five (a), seven (b), and ten (c) nodes with a decoherence rate $\tau = 50$. Each data point was computed by averaging 10^5 Monte Carlo shots. Error bars are not displayed since they are smaller than the symbol size.

5.2.1. Deterministic swaps

Figure 5.5 shows the comparison in terms of fidelity and rate between one-time and continuous entanglement generation for deterministic swaps. We observe that all policies yield higher rates and fidelity as p increases, with swap-asap providing the best results for both performance measures. The only exception is the nonadjacent policy, for which we observe decreases in entanglement rate and end-to-end fidelity when $p = 1$ and $n = 10$. If entanglement generation is deterministic, the average number of waiting nodes significantly increases compared to the case with $p = 0.9$.

Continuous entanglement generation typically yields faster rates than one-time generation, as the “remaining links” have a larger probability of forming new end-to-end links. For small p , nevertheless, those links have a very small probability of reaching end-to-end entanglement. This is detrimental to the rate, since nodes cannot generate elementary links if their qubits are occupied.

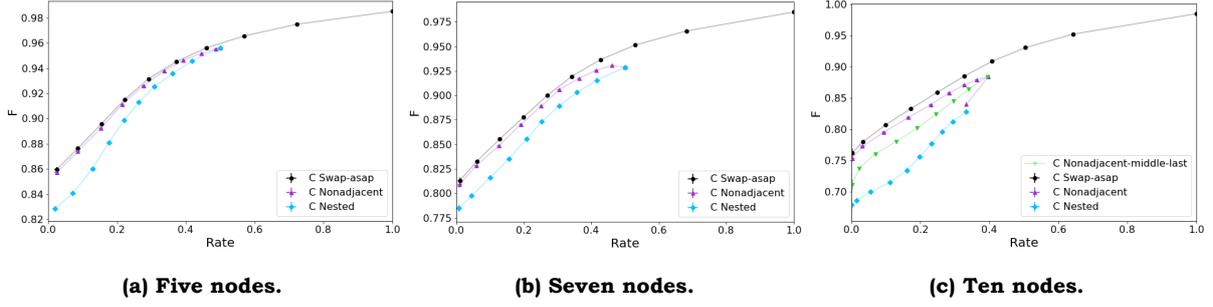


Figure 5.5: Swap-asap shows the best performance when $p_s = 1$. End-to-end fidelity F versus entanglement generation rate (measured in the inverse of number of time steps) when swaps are deterministic and communication time is negligible. We simulate continuous entanglement generation for repeater chains of five (a), seven (b), and ten (c) nodes with a decoherence rate $\tau = 50$. Each data point was computed by averaging 10^6 Monte Carlo shots. Error bars are not displayed since they are smaller than the symbol size. We consider ten equally spaced entanglement generation probability values ranging from 0.1 (leftmost and bottommost data points of each policy) to 1.

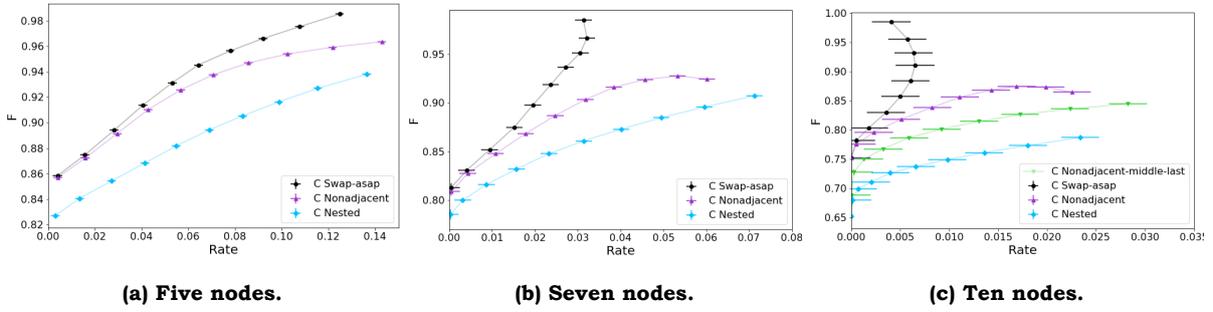


Figure 5.6: Our proposed policies are faster than swap-asap when $p_s = 0.5$ and p is large, but they show a lower fidelity. End-to-end fidelity F versus entanglement generation rate (measured in the inverse of number of time steps) when swaps are probabilistic ($p_s = 0.5$) and communication times are negligible. We simulate continuous entanglement generation for repeater chains of five (a), seven (b), and ten (c) nodes with a decoherence rate $\tau = 50$. Each data point was computed by averaging 10^6 Monte Carlo shots. We consider ten equally spaced entanglement generation probability values ranging from 0.1 (leftmost and bottommost data points of each policy) to 1.

Moreover, reusing pre-existing (and hence older) links to generate end-to-end entanglement generally also negatively affects fidelity. These effects become more evident for long chains, where the number of “remaining links” is higher on average.

5.2.2. Probabilistic Swaps

When swaps are probabilistic, the number of “remaining links” after end-to-end entanglement is achieved is typically very low, so the differences between one-time and continuous entanglement generation become much less significant than for deterministic swaps. From Figure 5.6, we arrive at the conclusions previously discussed in Section 5.1.2: swap-asap produces higher end-to-end fidelity than all of our policies, but these reach lower end-to-end delivery times for large p . For chains with seven or more nodes, swap-asap exhibits

a non-monotonic behaviour in the rate, and nonadjacent is non-monotonic in the fidelity, both with respect to p .

5.3. Non-negligible communication time

In the previous examples, classical communication time did not have a significant contribution to the overall time step length. Nonetheless, in global-scale quantum networks, where distances between nodes are potentially large, classical communication delays ought to be considered. We will focus on the case in which classical communication is the main contribution to the time step length, and study quantum repeater chains under the following conditions:

- The time required for performing cutoffs and swaps is negligible compared to the entanglement generation time. This assumption can be justified if the distance between nodes l is very large, since entanglement generation requires the transmission of heralding signals. On the other hand, cutoffs and swaps are local operations that do not depend on l .
- There is only one entanglement generation attempt per time step ($N_A = 1$).

Then, from Eqs. (3.4) and (3.6):

$$T_{\text{step}}^{\pi} = T_{\text{op}} + T_{\text{comm}}^{\pi} = T_{\text{op}} (1 + (C_1^{\pi} + C_2^{\pi})(n-1)) = T_{\text{op}} \left(1 + \frac{d_{\text{cc}1}^{\pi} + d_{\text{cc}2}^{\pi}}{l} \right). \quad (5.1)$$

Using the expressions for the classical communication distance $d_{\text{cc}i}^{\pi}$ for MK (3.9) and the definition of LK (3.2.1) and GK (3.2.2), we can write the time step length in each of our policies as a function of the number of nodes (recall that swap-asap is LK-GK, nonadjacent is GK-GK, nml is MK-MK and nested is LK-MK):

$$T_{\text{step}}^{\text{LK-GK}} = T_{\text{op}} (n-1), \quad (5.2)$$

$$T_{\text{step}}^{\text{GK-GK}} = T_{\text{op}} (2n-3), \quad (5.3)$$

$$T_{\text{step}}^{\text{MK-MK}} = \begin{cases} T_{\text{op}}(n) & \text{if } n \text{ odd} \\ T_{\text{op}}(1+n) & \text{if } n \text{ even,} \end{cases} \quad (5.4)$$

$$T_{\text{step}}^{\text{LK-MK}} = \begin{cases} T_{\text{op}} \left(1 + \frac{n-1}{2} \right) & \text{if } n \text{ odd} \\ T_{\text{op}} \left(1 + \frac{n}{2} \right) & \text{if } n \text{ even.} \end{cases} \quad (5.5)$$

To compare our policies, we take the time step length of swap-asap (LK-GK) as a baseline and scale the time step length of our candidate policies accordingly. E.g., to find the ratio between GK-GK and LK-GK we divide (5.3) by (5.2). This yields $T_{\text{step}}^{\text{GK-GK}} = \frac{2n-3}{n-1} T_{\text{step}}^{\text{LK-GK}}$. Note that the nested policy (5.5) yields the shortest time steps while nonadjacent (5.3) has the longest. For a fair comparison between all policies, the cutoff must also be scaled (inversely to the time step

Policy	Swap-asap	Nonadjacent	Nonadjacent-middle-last	Nested
Classification	LK-GK	GK-GK	MK-MK	LK-MK
$R_{\text{step}}^\pi(5)$	1	1.25	1.75	0.75
$R_{\text{cut}}^\pi(5)$	1	0.8	0.57	1.33
$R_{\text{step}}^\pi(10)$	1	1.22	1.89	0.67
$R_{\text{cut}}^\pi(10)$	1	0.82	0.53	1.5

Table 5.1: Scaling factors for the time step lengths and cutoff times for the different policies. $R_{\text{step}}^\pi(n)$ and $R_{\text{cut}}^\pi(n)$ are the ratios between the time step lengths and cutoff times of policy π and swap-asap for a chain of n nodes, assuming communication time is the main contribution to the time step length.

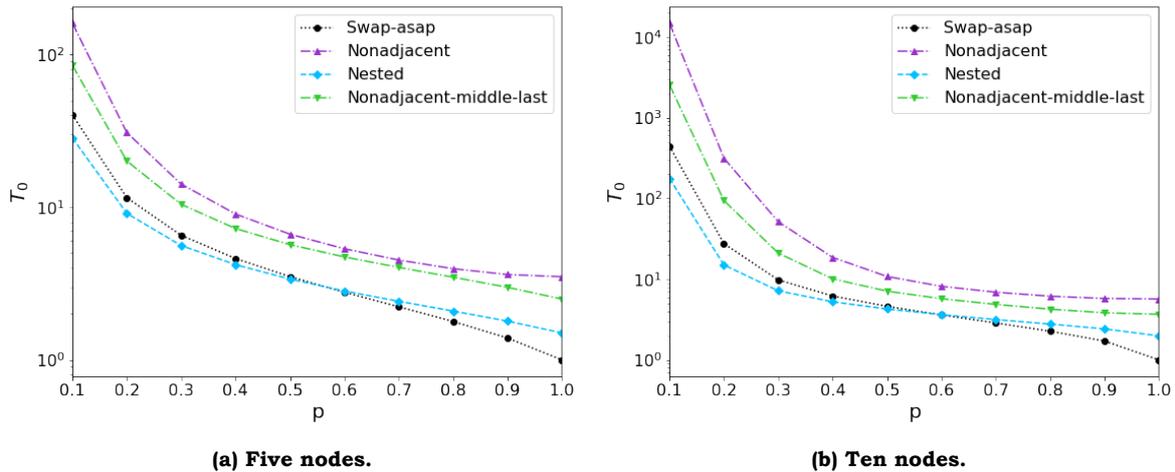


Figure 5.7: Nested is faster than swap-asap when $p_s = 1$ and p is small. End-to-end delivery time T_0 (measured in number of time steps of the swap-asap policy) versus entanglement generation probability p when swaps are deterministic and communication times are the main contribution in the length of the time steps. We simulate one-time entanglement generation for repeater chains of five (a), and ten (b) nodes. Each data point was computed by averaging 10^5 Monte Carlo shots. Error bars are not displayed since they are smaller than the symbol size.

length) to guarantee the same end-to-end fidelity threshold. In our case, we consider repeater chains with five and ten nodes. The time step length and cutoff time scaling for all policies are summarized in Table 5.1.

Figure 5.7 shows the (scaled) end-to-end delivery time for all different policies taking $t_{\text{cut}}^{\text{LK-GK}} = 10$ when $p_s = 1$. Contrarily to the case in which communication time is negligible, our nested policy improves the end-to-end delivery time of swap-asap when p is sufficiently small. This is an important result, as swap-asap is often naively considered the best policy when swaps are deterministic. On the other hand, the nonadjacent and nml policies always provide larger delivery times than swap-asap. For probabilistic swaps, the delivery time of the nested policy is significantly smaller than the rest of policies for all values of p , even being up to 12 times faster than swap-asap for $n = 10$ and $p = 1$ (see Fig. 5.8). For large p and n , all of our policies provide a lower delivery time than the swap-asap policy.

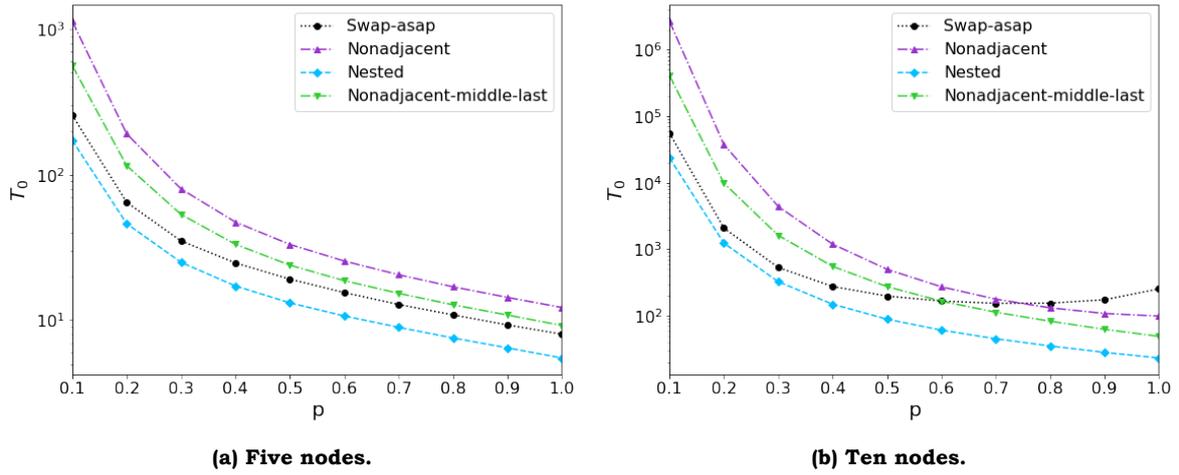


Figure 5.8: Nested is faster than swap-asap when $p_s = 0.5$. End-to-end delivery time T_0 (measured in number of time steps of the swap-asap policy) versus entanglement generation probability p when swaps are probabilistic ($p_s = 0.5$) and communication times are the main contribution in the length of the time steps. We simulate one-time entanglement generation for repeater chains of 5 (a), and 10 (b) nodes. Each data point was computed by averaging 10^5 Monte Carlo shots. Error bars are not displayed since they are smaller than the symbol size.

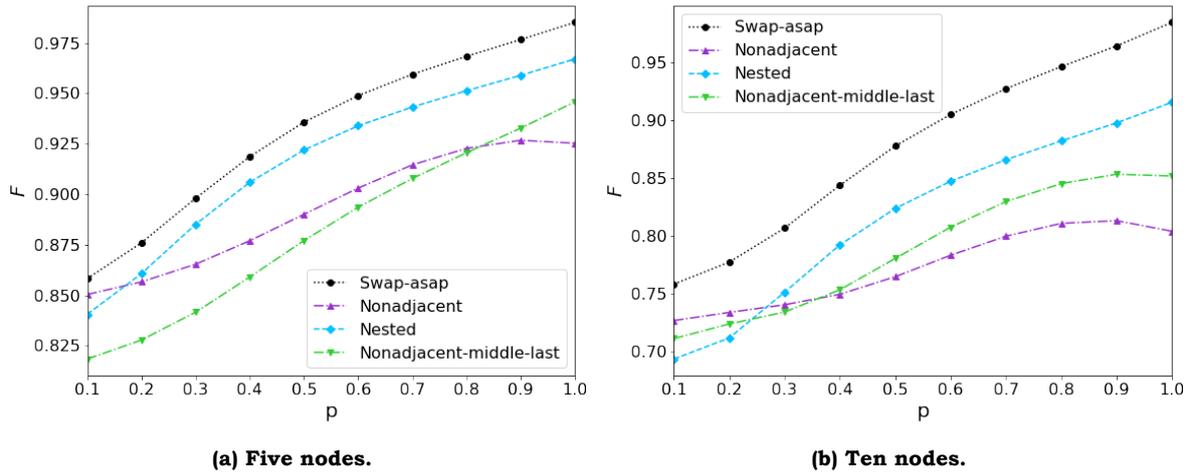


Figure 5.9: The fidelity of swap-asap is higher than that of the other policies when $p_s = 1$. End-to-end fidelity versus entanglement generation probability p when swaps are deterministic and communication times are the main contribution in the length of the time steps. We simulate one-time entanglement generation for repeater chains of 5 (a), and 10 (b) nodes with a decoherence rate $\tau = 50$. Each data point was computed by averaging 10^5 Monte Carlo shots. Error bars are not displayed since they are smaller than the symbol size.

The end-to-end fidelity is also affected when considering classical communication time. Varying the time step length implies that memory qubits will decohere at different rates (in terms of time steps). In policies with shorter time steps, the loss in fidelity in each time step is smaller, but links are allowed to be stored for more time steps. Overall we observe that policies with longer time steps (nonadjacent and nml) show a decrease in fidelity compared to the negligible communication time case, unlike the nested policy. Still,

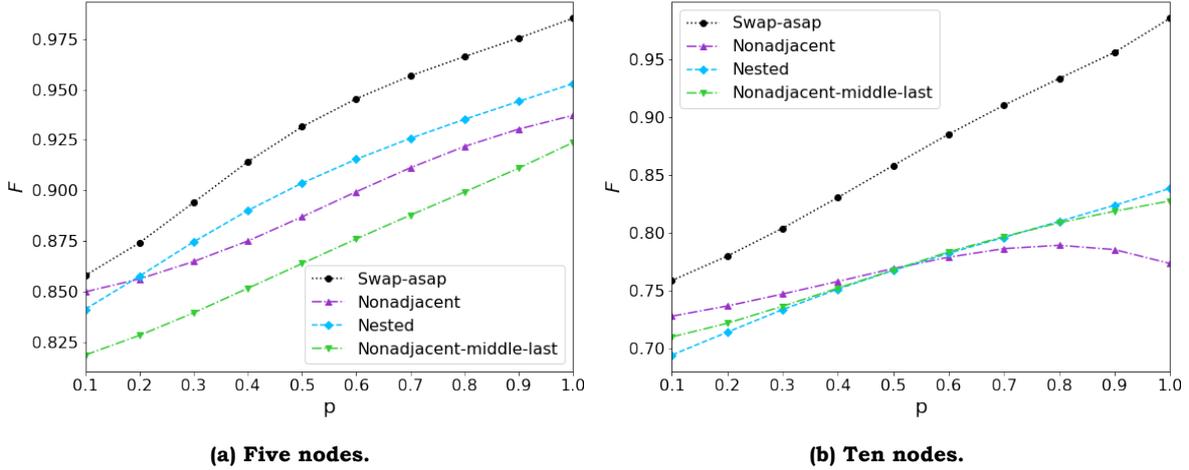


Figure 5.10: The fidelity of swap-asap is higher than that of the other policies when $p_s = 0.5$. End-to-end fidelity versus entanglement generation probability p when swaps are probabilistic ($p_s = 0.5$) and communication times are the main contribution in the length of the time steps. We simulate one-time entanglement generation for repeater chains of 5 (a), and 10 (b) nodes with a decoherence rate $\tau = 50$. Each data point was computed by averaging 10^5 Monte Carlo shots. Error bars are not displayed since they are smaller than the symbol size. Similarly as in the case with deterministic swaps, swap-asap yields the highest end-to-end fidelity for all values of p .

swap-asap remains the policy that better performs in terms of this measure (Figures 5.9 and 5.10).

5.4. Remarks

From our observations, we can conclude that for negligible communication time, our proposed policies produce lower end-to-end delivery times than swap-asap when swaps are probabilistic and p is large. Delaying simultaneous adjacent swaps is only justified if it reduces the probability of losing entanglement. Therefore, our policies do not provide any advantage with respect to swap-asap in terms of delivery time or fidelity for deterministic swaps. However, if communication time is considered, LK-MK policies such as nested are often faster than policies that require global knowledge, like swap-asap or nonadjacent, even when $p_s = 1$.

Our heuristic search was restricted to entanglement swapping policies. To improve the delivery times of swap-asap for $p_s = 1$, small p , and negligible communication time, it may be convenient to consider also different cutoff policies. For example, short and relatively old links have a large probability of being discarded before end-to-end entanglement is achieved. Therefore, imposing dynamic cutoffs according to the length of entangled links may be an alternative to effectively reduce the delivery time. Increasing the fidelity without negatively affecting the delivery time is not always possible, since there exists a trade-off between these two quantities. For instance, the maximum fidelity is obtained if the cutoff time is set to one time step. In this case, swap-asap is trivially the optimal policy. Setting $t_{\text{cut}} = 1$ involves that all elementary

links must be generated simultaneously, and all swaps must be successful. The probability that this occurs in a single time step is $p^{n-1}p_s^{n-2}$, otherwise, all entanglement will be lost. As a consequence, the expected delivery time will be $\frac{1}{p^{n-1}p_s^{n-2}}$. Lastly, increasing the rate can indirectly result in higher fidelities, since entanglement distillation protocols allow to use multiple copies of a noisy entangled state to extract fewer highly entangled states. This is nevertheless out of the scope of this thesis.

Conclusion

In this work, we propose some guidelines to design scalable and easy to implement entanglement distribution policies in quantum repeater chains that aim to reduce the delivery time of the swap-asap policy. For example, we show that the risk of losing entanglement is reduced by avoiding simultaneous adjacent probabilistic swaps. In addition, we introduce classical communication time in our model and demonstrate that in our framework, fully local-knowledge policies (in which nodes only have information about the state of the qubits they hold) do not exist for repeater chains with more than three nodes. We also found that monotonicity is a necessary condition for optimal policies, and that swap-asap does not satisfy it.

We use the end-to-end delivery time and end-to-end fidelity as measures of performance to evaluate the proposed policies (nonadjacent, nonadjacent-middle-last, and nested), considering the cases of one-time and continuous entanglement distribution. Our policies are significantly faster than swap-asap for probabilistic swaps and large entanglement generation probability. This effect is enhanced as the number of nodes in the chain is increased. For deterministic swaps, swap-asap requires on average less number of time steps than our proposed policies. However, if the contribution of classical communication to the total time step length becomes very large, our nested policy is often faster than swap-asap. Delaying swaps typically entails lower fidelity end links, and none of our policies was found to improve over the end-to-end fidelity of swap-asap.

This project also opens future lines of research. We can design more flexible policies by tuning the maximum number of simultaneous adjacent swaps, as well as considering alternative cutoff policies. This way, we could potentially outperform swap-asap over a wider range of hardware parameters. Furthermore, a similar analysis to the one presented in this work can be used to explore non-homogeneous repeater chains, in which distances between nodes (and therefore entanglement generation probabilities and communication time) are not constant. Finally, some of the heuristics discussed in this work could also be extended to networks with different topologies.

Bibliography

- [1] H. J. Kimble. “The quantum internet”. In: *Nature* 453.7198 (June 2008), pp. 1023–1030.
- [2] Artur K. Ekert. “Quantum cryptography based on Bell’s theorem”. In: *Phys. Rev. Lett.* 67 (Aug. 1991), pp. 661–663.
- [3] A Pirker, J Wallnöfer, and W Dür. “Modular architectures for quantum networks”. In: *New Journal of Physics* 20.5 (May 2018), pp. 53–54.
- [4] Tim van Leent et al. “Long-Distance Distribution of Atom-Photon Entanglement at Telecom Wavelength”. In: *Phys. Rev. Lett.* 124 (Jan. 2020), p. 010510.
- [5] Yong Yu et al. “Entanglement of two quantum memories via fibres over dozens of kilometres”. In: *Nature* 578.7794 (Feb. 2020), pp. 240–245.
- [6] Xiao-Min Hu et al. “Long-Distance Entanglement Purification for Quantum Communication”. In: *Phys. Rev. Lett.* 126 (Jan. 2021), p. 010503.
- [7] Stephanie Wehner, David Elkouss, and Ronald Hanson. “Quantum internet: A vision for the road ahead”. In: *Science* 362.6412 (2018), eaam9288.
- [8] William K Wootters and Wojciech H Zurek. “A single quantum cannot be cloned”. In: *Nature* 299.5886 (1982), pp. 802–803.
- [9] Ming-Jun Li and Tetsuya Hayashi. “Advances in low-loss, large-area, and multicore fibers”. In: *Optical Fiber Telecommunications VII*. Elsevier, 2020, pp. 3–50.
- [10] Yoshiaki Tamura et al. “Lowest-ever 0.1419-dB/km loss optical fiber”. In: *Optical fiber communication Conference*. Optica Publishing Group, 2017, Th5D-1.
- [11] K. Boone et al. “Entanglement over global distances via quantum repeaters with satellite links”. In: *Phys. Rev. A* 91 (May 2015), p. 052325.
- [12] Ji-Gang Ren et al. “Ground-to-satellite quantum teleportation”. In: *Nature* 549.7670 (Aug. 2017), pp. 70–73.
- [13] H.-J. Briegel et al. “Quantum Repeaters: The Role of Imperfect Local Operations in Quantum Communication”. In: *Phys. Rev. Lett.* 81 (Dec. 1998), pp. 5932–5935.
- [14] Jian-Wei Pan et al. “Experimental Entanglement Swapping: Entangling Photons That Never Interacted”. In: *Phys. Rev. Lett.* 80 (May 1998), pp. 3891–3894.
- [15] Peter C. Humphreys et al. “Deterministic delivery of remote entanglement on a quantum network”. In: *Nature* 558.7709 (June 2018), pp. 268–273.

- [16] M. Pompili et al. “Realization of a multinode quantum network of remote solid-state qubits”. In: *Science* 372.6539 (Apr. 2021), pp. 259–264.
- [17] Carlo Liorni, Hermann Kampermann, and Dagmar Bruß. “Quantum repeaters in space”. In: *New Journal of Physics* 23.5 (May 2021), p. 053021.
- [18] Ryszard Horodecki et al. “Quantum entanglement”. In: *Rev. Mod. Phys.* 81 (June 2009), pp. 865–942.
- [19] David P. DiVincenzo. “The Physical Implementation of Quantum Computation”. In: *Fortschritte der Physik* 48.9–11 (Sept. 2000), pp. 771–783.
- [20] David L Moehring et al. “Entanglement of single-atom quantum bits at a distance”. In: *Nature* 449.7158 (2007), pp. 68–71.
- [21] Hannes Bernien et al. “Heralded entanglement between solid-state qubits separated by three metres”. In: *Nature* 497.7447 (2013), pp. 86–90.
- [22] Robert Stockill et al. “Phase-tuned entangled state generation between distant spin qubits”. In: *Physical review letters* 119.1 (2017), p. 010503.
- [23] Aymeric Delteil et al. “Generation of heralded entanglement between distant hole spins”. In: *Nature Physics* 12.3 (2016), pp. 218–223.
- [24] Stephan Ritter et al. “An elementary quantum network of single atoms in optical cavities”. In: *Nature* 484.7393 (2012), pp. 195–200.
- [25] Julian Hofmann et al. “Heralded entanglement between widely separated atoms”. In: *Science* 337.6090 (2012), pp. 72–75.
- [26] Hans K. C. Beukers et al. *Tutorial: Remote entanglement protocols for stationary qubits with photonic interfaces*. 2023. arXiv: [2310.19878](https://arxiv.org/abs/2310.19878) [quant-ph].
- [27] Norbert Lütkenhaus, John Calsamiglia, and K-A Suominen. “Bell measurements for teleportation”. In: *Physical Review A* 59.5 (1999), p. 3295.
- [28] John Calsamiglia and Norbert Lütkenhaus. “Maximum efficiency of a linear-optical Bell-state analyzer”. In: *Applied Physics B* 72 (2001), pp. 67–71.
- [29] Matthias J Bayerbach et al. “Bell-state measurement exceeding 50% success probability with linear optics”. In: *Science Advances* 9.32 (2023), eadf4080.
- [30] W. P. Grice. “Arbitrarily complete Bell-state measurement using only linear optical elements”. In: *Phys. Rev. A* 84 (Oct. 2011), p. 042331.
- [31] Álvaro G. Iñesta et al. “Optimal entanglement distribution policies in homogeneous repeater chains with cutoffs”. In: *npj Quantum Information* 9.1 (May 2023).
- [32] V Krutyanskiy et al. “Entanglement of trapped-ion qubits separated by 230 meters”. In: *Physical Review Letters* 130.5 (2023), p. 050803.
- [33] Michael J Biercuk et al. “Optimized dynamical decoupling in a model quantum memory”. In: *Nature* 458.7241 (2009), pp. 996–1000.
- [34] Ye Wang et al. “Single-qubit quantum memory exceeding ten-minute coherence time”. In: *Nature Photonics* 11.10 (2017), pp. 646–650.

- [35] Wolfgang Dür et al. “Standard forms of noisy quantum operations via depolarization”. In: *Physical Review A* 72.5 (2005), p. 052326.
- [36] Reinhard F Werner. “Quantum states with Einstein-Podolsky-Rosen correlations admitting a hidden-variable model”. In: *Physical Review A* 40.8 (1989), p. 4277.
- [37] Can M. Knaut et al. *Entanglement of Nanophotonic Quantum Memory Nodes in a Telecommunication Network*. 2023. arXiv: [2310.01316 \[quant-ph\]](#).
- [38] Bethany Davies et al. “Tools for the analysis of quantum protocols requiring state generation within a time window”. In: *IEEE Transactions on Quantum Engineering* (2024).
- [39] Yair Censor. “Pareto optimality in multiobjective problems”. In: *Applied Mathematics and Optimization* 4.1 (1977), pp. 41–59.
- [40] Martin L Puterman. “Markov decision processes”. In: *Handbooks in operations research and management science* 2 (1990), pp. 331–434.
- [41] Evgeny Shchukin and Peter van Loock. “Optimal Entanglement Swapping in Quantum Repeaters”. In: *Phys. Rev. Lett.* 128 (Apr. 2022), p. 150502.
- [42] Simon D. Reiß and Peter van Loock. “Deep reinforcement learning for key distribution based on quantum repeaters”. In: *Phys. Rev. A* 108 (July 2023), p. 012406.
- [43] Stav Haldar et al. *Fast and reliable entanglement distribution with quantum repeaters: principles for improving protocols using reinforcement learning*. 2023. arXiv: [2303.00777 \[quant-ph\]](#).
- [44] Stav Haldar et al. *Policies for multiplexed quantum repeaters: theory and practical performance analysis*. 2024. arXiv: [2401.13168 \[quant-ph\]](#).
- [45] Matthijs TJ Spaan. “Partially observable Markov decision processes”. In: *Reinforcement learning: State-of-the-art*. Springer, 2012, pp. 387–414.
- [46] Satinder P. Singh, Tommi Jaakkola, and Michael I. Jordan. “Learning Without State-Estimation in Partially Observable Markovian Decision Processes”. In: *Machine Learning Proceedings 1994*. Ed. by William W. Cohen and Haym Hirsh. San Francisco (CA): Morgan Kaufmann, 1994, pp. 284–292.
- [47] Hui Zhou, Tao Li, Keyu Xia, et al. “Parallel and heralded multiqubit entanglement generation for quantum networks”. In: *Physical Review A* 107.2 (2023), p. 022428.
- [48] Charles H Bennett et al. “Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels”. In: *Physical review letters* 70.13 (1993), p. 1895.
- [49] Tim Coopmans et al. “NetSquid, a NETWORK Simulator for QUANTUM Information using Discrete events”. In: *Communications Physics* 4.1 (July 2021).

-
- [50] Adrià Labay-Mora, Francisco Ferreira da Silva, and Stephanie Wehner. *Reducing hardware requirements for entanglement distribution via joint hardware-protocol optimization*. 2023. arXiv: [2309.11448](https://arxiv.org/abs/2309.11448) [quant-ph].
 - [51] John Clark and Derek Allan Holton. *A first look at graph theory*. World Scientific, 1991.
 - [52] W. Dür et al. “Quantum repeaters based on entanglement purification”. In: *Phys. Rev. A* 59 (Jan. 1999), pp. 169–181.
 - [53] William J Munro et al. “Inside quantum repeaters”. In: *IEEE Journal of Selected Topics in Quantum Electronics* 21.3 (2015), pp. 78–90.
 - [54] W Keith Hastings. “Monte Carlo sampling methods using Markov chains and their applications”. In: (1970).
 - [55] Álvaro G Iñesta and Stephanie Wehner. “Performance metrics for the continuous distribution of entanglement in multiuser quantum networks”. In: *Physical Review A* 108.5 (2023), p. 052615.

Link post-swapping age

In this Appendix we show that the post-swapping age of two Werner states subject to a depolarizing process can be computed as

$$t_{\text{swap}} = t_1 + t_2 - \tau \ln \left(\frac{4F_0 - 1}{3} \right), \quad (\text{A.1})$$

where t_1 and t_2 are the ages of the involved links in the swap, τ is the depolarizing rate introduced by noisy memories and F_0 is the fidelity of newly generated links. In [31], authors show that the fidelity of a Werner state experiencing a depolarizing process evolves as

$$F(t) = \frac{1}{4} + \left(F(t - \Delta t) - \frac{1}{4} \right) e^{-\frac{\Delta t}{\tau}}, \quad (\text{A.2})$$

where Δt is an arbitrary interval of time. In particular, we can set $\Delta t = t$, so from (A.2) we can write

$$F(t) = \frac{1}{4} + \left(F_0 - \frac{1}{4} \right) e^{-\frac{t}{\tau}}. \quad (\text{A.3})$$

Therefore the relation between the age of a link and its fidelity can be found in (A.4):

$$t = -\tau \ln \left(\frac{4F(t) - 1}{4F_0 - 1} \right). \quad (\text{A.4})$$

Afer swapping two Werner states with fidelities F_1 and F_2 , the resulting link is a Werner state with fidelity [53]

$$F_{\text{swap}}(F_1, F_2) = F_1 \cdot F_2 + \frac{1}{3} ((1 - F_1)(1 - F_2)). \quad (\text{A.5})$$

Then, inserting (A.3) into (A.5):

$$F_{\text{swap}}(t_1, t_2) = \frac{1}{16} \left(\left(1 + (4F_0 - 1) e^{-\frac{t_1}{\tau}} \right) \left(1 + (4F_0 - 1) e^{-\frac{t_2}{\tau}} \right) + \frac{1}{3} \left(3 - (4F_0 - 1) e^{-\frac{t_1}{\tau}} \right) \left(3 - (4F_0 - 1) e^{-\frac{t_2}{\tau}} \right) \right) \quad (\text{A.6})$$

$$= \frac{1}{4} \left(1 + \frac{(4F_0 - 1)^2}{3} e^{-\frac{(t_1+t_2)}{\tau}} \right). \quad (\text{A.7})$$

Finally, from the post-swapping fidelity we can find the post-swapping age of a link by using (A.4)

$$t_{\text{swap}}(t_1, t_2) = -\tau \ln \left(\frac{(4F_0 - 1)^2 e^{-\frac{(t_1+t_2)}{\tau}}}{3(4F_0 - 1)} \right) = t_1 + t_2 - \tau \ln \left(\frac{4F_0 - 1}{3} \right). \quad (\text{A.8})$$

End-to-end delivery time of middle-last

Here we demonstrate the equations for the single-shot end-to-end delivery time of the middle-last policy assuming $p = 1$ and infinite cutoff time:

$$T_{s_0}^{\text{middle}} = \frac{3 - p_s^{n-3}}{2p_s^{\frac{n-1}{2}} - p_s^{n-2}} \quad \text{if } n \text{ odd.} \quad (\text{B.1})$$

$$T_{s_0}^{\text{middle}} = \frac{1 + 2p_s + p_s^{\frac{n-4}{2}} - p_s^{\frac{n-2}{2}} - p_s^{n-4} + p_s^{3n-11} - p_s^{\frac{3n-12}{2}}}{2p_s^{\frac{n}{2}} - p_s^{n-2} - p_s^{\frac{3n-4}{2}} + p_s^{\frac{3n-2}{2}}} \quad \text{if } n \text{ even.} \quad (\text{B.2})$$

We will use the Bellman equations (B.3) to express the delivery time of the empty chain in terms of the rest of states.

$$T_s^\pi = 1 + \sum_{s' \in \mathcal{S}} P(s'|s, \pi) \cdot T_{s'}^\pi. \quad (\text{B.3})$$

Here T_s^π is the average number of time steps needed to achieve end-to-end entanglement starting from state s under policy π , \mathcal{S} is the state space, and $P(s'|s, \pi)$ is the transition probability of reaching state s from state s' when following policy π . For the rest of the discussion, we will omit the superscript π as we will always consider the middle-last policy.

A significant number of states can be removed from the Bellman equations taking into account the following considerations. Firstly, link ages do not play a role since the cutoff is infinite. Moreover, since $p = 1$, states that differ only in the existence of elementary links are equivalent. Middle-last is a symmetric policy, so the delivery time of a state and its mirrored version must be the same. Then, we will only consider half of the non-symmetric states.

B.1. Odd n

We first consider an odd number of nodes. Figure B.1 shows the four possible different states in the system. Given that $p = 1$, all nodes except the central one will perform swaps in the first time step. If all swaps are successful, the system will reach state s_1 . State s_2 will be reached instead if all of the swaps on one of the sides of the chain are successful, but at least one swap on the opposite side fails. Finally, if swaps fail at both sides of the chain, the system returns to state s_0 . The end-to-end delivery time can be computed as:

$$T_{s_0} = 1 + p_s^{2m} T_{s_1} + 2p_s^m (1 - p_s^m) T_{s_2} + (1 - p_s^m)^2 T_{s_0}, \quad (\text{B.4})$$

where we have introduced the variable $m = \frac{n-3}{2}$ for clarity. Similarly, we can find the equations for the end-to-end delivery time of the rest of states. In

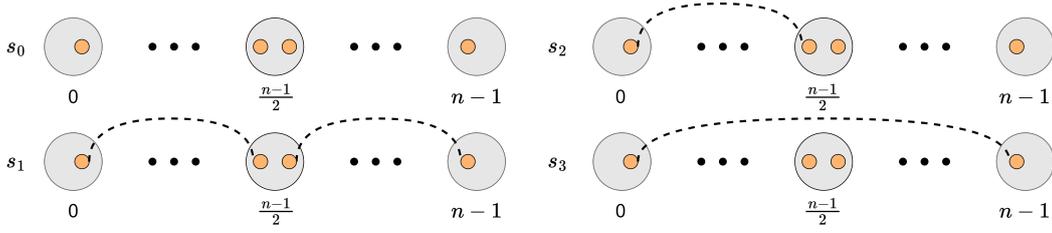


Figure B.1: States in the middle-last policy when n is odd, $p = 1$, and $t_{\text{cut}} = \infty$.

(B.5) we have used that $T_{s_3} = 0$ since it is already an end-to-end entangled state.

$$T_{s_1} = 1 + p_s T_{s_3} + (1 - p_s) T_{s_0} = 1 + (1 - p_s) T_{s_0}. \quad (\text{B.5})$$

$$T_{s_2} = 1 + p_s^m T_{s_1} + (1 - p_s^m) T_{s_2}. \quad (\text{B.6})$$

Solving (B.6) yields

$$T_{s_2} = \frac{1}{p_s^m} + T_{s_1}. \quad (\text{B.7})$$

Substituting Eqs. (B.5) and (B.6) in (B.4)

$$T_{s_0} = 1 + p_s^{2m} + (1 - p_s) p_s^{2m} T_{s_0} + 2(1 - p_s^m) + 2p_s^m(1 - p_s^m) + 2p_s^m(1 - p_s^m)(1 - p_s) T_{s_0} + (1 - p_s^m)^2 T_{s_0} \quad (\text{B.8})$$

Finally, after simplifying we reach (B.9), from which we retrieve (B.1) replacing m by $\frac{n-3}{2}$.

$$T_{s_0} = \frac{3 - p_s^{2m}}{2p_s^{m+1} - p_s^{2m+1}}. \quad (\text{B.9})$$

B.2. Even n

The case of an even number of nodes is equivalent to the previous scenario, but we must now consider that there are six possible states (See Fig. B.2). The central nodes (indexes $\frac{n}{2} - 1$ and $\frac{n}{2}$) will only attempt swaps if they share an entangled link with their respective closer end node. We now set $m = \frac{n}{2} - 2$ to simplify our expressions. Starting from an empty chain, the system will reach state s_0 if all $2m$ swaps (m on every side of the chain) are successful. If all swaps are successful on one side, but some fail on the other side, the system will reach state s_2 . Finally, if at least one swap fails on each side, the system goes back to s_0 . Then, the end-to-end delivery time reads

$$T_{s_0} = 1 + p_s^{2m} T_{s_1} + 2p_s^m(1 - p_s^m) T_{s_2} + (1 - p_s^m)^2 T_{s_0}. \quad (\text{B.10})$$

The equations for the delivery time of the different states can be found similarly:

$$T_{s_1} = 1 + p_s^2 T_{s_5} + (1 - p_s^2) T_{s_0} = 1 + (1 - p_s^2) T_{s_0}, \quad (\text{B.11})$$

where we have used $T_{s_5} = 0$.

$$T_{s_2} = 1 + p_s^{m+1} T_{s_3} + p_s^m(1 - p_s) T_{s_2} + p_s(1 - p_s^m) T_{s_4} + (1 - p_s)(1 - p_s^m) T_{s_0}. \quad (\text{B.12})$$

Equations for s_3 and s_4 are similar to Eqs. (B.11) and (B.12).

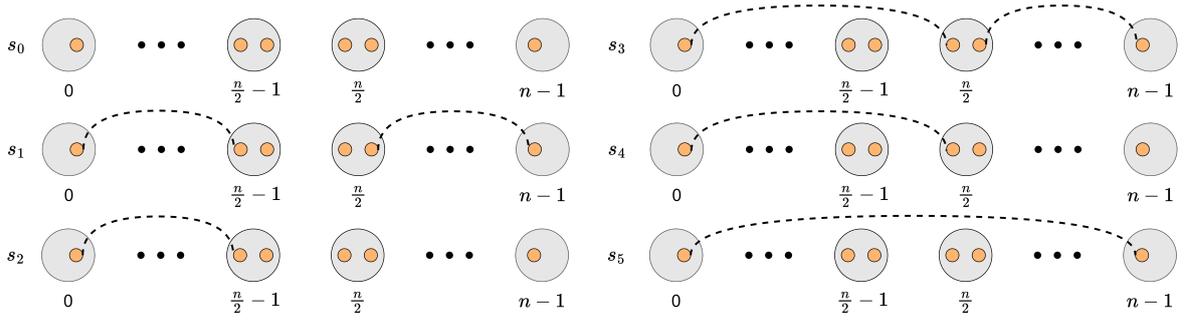


Figure B.2: States in the middle-last policy when n is even, $p = 1$, and $t_{\text{cut}} = \infty$.

$$T_{s_3} = 1 + p_s T_{s_5} + (1 - p_s) T_{s_0} = 1 + (1 - p_s) T_{s_0}. \quad (\text{B.13})$$

$$T_{s_4} = 1 + p_s^m T_{s_3} + (1 - p_s^m) T_{s_4}. \quad (\text{B.14})$$

Finally, substituting in (B.10) and simplifying, we find (B.15).

$$T_{s_0} = \frac{1 + 2p_s + p_s^m - p_s^{m+1} - p_s^{2m}}{2p_s^{m+2} - p_s^{2m+2} - p_s^{3m+2} + p_s^{3m+3}} = \frac{p_s^{3m+1} - p_s^{3m}}{2p_s^{m+2} - p_s^{2m+2} - p_s^{3m+2} + p_s^{3m+3}}. \quad (\text{B.15})$$

Again, this is equivalent to (B.2) if we replace m by $\frac{n}{2} - 2$.

Nested vector for ten nodes

In this Appendix, we derive the equation for the nested vector of a ten-node chain following the algorithm described in Section 4.3.

1. Ten is an even number so we define the label of the central nodes as

$$m_1 = \frac{n}{2} - 1 = 4. \quad (\text{C.1})$$

$$m_2 = \frac{n}{2} = 5. \quad (\text{C.2})$$

2. Set $\nu_{m_1}(n) = Lm_2$, $\nu_{m_2}(n) = Rm_2$.

$$\nu_4(10) = L5. \quad (\text{C.3})$$

$$\nu_5(10) = R5. \quad (\text{C.4})$$

3. Set $\nu_0(n) = \nu_{n-1}(n) = Em_2$.

$$\nu_0(10) = E5. \quad (\text{C.5})$$

$$\nu_9(10) = E5. \quad (\text{C.6})$$

4. Calculate $\vec{\nu}(m_2) = \vec{\nu}(5)$. We must now use the algorithm for an odd number of nodes.

- (a) Define the label of the central node.

$$m = \frac{m_2 - 1}{2} = 2. \quad (\text{C.7})$$

- (b) Set $\nu_m(m_2) = Wm$.

$$\nu_2(5) = W2. \quad (\text{C.8})$$

- (c) Set $\nu_0(m_2) = \nu_{m_2-1}(m_2) = Em$.

$$\nu_0(5) = \nu_4(5) = E2. \quad (\text{C.9})$$

- (d) Calculate $\vec{\nu}(m+1) = \vec{\nu}(3)$. This nested vector was defined as

$$\vec{\nu}(3) = [E1, S1, E1]. \quad (\text{C.10})$$

- (e) Set $\nu_i(m_2) = \nu_{m_2-i-1}(m_2) = \nu_i(m+1)$ for $i = 1, \dots, m-1$.

$$\nu_1(5) = \nu_3(5) = S1. \quad (\text{C.11})$$

Therefore, the nested vector $\vec{\nu}(5)$ reads

$$\vec{\nu}(5) = [E2, S1, W2, S1, E2]. \quad (\text{C.12})$$

5. Set $\nu_i(n) = \nu_{n-i-1}(n) = \nu_i(m_2)$ for $i = 1, \dots, m_1 - 1$.

$$\nu_1(10) = \nu_8(10) = S1. \quad (\text{C.13})$$

$$\nu_2(10) = \nu_7(10) = W2. \quad (\text{C.14})$$

$$\nu_3(10) = \nu_6(10) = S2. \quad (\text{C.15})$$

Finally, all the elements of the nested vector $\vec{\nu}(10)$ are known. The full expression is given by (C.16).

$$\vec{\nu}(10) = [E5, S1, W2, S1, L5, R5, S1, W2, S1, E5]. \quad (\text{C.16})$$

Simulation methods

Analytic analysis of our repeater chain model is only possible under very specific assumptions, so we use simulations to benchmark our entanglement distribution policy candidates against the swap-asap policy. Here we outline the implementation of these simulation methods for one-time and continuous end-to-end entanglement delivery.

In order to estimate the main quantities of interest (end-to-end delivery time and end-to-end fidelity), we employ Monte Carlo (MC) simulations. These take advantage of the concept of random sampling to model the probability distribution of a given event. MC sampling methods often use Markov chains [54]. This is especially suited for our repeater chain model, which can be formulated as a Markov decision process (MDP). Within each time step, elementary links are generated with probability p , swaps are attempted following the specific policy (and are successful with probability p_s) and links are discarded if they exceed the cutoff time t_{cut} . At the end of each time step, the ages of all links are increased by one. Note that for applying cutoffs, ages are calculated using the rule (D.1)

$$t_{\text{swap}} = \max(t_1, t_2), \quad (\text{D.1})$$

where t_{swap} is the age of a link resulting from a swapping operation and t_1 and t_2 are the ages of the links involved in the swap. This rule is useful to compute the value of the cutoff that ensures a minimum value of the end-to-end fidelity F_{min} , but it does not provide an accurate description of the evolution of the fidelity. Therefore, we parallelly keep track of the link ages using the post-swapping formula derived in Appendix A to compute the end-to-end fidelity. It is also possible to directly keep track of the fidelities or the Werner parameters of the states, which would simplify the expressions for the post-swapping fidelity. However, this would involve additional calculations to introduce decoherence experienced by the qubits in each time step (ages simply increase their value by one).

$$t_{\text{swap}} = t_1 + t_2 - \tau \ln \left(\frac{4F_0 - 1}{3} \right) \quad (\text{D.2})$$

To reduce the number of parameters in our system we assumed ideal fidelity of elementary links ($F_0 = 1$). In this case, the post-swapping age is simply $t_{\text{swap}} = t_1 + t_2$. We also arbitrarily set $\tau = 50$.

We distinguish between one-time and continuous entanglement generation (defined in Section 2.3). For one-time entanglement delivery, the system

evolves starting from an empty chain until it reaches end-to-end entanglement. Then, the number of elapsed time steps and the fidelity of the end-to-end link are stored. This procedure is repeated until a sufficient number of end-to-end entanglement episodes ($N_{\text{one-time}}$) have occurred.

In continuous entanglement delivery simulations, the system evolves a total number of t_{max} time steps. When end-to-end entanglement is achieved, the fidelity of the end-to-end link is stored, and the number of counts is increased by one. The end-to-end entanglement delivery rate $R(t)$ can then be estimated as the ratio between the number of counts at time step t , c_{e2e} and the total number of realizations N_{cont} :

$$\hat{R}(t) = \frac{c_{e2e}(t)}{N_{\text{cont}}} \quad (\text{D.3})$$

$R(t)$ does in principle depend on time, but it is expected that after a sufficiently large number of time steps, the system is no longer affected by its initial conditions, so the rate should converge to a constant value, as has been previously shown for similar systems [55]. This “long term” regime is precisely the region of interest, as it captures the dynamics of the system after it has stabilized. We set $t_{\text{max}} = 1000$ and confirm from the simulation results that the system always reaches equilibrium in this time scale.

The error in our estimates is inversely proportional to the square root of the number of samples in our simulations.

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{N}} \approx \frac{\sigma_x}{\sqrt{N}}, \quad (\text{D.4})$$

where $\sigma_{\bar{x}}$ is the standard error of the sample mean of the random variable x , σ is the standard deviation of the population, σ_x is the sample standard deviation, and N is the total number of samples. Therefore, there exists a trade-off between reliability and use of computational resources. Our choice for the number of samples was to set $N_{\text{one-time}} = 10^5$ and $N_{\text{cont}} = 10^6$. The standard error can be used to find confidence intervals for our estimates. The z-score corresponding to a confidence interval of 95% is 1.96. That is, with a 95% certainty, the mean value of the population will be in the interval $(\bar{x} - 1.96\sigma_{\bar{x}}, \bar{x} + 1.96\sigma_{\bar{x}})$. The error bars in our figures are also calculated using this z-score.

The Python simulations employed in this project can be found in the following GitHub repository:

<https://github.com/hectorcaleromas/Homogenous-Repeater-Chains>.