Sharing entanglement efficiently

Protocols and architectures for quantum networks

Iñesta, Á. G.

DOI
10.4233/uuid:b4e32ae6-93a2-46c2-899d-d5304e02fa5f

Publication date
2025

Document Version
Final published version

Citation (APA)
Iñesta, Á. G. (2025). *Sharing entanglement efficiently: Protocols and architectures for quantum networks*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:b4e32ae6-93a2-46c2-899d-d5304e02fa5f

Sharing Entanglement Efficiently

Protocols and Architectures for Quantum Networks

Álvaro G. Iñesta

Sharing Entanglement Efficiently: Protocols and Architectures for Quantum Networks

SHARING ENTANGLEMENT EFFICIENTLY: PROTOCOLS AND ARCHITECTURES FOR QUANTUM NETWORKS

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen
op 26 mei 2025 om 10:00 uur

door

Álvaro G. IÑESTA

Master of Science in Applied Physics,
Technische Universiteit Delft, Nederland,
geboren te Novelda, Spanje.

Dit proefschrift is goedgekeurd door de promotoren.

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. S. D. C. Wehner	Technische Universiteit Delft, promotor
Prof. dr. ir. R. Hanson	Technische Universiteit Delft, promotor
Onafhankelijke leden:	
Dr. M. Blaauboer	Technische Universiteit Delft
Prof. dr. E. Diamanti	Sorbonne Université, CNRS
Prof. dr. ir. L. M. K. Vandersypen	Technische Universiteit Delft
Prof. dr. ir. F. A. Kuipers	Technische Universiteit Delft
Prof. dr. G. A. Steele	Technische Universiteit Delft, reservelid

Printed by: Ridderprint | www.ridderprint.nl

Cover: Á. G. Iñesta

ISBN 978-94-6522-317-9

An electronic version of this dissertation is available at
https://repository.tudelft.nl/.

To those who leave coffee-stained paw prints.

CONTENTS

SUMMARY

Quantum networks are expected to enable applications that are provably impossible with classical communication alone, such as generation of secret keys for secure communication and high-precision distributed sensing. A fundamental resource needed for many of these applications is **shared entanglement** among distant parties. Hence, the viability of an application relies on the underlying protocol for entanglement distribution. Existing protocols often suffer from long waiting times, as they rely on the success of multiple random events, each with a low probability of success. Moreover, pre-distribution of entanglement is difficult, since entanglement degrades over time when stored in memory, eventually becoming unusable. In this thesis, we address these challenges by designing efficient entanglement distribution protocols and architectures.

First, we focus on **on-demand entanglement distribution**, in which the entanglement distribution process is initiated only after some users request it. We find optimal protocols that minimize the waiting time for distributing entanglement among two users that are connected by a chain of two-way quantum repeaters. The performance of these protocols sets a benchmark for on-demand distribution of quantum states. We also study a multi-user network of one-way quantum repeaters, and we conclude that finite waiting times are only achievable when the users are at most a few kilometers apart from each other, irrespective of the number of repeaters available.

Next, we examine protocols for **continuous entanglement distribution**, in which the distribution process is initiated before any user requests. While these protocols can sometimes lead to resource wastage – as noise in memory renders the entanglement unusable if distributed too early –, they offer the potential to reduce expected waiting times compared to on-demand methods. Surprisingly, we find that, when the time required to distribute entanglement follows a broad probability distribution, initiating the process preemptively can actually result in longer expected waiting times compared to an on-demand approach.

Lastly, we propose an architecture for **buffering high-quality entanglement**, ensuring it is readily available for use when needed. A key feature of this system is the use of purification subroutines to prevent the buffered entanglement from degrading over time due to quantum decoherence. Among other findings, we show that maximizing entanglement quality upon consumption requires frequent purification, even if this process often fails and results in the loss of high-quality buffered entanglement.

The results presented in this dissertation were obtained mostly analytically, leveraging tools from performance analysis, including queueing theory and renewal theory, and supported by extensive discrete-event simulations. Our theoretical insights provide benchmarks and identify fundamental limitations of quantum networks, offering valuable guidance for the design of reliable entanglement distribution systems.

SAMENVATTING

Kwantumnetwerken zullen naar verwachting toepassingen mogelijk maken die bewezen onmogelijk zijn met alleen klassieke communicatie, zoals het genereren van geheime sleutels voor veilige communicatie en zeer nauwkeurige gedistribueerde detectie. Een essentiële hulpbron voor veel van deze toepassingen is **gedeelde verstrengeling** tussen partijen die ver van elkaar verwijderd zijn. De levensvatbaarheid van een toepassing hangt daarom af van het protocol voor verstrengelingsdistributie. Bestaande protocollen kampen vaak met lange wachttijden, omdat ze afhankelijk zijn van meerdere willekeurige gebeurtenissen met een lage slagingskans. Daarnaast is pre-distributie van verstrengeling moeilijk, omdat verstrengeling na verloop van tijd degradeert wanneer opgeslagen in geheugen, wat het uiteindelijk onbruikbaar maakt. In dit proefschrift pakken we deze uitdagingen aan door efficiënte verstrengelingsdistributieprotocollen en -architecturen te ontwerpen.

Ten eerste richten we ons op **on-demand verstrengelingsdistributie**, waarbij het proces pas start wanneer gebruikers erom vragen. We ontwikkelen optimale protocollen die de wachttijd minimaliseren voor het distribueren van verstrengeling tussen twee gebruikers, verbonden door een keten van tweerichtingskwantumrepeaters. De prestaties van deze protocollen dienen als referentiepunt voor on-demand distributie van kwantumtoestanden. We onderzoeken ook een multi-user netwerk van eenrichtingskwantumrepeaters en concluderen dat eindige wachttijden alleen haalbaar zijn als gebruikers niet meer dan een paar kilometer van elkaar verwijderd zijn, ongeacht het aantal repeaters.

Vervolgens analyseren we protocollen voor **continue verstrengelingsdistributie**, waarbij het proces start vóórdat gebruikers een verzoek doen. Hoewel dit soms leidt tot verspilling van middelen, omdat ruis in het geheugen verstrengeling onbruikbaar maakt wanneer het te vroeg wordt gedistribueerd, kunnen deze protocollen de gemiddelde wachttijden verkorten ten opzichte van on-demandmethoden. Verrassend genoeg blijkt dat wanneer de distributietijd een brede waarschijnlijkheidsverdeling volgt, preventief starten juist kan leiden tot langere wachttijden dan een on-demand aanpak.

Tot slot stellen we een architectuur voor om **hoogwaardige verstrengeling te bufferen**, zodat deze direct beschikbaar is wanneer nodig. Een belangrijk onderdeel van dit systeem is het gebruik van zuiveringssubroutines om te voorkomen dat de gebufferde verstrengeling degradeert door kwantumdecoherentie. We tonen aan dat het maximaliseren van de verstrengelingskwaliteit bij gebruik frequente zuivering vereist, zelfs als dit proces vaak mislukt en resulteert in het verlies van hoogwaardige gebufferde verstrengeling.

De resultaten in dit proefschrift zijn grotendeels analytisch verkregen, met behulp van hulpmiddelen uit de prestatieanalyse, waaronder wachtrijtheorie en vernieuwingstheorie, en ondersteund door uitgebreide discrete-gebeurtenissimulaties. Onze theoretische inzichten bieden benchmarks, identificeren fundamentele beperkingen van kwantumnetwerken en bieden waardevolle richtlijnen voor het ontwerp van betrouwbare verstrengelingsdistributiesystemen.

PREFACE

Dear reader,

My PhD journey began in November 2020, in the midst of a global pandemic. During the first year, virtual meetings became the norm. Like many researchers, I found myself building my own little island of knowledge, with limited connection to others. It was during these early stages that Kaku arrived. His scars initially made me hesitant to trust him, but he soon proved to be the sweetest (and most hard-working) feline companion I could ask for.

As social restrictions gradually eased, those isolated islands of knowledge started to reconnect, fostering collaboration and renewing a sense of community for researchers like me who were still finding their feet. I was very fortunate to work alongside researchers I deeply respect, many of whom I now consider my friends. This journey also led me on adventures I thoroughly enjoyed, taking me to conferences, workshops, and research visits in cities such as Chicago, Boston, Paris, Las Vegas, Amherst, Montreux, and Montreal. These experiences expanded my horizons, exposing me to new perspectives and sending me back to Delft with fresh and exciting new ideas.

Now, after four years of navigating uncharted territory, I am pleased to bring this journey to its end.

This dissertation presents an extensive summary of my research. Like most scientific works, it is not meant to be read in its entirety in a linear fashion. I encourage you to explore the Table of Contents and focus on the chapters that capture your interest. To offer a gentle introduction for those new to the field of quantum networks, I have included a brief introductory chapter. Additionally, each main chapter is self-contained, offering its own independent introduction for clarity and context.

The results presented in this dissertation address several key questions essential to the design of quantum networks, but they also open the door to new inquiries. As such, I hope that these contributions not only provide valuable insights but also spark further exploration and innovation in the field of quantum networks.

Álvaro G. Iñesta
Delft, November 2024

1

INTRODUCTION

Quantum networks are expected to enable multi-party applications that are impossible, or highly inefficient, by using only classical information. Notable examples include quantum key distribution [12, 59], which allows remote parties to generate secret keys for secure communication; distributed quantum computing [26, 46], where a coalition of small quantum computers can perform tasks that would otherwise require a large and powerful quantum computer; and distributed quantum sensing [74, 152, 202], which can enhance the precision of certain physical measurements. At the core of most of these applications is the use of *shared entanglement as a critical consumable resource*: the parties involved must share entangled states to successfully execute the application.

Two or more quantum systems are said to be entangled when their individual states cannot be described independently of each other [138]. A quantum network must be capable of entangling systems held by parties at remote physical locations, a process known as *entanglement distribution* (or *delivery*). Entanglement is generally distributed using entangled photons [9, 33]. For example, in the case of bipartite entanglement, one party may locally generate an entangled pair of particles and send one of them, as a photon, to the other party. In some situations, the distance between parties is short, spanning only a few meters, such as when quantum illumination is used to enhance the signal-to-noise ratio of an image [61, 123]. In other cases, entanglement may need to be distributed over several kilometers, for example, to improve radar measurements [199, 208], or even across continental distances, as required for long-range quantum key distribution [124, 181]. Despite recent experimental advances in distributing entanglement over both short and long distances [16, 109, 118, 180, 207], efficient entanglement distribution remains an open problem.

A major issue for efficient entanglement distribution is the fragility of quantum states during transmission and storage:

- NOISY COMMUNICATION – Photon loss is a well-known challenge in optical communication networks, hampering the transmission of signals over long distances. In classical networks, optical amplifiers and repeaters are commonly used

to strengthen signals and enable long-distance transmission. However, these solutions cannot be applied to quantum communication, as quantum states cannot be cloned or amplified without altering the information they carry due to the no-cloning theorem [138].

- NOISY STORAGE – When stored in memory, quantum states experience *quantum decoherence*, causing their quality to degrade over time due to environmental noise [38, 58]. Moreover, the no-cloning theorem prevents the creation of backup copies of quantum data, adding another layer of complexity. It is also important to note that local processing of stored quantum states, such as applying quantum gates, often constitutes an additional source of noise.

Noisy communication and storage has major implications in quantum networks, as it leads to important timing constraints and challenges in the entanglement distribution process. Before delving into the specific challenges addressed in this dissertation, we first introduce the key concepts and terminology. Section 1.1 offers a brief introduction to the key quantum information concepts used throughout this thesis. Then, in Section 1.2, we discuss general quantum network principles and establishes the terminology for the following chapters. In Section 1.3, we outline the main analytical and numerical tools we use to study the performance of entanglement distribution protocols in quantum networks. Finally, Section 1.4 presents the key challenges that remained open at the start of our PhD research and explains how they are addressed in this dissertation.

1.1. QUANTUM PRELIMINARIES

Here, we introduce the basic quantum information concepts employed throughout this dissertation. This section is adapted from our previous work [50]. For a general reference on quantum information theory, see, e.g., ref. [138].

In classical computer science, information is generally stored in the form of *bits*, discrete binary variables that can take value zero or one. In quantum information theory, bits are generalized to *qubits*. Qubits describe systems that can be in a linear combination of two different states (say, state zero and state one). As a consequence, a qubit must not be described as a binary variable but as a vector $|\psi\rangle \in \mathbb{C}^2$ with unit norm. We represent these vectors using the Dirac notation, and we refer to $|\cdot\rangle$ as a ket and to its conjugate transpose $|\cdot\rangle^\dagger$ as a bra, also written as $\langle\cdot|$. The state of a *pure* n-qubit system is then described by a d-dimensional vector $|\psi\rangle \in \mathbb{C}^d$, with $d = 2^n$. The elements of the basis of the n-qubit Hilbert space are usually labeled $|\mathbf{x}\rangle$, with $\mathbf{x} \in \{0,1\}^n$. For example, the basis of the two-qubit space can be written as $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. We can modify the states of qubits via quantum gates and measurements. *Quantum gates* are unitary transformations acting on the Hilbert space. Some of them are analogous to logic gates – for example, a bit can be flipped with a NOT gate and a qubit can be flipped from $|0\rangle$ to $|1\rangle$ with a Pauli X gate [138]. *Measurements* change the state of the qubit according to some positive operator-valued measure (POVM), which determines the probability of obtaining each measurement outcome. For example, if we measure a qubit $|\psi\rangle = \sin\theta\,|0\rangle + \cos\theta\,|1\rangle$, $\theta \in [0,\pi]$, in the basis $\{|0\rangle, |1\rangle\}$, the state of the qubit after the measurement will be either $|0\rangle$ or $|1\rangle$. For

formal definitions and further examples of quantum gates and measurements, see, e.g., ref. [138].

Noise and quantum operations, such as *gates* and *measurements*, modify the state of a quantum system. When dealing with noisy systems, it is necessary to represent quantum states using the *density matrix* formalism instead of kets and bras. The density matrix of a *pure* state $|\psi\rangle$ can be written as the outer product $\rho = |\psi\rangle\langle\psi|$. When the system is not pure, it is called *mixed*, and it is written as a mixture of pure states: $\rho = \sum_i \alpha_i |\psi_i\rangle\langle\psi_i|$, with $\alpha_i \in [0,1]$ and $\sum_i \alpha_i = 1$. Intuitively, this corresponds to a system with some degree of uncertainty from a classical point of view. For example, consider a device that prepares the state $|\psi_1\rangle$ with probability α and the state $|\psi_2\rangle$ with probability $1-\alpha$. We can write the output state of the device as a mixed state $\alpha |\psi_1\rangle\langle\psi_1| + (1-\alpha)|\psi_2\rangle\langle\psi_2|$.

Let us now consider a two-qubit state $|\psi\rangle \in \mathbb{C}^4$. When one of the qubits is in state $|\psi\rangle_1 \in \mathbb{C}^2$ and the other qubit is in state $|\psi\rangle_2 \in \mathbb{C}^2$, we can write the joint state of the system as the tensor product of the individual qubit states: $|\psi\rangle = |\psi\rangle_1 \otimes |\psi\rangle_2$. This is called a *product state*. When the two qubits are *entangled*, it is not possible to describe their joint state as a tensor product[1]. One of the intuitive effects of the entanglement is that, if both qubits are measured, the measurement outcomes will be correlated. We refer to a two-qubit (mixed) state ρ as an *entangled link* – in most setups considered in this thesis, both qubits are situated at distant locations, and the entanglement can be regarded as "link" connecting them. Entangled links can be entangled to different degrees. The *Bell states* are examples of two-qubit maximally entangled (pure) states:

$$|\phi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}, \; |\psi^+\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}}, \; |\psi^-\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}, \; |\phi^-\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}}. \tag{1.1}$$

These four states form an orthonormal basis of the two-qubit Hilbert space. Measurements on the individual qubits of any of these states yield maximally correlated outcomes. Additionally, all two-qubit maximally entangled states are equivalent, since they can be mapped via single-qubit operations to each other. Therefore, we can measure the *quality of the entanglement* of a two-qubit state by measuring how close the state is to one of the Bell states, say $|\phi^+\rangle$. Formally, we do this using the *fidelity* of the state:

$$F(\rho) = \langle\phi^+|\rho|\phi^+\rangle, \tag{1.2}$$

where ρ is the density matrix of an arbitrary two-qubit state.

Lastly, an important type of mixed entangled state is the *Bell-diagonal state*:

$$\rho_{\text{BD}} = F_{\text{BD}} |\phi^+\rangle\langle\phi^+| + \lambda_1 |\psi^+\rangle\langle\psi^+| + \lambda_2 |\psi^-\rangle\langle\psi^-| + \lambda_3 |\phi^-\rangle\langle\phi^-|,$$

with $F_{\text{BD}}, \lambda_1, \lambda_2, \lambda_3 \in [0,1]$ subjected to the normalization constraint $F_{\text{BD}} + \lambda_1 + \lambda_2 + \lambda_3 = 1$. The fidelity of this state is F_{BD}. Bell-diagonal states are relevant because any two-qubit state can be transformed into Bell-diagonal form while preserving the fidelity by applying extra noise, a process known as *twirling* [13]. A specific instance of Bell-diagonal state is the Werner state [197]:

$$\rho_{\text{W}} = F|\phi^+\rangle\langle\phi^+| + \frac{1-F}{3}|\psi^+\rangle\langle\psi^+| + \frac{1-F}{3}|\psi^-\rangle\langle\psi^-| + \frac{1-F}{3}|\phi^-\rangle\langle\phi^-|, \tag{1.3}$$

[1] Note that entanglement is a more general concept that not only applies to two-qubit systems, but also to multi-partite quantum states. In this thesis, however, we restrict ourselves to bipartite entanglement unless otherwise specified.

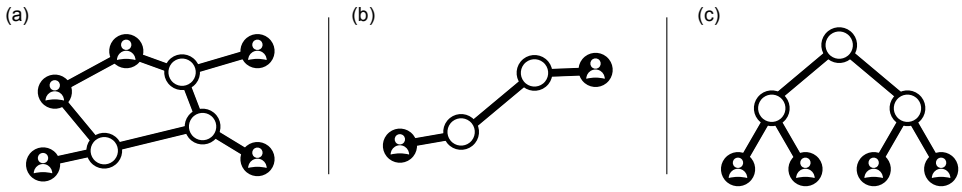

Figure 1.1: **Examples of network topology.** (a) Arbitrary topology, (b) repeater chain, and (c) tree network. Users are connected via repeaters (empty circles) and physical links (solid lines). The quantum repeater chain is investigated in Chapter 2. The quantum tree topology is analyzed in depth in ref. [40] and used in some of the examples in Chapter 4.

with fidelity $F \in [0,1]$. The Werner state corresponds to a maximally entangled state that has been subjected to isotropic noise, and it is entangled if and only if $F > 1/2$ (see Appendix E.3 from ref. [50]). A recurring assumption in this thesis is that entangled links are Werner states – this is a worst-case assumption, since we can apply additional noise to any two-qubit state to transform it into Werner form via twirling [58, 86]. Consequently, we measure the quality of entangled links using a single parameter, the fidelity, unless otherwise specified.

Note that, when distributing entanglement in a network, multiple sources of noise may degrade the fidelity of the quantum states – even when the state is simply stored in memory its fidelity will decay over time [38, 58, 203]. To alleviate the effects of noise, engineering tools such as cutoffs, quantum error correction, and entanglement purification are used. We discuss them at the end of the next section.

1.2. OVERVIEW OF QUANTUM NETWORKS

Quantum networks are networked systems that facilitate quantum communication between multiple parties. Quantum networks are composed of *nodes*, which are connected by *physical links* (see Figure 1.1). These physical links allow for quantum information to be transmitted from node to node, and they can be realized with optical fibers [174, 206] or free space [169, 181]. The nodes of the network can be classified into *users*[2] and *quantum repeaters* (sometimes called *routers* or *switches*, depending on the context). Here, we focus on the distribution of entanglement among user nodes: after entanglement is distributed, each user will hold part of a shared entangled state. **Quantum repeaters** alleviate the effects of photon loss and enable long-distance entanglement distribution. This can be achieved in different ways, depending on the type of repeater employed (for a detailed review see, e.g., refs. [132, 136]):

1. *Two-way quantum repeaters* generate entanglement between users following a two-step process that requires two-way communication (see, e.g., [22, 164]): first, entanglement is generated among each pair of nodes that share a physical link [9, 33], over a path between the two users; then, entanglement swapping is used to fuse

[2]We use the terms *user*, *user node* and *end node* interchangeably throughout this thesis, unless otherwise specified.

short-distance entangled links into long-distance ones [55, 209] – see Figure 1.2 for an illustration. In general, entanglement generation and swapping are probabilistic, i.e., they can fail, destroying all entangled links involved.

2. *One-way quantum repeaters* use quantum error correction to transmit quantum data using only one-way communication [133, 135]. To distribute an entangled link among two users, the entangled state must be generated locally by one user, who then sends half of the state to the other user. This process is conceptually similar to sending a data packet in classical networks [8, 51], with the key distinction that the "packet" now contains quantum information rather than a classical bit string [54]. While one-way quantum repeaters are expected to enable faster entanglement distribution, they demand more physical resources (to construct robust encoded quantum states) and typically require shorter distances between repeaters compared to two-way quantum repeaters.

As of now, quantum networks have not yet been fully implemented for applications that consume entanglement[3]. Therefore, it is hard to predict what would be the **topology** of such networks. Current research usually focuses on the following topologies:

- The *quantum repeater chain* (Fig. 1.1b) is a canonical example of network used to distribute entanglement among two fixed users [22, 44, 91, 96, 102, 167].
- *Tree networks* are structured such that each user is connected to a repeater, with repeaters arranged hierarchically, all connecting to higher-level repeaters, except for the top-most one (Fig. 1.1c). In this topology, any node can be reached from any other node by following a single, unique path, making route optimization straightforward [40].
- A particular instance of tree network that has received a lot of scientific attention is the *star network*, where all users are directly connected to a central repeater station, sometimes called *quantum switch* [6, 89, 179, 187].
- *Regular networks* typically consist of two-dimensional regular arrangements of nodes, such as in a square grid. These topologies are commonly considered in the context of modular quantum computing, where multiple cores are arranged in a regular pattern and require entanglement between them to operate [2, 41, 80, 157, 177].

Regardless of their topology, the ultimate goal of the networks considered here is to distribute entangled links among the users. To achieve this, we must design effective **entanglement distribution protocols**, which dictate how the network operates (e.g., when and where to generate short-distance entanglement and when to perform entanglement swapping). These protocols adhere to one of the following general approaches:

- In *on-demand entanglement distribution*, users must request entanglement to initiate the protocol. Such requests may include some quality-of-service requirements

[3] Networks for quantum key distribution have been deployed, but this application does not necessarily consume entanglement. See, e.g., ref. [34] for a review.

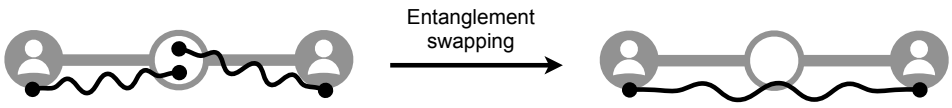

Figure 1.2: **Entanglement swapping.** Two-way quantum repeaters first establish entanglement between neighboring nodes, then apply entanglement swapping to extend this short-range entanglement into long-distance connections.

(e.g., entanglement with fidelity above some threshold). This type of protocol typically involves solving a routing problem and scheduling a series of operations on a selected subset of nodes (see, e.g., [22, 63, 91, 140, 187, 189]). In this dissertation, we focus on protocols for on-demand distribution in Chapters 2 (quantum repeater chain with two users and two-way repeaters) and 3 (star network with multiple users and one-way repeaters).

- Protocols for *continuous distribution of entanglement* are not triggered by user requests. Instead, these protocols distribute entanglement continuously throughout the network, allowing users to consume entangled links whenever needed [35, 94, 105, 146, 149]. This approach enables applications to continuously operate and consume entanglement in the background. For example, a quantum key distribution application could run in the background, continuously consuming entanglement to generate shared key for future use. Continuous-distribution protocols can offer improved performance over on-demand protocols, potentially meeting entanglement needs at a faster rate. However, this comes at the cost of quantum resource wastage: if entanglement is shared and stored for too long, its quality degrades and eventually becomes unusable. Continuous-distribution protocols are investigated in Chapter 4, where we propose metrics to measure their performance. In Chapter 5, we compare continuous and on-demand distribution protocols in terms of both performance and cost (in terms of wasted resources). Additionally, in Chapters 6 and 7, we propose an entanglement buffering system designed to support continuous distribution of entanglement.

Intermediate strategies that combine on-demand and continuous-distribution philosophies may also be viable. For instance, the network could continuously distribute entanglement under normal conditions but switch to on-demand distribution when saturated with specific demands, thereby enhancing efficiency in critical scenarios.

Lastly, it is important to note that quantum network applications commonly require high-quality entanglement. However, quantum operations and storage in quantum memories are generally noisy processes that degrade the quality of the entanglement. To **overcome the loss of quality in entangled links** during their distribution across a network, three fundamental techniques have traditionally been employed:

- *Cutoffs* consist in discarding a quantum state after some condition is met. Usually, this is done after some fixed amount of time since the generation of the quantum state [43, 91, 116, 160].

- *Quantum error correction* involves encoding a quantum state into a higher-dimensional state from a larger Hilbert space – e.g., a single qubit can be encoded into a five-qubit state [112]. This delocalization of quantum information makes the encoded state more robust against noise, and allows for the detection and correction of errors that may occur during transmission or processing. For a general introduction to quantum error correction, see, e.g., ref. [69].

- *Purification protocols* are processes that consume n entangled quantum states of low quality and output m states with a higher quality, where typically $m < n$. Examples can be found in refs. [15, 53, 56]; for a review see, e.g., ref. [205].

1.3. PERFORMANCE ANALYSIS IN QUANTUM NETWORKS

Effective entanglement distribution protocols must deliver high-quality entanglement as quickly as possible. To design such protocols, we need to define meaningful *performance metrics*, which are essential for optimizing individual protocols and enabling fair comparisons between different protocols and architectures.

The choice of performance metric(s) heavily depends on the goal of the system. While the overarching goal is to distribute entanglement, each system may have different specific requirements. Consider the following examples: (*i*) an entanglement-based quantum key distribution application that can only generate secret key when consuming entangled links with fidelity above a certain threshold [165] and (*ii*) a blind quantum computation that requires multiple coexisting entangled links to teleport several qubits simultaneously from a client to a server [49, 115]. In the first example, one may want to ensure that entangled links have a large enough fidelity, and then optimize the rate at which we can provide entanglement. In the second example, the focus might be on ensuring that multiple entangled links can be distributed within a short time frame, followed by an optimization of their fidelities to minimize errors in the teleportation subroutine. As we can see, different types of systems and protocols require different performance metrics. We can classify these metrics in three main categories:

- *Quality-based metrics* evaluate the quality of the quantum states produced. Common examples used to study on-demand protocols include the fidelity of the entangled links delivered [138], the distillable entanglement [153, 154], and the negativity [190]. Similar quantities can be defined to analyze protocols for continuous distribution of entanglement, such as the average fidelity of entangled links upon consumption [50].

- *Rate-based metrics* focus on the rate at which entanglement can be generated. This includes quantities like the expected time until the first entangled link is distributed [78, 91, 96], which is relevant for on-demand protocols, and the average entanglement distribution rate in the steady state [6, 89], applicable to continuous-distribution protocols.

- *Situational metrics* can be used in special circumstances. For instance, the availability [50] measures the fraction of time in which entanglement is available for

consumption in an entanglement buffering system. In cases where a continuous-distribution protocol is used to distribute multiple entangled states across a large network, the performance can be evaluated using the virtual neighborhood size [94], percolation thresholds [2, 41], and clustering coefficients [25]. Application-specific metrics can also be used. A notable example is the secret key rate [73, 165], which quantifies the amount of secret key that can be extracted per unit time in a quantum key distribution application – this metric depends on both the quality of the entanglement consumed in the application and the rate at which entanglement is distributed.

A plethora of methods have been employed to evaluate the performance of quantum network protocols, both analytically and numerically. Regarding analytical methods, recent studies have used Markov decision processes [78, 91, 101], queueing theory [36, 64, 89], and graph theory [25, 41, 94]. In this dissertation, we borrow techniques from these three disciplines – for a general reference on these topics, see, e.g., refs. [76, 184]. However, quantum network systems often become too complex for fully analytical studies, making Monte Carlo simulations a necessary tool in such cases. Discrete-event simulations are particularly useful, as quantum network protocols frequently involve a variety of discrete stochastic processes. For instance, entanglement generation typically happens in discrete attempts with a low probability of success [148]. Examples of community-built simulators are NetSquid [45], SeQUeNCe [201], and ReQuSim [193].

1.4. THESIS CONTENTS

Next, we present the main questions that remained unanswered at the start of our research, and we explain how we addressed them. At the end of this section, we also provide a brief review of our recent works that are not included in this dissertation.

(1) OPTIMAL STRATEGIES FOR TWO-WAY REPEATER CHAINS – As previously discussed, two-way quantum repeaters enable long-distance entanglement distribution by first generating entangled links between adjacent repeaters along a linear chain, which are then converted into an end-to-end link through entanglement swapping operations. Since entanglement generation is probabilistic, some links must be stored in memory until neighboring links are available for swapping. However, prolonged storage degrades entanglement quality, eventually rendering some states unusable. One approach is to withhold all swaps while discarding old links (i.e., applying cutoffs) until all links are fresh and ready to be swapped, but this can lead to excessive waiting. The opposite strategy – swapping as soon as possible – may also introduce unwanted delays, as fresh links risk being combined with degraded ones, resulting in low-quality output links that must be discarded sooner than the original fresh link. This suggests that early, but not premature, swapping might optimize the overall entanglement distribution rate. Prior to our work, various repeater chain strategies were explored (see, e.g., refs. [22, 44, 96, 116]), each yielding vastly different distribution rates depending on system parameters. However, no clear answer existed as to which approach was best. This led to a

fundamental question: *which is the optimal strategy to operate a repeater chain in the presence of decoherence?*

> In Chapter 2, we find **optimal strategies for operating repeater chains** that outperform all known approaches, particularly when swapping operations succeed probabilistically. Importantly, we assume that links are discarded after some storage time, to ensure a minimum quality of the end-to-end link. Our solutions set benchmarks for evaluating practical strategies within the limitations of existing infrastructure.

(2) FUNDAMENTAL LIMITS TO ONE-WAY REPEATER NETWORKS – One-way quantum repeaters have been proposed as a technology capable of achieving fundamentally higher entanglement distribution rates than two-way repeaters, albeit at a higher cost [133, 136]. However, prior research has largely focused on their design in isolation [7, 18, 135]. When integrated into a network, these repeaters can route quantum data packets similarly to classical networks [119]. Inspired by the success of classical packet switching [8, 51], quantum packet switching strategies have been explored [54]. However, this approach typically allows repeaters to store and delay packet relay based on network traffic. Given the strict timing constraints inherent to quantum networks, where delays can significantly degrade quantum states, a circuit switching approach with a reservation system could offer superior performance by minimizing the impact of network-induced delays on state delivery. *Can quantum circuit switching be a practical solution? Is it feasible to build a large-scale, functional quantum network using this approach?*

> In Chapter 3, we propose a **quantum circuit switching** protocol for one-way quantum repeater networks. Among other findings, we show that, despite the presence of numerous repeaters to mitigate losses, users must remain within a metropolitan area – only a few kilometers from each other – to ensure entanglement delivery within a finite time.

(3) CONTINUOUS ENTANGLEMENT DISTRIBUTION – In contrast to on-demand distribution, investigated in the previous research questions, continuous distribution allocates entanglement to users before they request it. While continuous-distribution protocols have been proposed [35, 66, 105, 149], they have typically been evaluated using traditional performance metrics, which may not fully capture the unique properties of these strategies. For example, the expected time to distribute end-to-end entanglement is commonly used to assess on-demand protocols. However, this metric may be less relevant for continuous-distribution protocols, as entangled links may be distributed but never used if not requested. In such cases, the rate at which those links are distributed becomes irrelevant, as the focus shifts to their availability when needed. Furthermore, a comprehensive comparison of both strategies has been lacking in the literature. This raises important questions:

how should we measure the performance of continuous-distribution protocols? How do these protocols compare to on-demand strategies?

> In Chapter 4, we introduce two key **performance metrics** for continuous-distribution protocols. We then use the concept of Pareto optimality to formulate a multi-objective optimization problem to maximize performance.
> In Chapter 5, we evaluate the **performance enhancement that continuous distribution may provide** over on-demand distribution. For this analysis, we assume a specific request model and we use the expected waiting time to complete requests as a performance metric. We find that, in certain scenarios, continuous distribution, which delivers entanglement preemptively before requests are made, can surprisingly lead to longer waiting times.

(4) ENTANGLEMENT BUFFERING – Once distributed, users may wish to store entanglement for later use, but quantum states degrade over time due to decoherence. Purification protocols have been widely studied to address this issue [15, 53, 56, 205]. These protocols have typically been analyzed in static settings, where a fixed set of input states is assumed, and the goal is to compute the output state after one or more applications of the protocol. In a networked scenario, however, entangled states are delivered and consumed dynamically, often following stochastic processes. *Is it possible to leverage purification protocols to design a dynamic architecture that ensures fresh entanglement is available at any time?*

> In Chapter 6, we develop a systematic framework for the study and design of **entanglement buffers** that make use of purification subroutines to store high-quality bipartite entanglement, introducing two key performance metrics: the probability that entanglement is available at any given time and the average entanglement quality upon consumption. We then derive analytical solutions to study a buffer that only uses two quantum memories.
> In Chapter 7, we find closed-form solutions for the performance of a more general system containing an **arbitrary number of quantum memories**, which allows for more sophisticated purification subroutines and boosted performance. We also show that purification must be performed as frequently as possible to maximize the average fidelity of consumed entanglement, even if this often leads to the loss of high-quality entanglement due to purification failures.

While our work has answered multiple questions, it has also raised many new ones. At the end of each chapter, we include an outlook section where we discuss specific directions for future research. In Chapter 8, we provide general concluding remarks and reflect on the future of the field.

1.4.1. CONTRIBUTIONS NOT INCLUDED IN THIS THESIS

Here, we provide a brief overview of recent works in which we have contributed during the course of the PhD, but which are not included in this dissertation. These publications,

while relevant, focus on different aspects of the research that extend beyond the scope of this thesis.

- In Chapter 2, we identify optimal strategies for operating a chain of quantum repeaters with the goal of delivering end-to-end entanglement as quickly as possible. One of our main assumptions is that classical communication is instantaneous. In ref. [31] (master's thesis supervised jointly with Prof. Vardoyan), we relax this assumption, and we present **heuristics for designing repeater chain policies** that maintain strong performance under more realistic engineering constraints, specifically classical communication delays.

- In ref. [75], we investigate another aspect of entanglement distribution in quantum repeater chains. Usually, intermediate entangled states are discarded once their quality has decayed below a certain threshold, after a fixed storage time. In ref. [75], we propose **discarding entangled states at random**. We demonstrate that this strategy, which avoids the need to track and communicate storage times, can achieve similar end-to-end delivery rates and end-to-end entanglement quality in certain parameter regimes.

- In ref. [40], we propose using **networks with a tree topology**, where users are positioned as the leaves of the tree, for on-demand multi-user entanglement distribution. We show that, although this topology is vulnerable to node deletion, tree networks require fewer qubits per node to prevent traffic congestion than other topologies, thereby offering a more efficient solution.

- In ref. [177], we investigate the use of protocols for **continuous distribution of entanglement in networks with a regular topology**. A remarkable insight is that the network's boundary conditions – whether fixed (nodes are not connected beyond the physical edges of the network) or periodic (nodes on one edge are connected to those on the opposite edge) – play a crucial role in determining how much entanglement the nodes can share at any given time.

- The study of quantum networks often focuses on optimizing the performance of protocols and architectures. However, in complex systems where analytical solutions are not available, traditional optimization techniques that rely on continuity, differentiability, or convexity may become inapplicable. In ref. [151], we introduce an **efficient optimization workflow for quantum network systems**. Using a network simulator, we train simple machine learning surrogate models that mimic the system's behavior, and we optimize the performance of such surrogates. The solutions found by our algorithm consistently outperform those obtained with related numerical approaches, such as simulated annealing [104] and Bayesian optimization [98].

2

Optimal Policies for Entanglement Distribution in Two-User Networks

Álvaro G. Iñesta, Gayane Vardoyan, Lara Scavuzzo, and Stephanie Wehner

Protocol is how we get on the same page;
in fact, the word is rooted in the Greek protokollon*, "first glue",*
which referred to the outer page attached to a book or manuscript.

— Brian Christian and Tom Griffiths

We study the limits of bipartite entanglement distribution using a chain of quantum repeaters that have quantum memories. To generate end-to-end entanglement, each node can attempt the generation of an entangled link with a neighbor, or perform an entanglement swapping measurement. A maximum storage time, known as cutoff, is enforced on the memories to ensure high-quality entanglement. Nodes follow a policy that determines when to perform each operation. Global-knowledge policies take into account all the information about the entanglement already produced. Here, we find global-knowledge policies that minimize the expected time to produce end-to-end entanglement. Our methods are based on Markov decision processes and value and policy iteration. We compare optimal policies to a policy in which nodes only use local information. We find that the advantage in expected delivery time provided by an optimal global-knowledge policy increases with increasing number of nodes and decreasing probability of successful swapping.

This chapter has been published separately in ref. [91].

2.1. INTRODUCTION

2

Bipartite entangled states shared between two parties are often required as a basic resource in quantum network applications. As an example, in cryptography, bipartite entanglement can be directly used for quantum key distribution between two parties [12, 59], but also in multi-party applications such as quantum secret sharing [10]. Bipartite entanglement can also be used to generate multipartite entangled states that are necessary for other applications [29, 107, 145]. As a consequence, a reliable method to distribute entanglement in a quantum network is crucial for the implementation of quantum cryptography applications.

Two neighboring nodes in a quantum network can generate a shared bipartite entangled state, which we call an entangled link. This can be done, e.g., by generating an entangled pair at one node and sending half of the pair to the neighbor via an optical fiber [174, 206] or free space [169, 181]. Two distant nodes can generate an entangled link by generating entanglement between each pair of adjacent nodes along a path that connects them, and then combining these entangled links into longer-distance bipartite entanglement via entanglement swap operations [55, 164]. This path constitutes a quantum repeater chain (see Figure 2.1). We consider repeater chains in which nodes can store quantum states in the form of qubits and perform operations and measurements on them. Experimentally, qubits can be realized with different technologies, such as NV centers [16, 83, 87, 147, 161] and trapped ions [130, 172].

We focus on a single repeater chain of n equidistant and identical nodes, which could be part of a larger quantum network. To generate an entangled link between the two end nodes, also called end-to-end entanglement, we assume the nodes can perform the following operations: (i) heralded generation of entanglement between neighbors [9, 16], which succeeds with probability p and otherwise raises a failure flag; (ii) entanglement swaps [55, 164, 209], which consume two adjacent entangled links to generate a longer-distance link with probability p_s; and (iii) removal of any entangled link that existed for longer than some cutoff time t_{cut}, to prevent generation of low-quality end-to-end entanglement due to decoherence [43, 103, 116, 160, 161]. Note that cutoff times are a key ingredient, since many applications require quantum states with a high enough quality.

We assume that nodes always attempt entanglement generation if there are qubits available. Cutoffs are always applied whenever an entangled link becomes too old. However, nodes are free to attempt swaps as soon as entangled links are available or some time later, so they must agree on an entanglement distribution policy: a set of rules that indicate when to perform a swap. We define an optimal policy as a policy that minimizes the expected entanglement delivery time, which is the average time required to generate

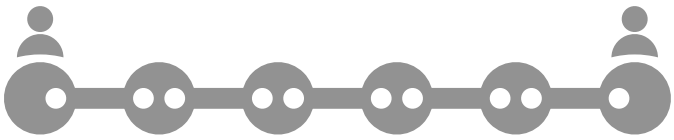

Figure 2.1: **A quantum repeater chain that can store two qubits per intermediate node and one qubit per end node.** White circles represent qubits. All nodes are equidistant and identical.

end-to-end entanglement. Here, we consider optimal global-knowledge policies, in which nodes have information about all the entangled links in the chain. A policy is local when the nodes only need to know the state of the qubits they hold. An example of local policy is the swap-asap policy, in which each node performs a swap as soon as both entangled links are available.

Previous work on quantum repeater chains has mostly focused on the analysis of specific policies rather than on the search for optimal policies. For example, [44] provides analytical bounds on the delivery time of a "nested" policy [22], and [96] optimizes the parameters of such a policy with a dynamic programming approach. Delivery times can be studied using Markov models. In [167], the authors introduce a methodology based on Markov chains to calculate the expected delivery time in repeater chains that follow a particular policy. Similar techniques have also been applied to other quantum network topologies, such as the quantum switch [185, 186]. Here, we focus on Markov decision processes (MDPs), which have already been applied to related problems, e.g., in [102], the authors use an MDP formulation to maximize the quality of the entanglement generated between two neighboring nodes and between the end nodes in a three-node repeater chain. Our work builds on [168], wherein the authors find optimal policies for quantum repeater chains with perfect memories. Since quantum memories are expected to be noisy, particularly in the near future, quantum network protocols must be suitable for imperfect memories. Here, we take a crucial step towards the design of high-quality entanglement distribution policies for noisy hardware. By formulating a generalized MDP to include finite storage times, we are able to find optimal policies in quantum repeater chains with imperfect memories. Our optimal policies provide insights for the design of entanglement distribution protocols.

Our main contributions are as follows:

- We introduce a general MDP model for homogeneous repeater chains with memory cutoffs. The latter constraint poses a previously unaddressed challenge: MDP states must incorporate not only entangled link absence/presence, but also link age.
- We find optimal policies for minimizing the expected end-to-end entanglement delivery time, by solving the MDP via value and policy iteration.
- Our optimal policies take into account global knowledge of the state of the chain and therefore constitute a lower bound to the expected delivery time of policies that use only local information.

Our main findings are as follows:

- The optimal expected delivery time in a repeater chain with deterministic swaps ($p_s = 1$) can be orders of magnitude smaller than with probabilistic swaps.
- When swaps are deterministic, the advantage in expected delivery time offered by an optimal policy as compared to the swap-asap policy increases for lower probability of entanglement generation, p, and lower cutoff time, t_{cut}, in the parameter region explored. However, when swaps are probabilistic, we find the opposite behavior: the advantage increases for higher p and t_{cut}.

- The advantage provided by optimal policies increases with higher number of nodes, both when swaps are deterministic and probabilistic, albeit the advantage is larger in case of the latter.

This chapter is structured as follows. In Section 2.2, we explain in detail our repeater chain model. Then, in Section 2.3 present our main results. In Section 2.4, we discuss the implications and limitations of our work. In Appendix 2.5, we provide more details on how to formulate the MDP and how to solve it.

2.2. NETWORK MODEL

We analyze quantum repeater chains wherein nodes can store quantum states in the form of qubits and can perform three basic operations with them: entanglement generation, entanglement swaps, and cutoffs.

ENTANGLEMENT GENERATION. Two adjacent nodes can attempt the heralded generation of an entangled link (i.e., a shared bipartite entangled state), succeeding with probability p. Generation of entanglement is heralded, meaning that the nodes receive a message stating whether they successfully generated an entangled link or not [9, 16]. We assume that entanglement generation is noisy. Hence, the newly generated entangled links are not maximally entangled states but Werner states [197]. Werner states are maximally entangled states that have been subjected to a depolarizing process, which is a worst-case noise model [58], and they can be written as follows:

$$\rho = \frac{4F-1}{3}\left|\phi^+\right\rangle\!\left\langle\phi^+\right| + \frac{1-F}{3}\mathbb{I}_4, \tag{2.1}$$

where $\left|\phi^+\right\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$ is a maximally entangled state, F is the fidelity of the Werner state to the state $\left|\phi^+\right\rangle$, and $\mathbb{I}_d$ is the d-dimensional identity. In our notation, the fidelity of a mixed state ρ to a pure state $\left|\phi\right\rangle$ is defined as

$$F(\rho, \left|\phi\right\rangle) := \left\langle\phi\right|\rho\left|\phi\right\rangle. \tag{2.2}$$

We assume that the fidelity of newly generated entangled links is $F_{\text{new}} \leq 1$.

ENTANGLEMENT SWAP. Two neighboring entangled links can be fused into a longer-distance entangled link via entanglement swapping. Consider a situation where node B shares an entangled link with node A, and another link with node C (see Figure 2.2). Then, B can perform an entanglement swap to produce an entangled link between A and C while consuming both initial links [55, 164, 209]. We refer to the link generated in a swap operation as a swapped link. This operation is also probabilistic: a new link is produced with probability p_s, and no link is produced (but both input links are still consumed) with probability $1 - p_s$.

The generation of an entangled link between two end nodes without intermediate repeaters is limited by the distance between the end nodes [132] – e.g., the noise affecting

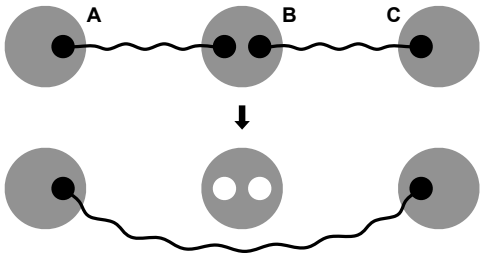

Figure 2.2: **Entanglement swap.** When node B performs a swap, an entangled link between nodes A and B and an entangled link between nodes B and C are consumed to produce a single entangled link between A and C. This operation is essential for the generation of long-distance entanglement.

a photon sent over an optical fiber grows exponentially with the length of the fiber [22]. Therefore, a repeater chain that makes use of entanglement swapping is needed to generate end-to-end entanglement over long distances.

CUTOFFS. The fidelity of a quantum state decreases over time due to couplings to the environment [38, 58]. These decoherence processes can be captured using a white noise model in which a depolarizing channel is applied to the entangled state at every instant. As a result, the fidelity of a Werner state at time t, $F(t)$, is given by

$$F(t) = \frac{1}{4} + \left(F(t-\Delta t) - \frac{1}{4}\right)e^{-\Delta t/\tau}, \tag{2.3}$$

where Δt is an arbitrary interval of time and τ is a parameter that characterizes the exponential decay in fidelity of the whole entangled state due to the qubits being stored in noisy memories. This parameter depends on the physical realization of the qubit. (2.3) is derived in Appendix 2.6.

In general, quantum network applications require quantum states with fidelity above some threshold value F_{min}. A common solution is to impose a cutoff time t_{cut} on the entangled links: all entangled links used to generate the final end-to-end link must be generated within a time window of size t_{cut} [160]. Imposing memory cutoffs requires keeping track of the time passed since the creation of each entangled link. We call this time the age of the link. A link is discarded whenever it gets older than t_{cut}. Moreover, we assume that an entangled link generated as a result of entanglement swapping assumes the age of the oldest link that was involved in the swapping operation. Another valid approach to calculate the age of a swapped link would be to re-compute the age based on the post-swap fidelity, although this would lead to a more complicated formulation to ensure that all the links that were used to produce a swapped link were generated within the time window of size t_{cut}. To produce end-to-end links with fidelity above F_{min} on a repeater chain that generates new links with fidelity F_{new}, it suffices to ensure that the sequence of events that produces the lowest end-to-end fidelity satisfies this requirement. In Appendix 2.7, we show that such a sequence of events corresponds to all links being simultaneously generated in the first attempt and all the entanglement swaps being performed at the end of the t_{cut} interval. Analyzing such a sequence of events leads

to the following condition for the cutoff time:

$$t_{\text{cut}} \leq -\tau \ln\left(\frac{3}{4F_{\text{new}}-1}\left(\frac{4F_{\text{min}}-1}{3}\right)^{\frac{1}{n-1}}\right), \tag{2.4}$$

where n is the number of nodes. For a full derivation of the previous condition, see Appendix 2.7.

In this chapter, we consider quantum networks that operate with a limited number of qubits. Specifically, we use the following additional assumptions:

(i) The chain is **homogeneous**, i.e., the hardware is identical in all nodes. This means that all pairs of neighbors generate links with the same success probability p and fidelity F_{new}, all swaps succeed with probability p_{s}, all states decohere according to some coherence time τ, and all nodes apply the same cutoff time t_{cut}. This assumption may not hold for some long-distance quantum networks where each node is implemented using a different technology, but may be directly applicable to, e.g., small metropolitan-scale networks.

(ii) We assume that each node has only **two storage qubits**, each of which is used to generate entanglement with one side of the chain. Each end node has a single storage qubit. This assumption is in line with the expectations for early quantum networks, in which nodes are likely to have a number of storage qubits on the order of the unit (e.g., in [147] the authors realized the first three-node quantum network using NV centers, each with a single storage qubit).

(iii) We also assume that classical communication between nodes is instantaneous. This means that every node has **global knowledge** of the state of the repeater chain in real time. In general, this is not a realistic assumption. However, given that classical communication delays decrease the performance of the network, our results constitute a lower bound on the expected delivery time of real setups and can be used as a benchmark.

(iv) Time is discretized into non-overlapping **time slots**. During one time slot: (i) first, each pair of neighboring nodes attempts entanglement generation if they have free qubits; (ii) second, some time is allocated for the nodes to attempt entanglement swaps; and (iii) lastly nodes discard any entangled link that existed for longer than t_{cut} time slots. To decide if they want to perform a swap in the second part of the time step, nodes can take into account the state of the whole chain, including the results from entanglement generation within the same time slot, since classical communication is instantaneous. The unit of time used in this chapter is the duration of a time slot, unless otherwise specified.

A repeater chain under the previous assumptions is characterized by four parameters:

- n: number of nodes in the chain, including end nodes.
- p: probability of successful entanglement generation.

- p_s: probability of successful swap.
- t_{cut}: cutoff time. Note that F_{new}, F_{min}, and τ are used to determine a proper value of cutoff time (see condition (2.4)), but they are not needed after that.

In an experimental setup, the value of p is determined by the inter-node distance and the type of hardware used, as quantum nodes can be realized using different technologies, such as NV centers [16, 83, 87, 147, 161] and trapped ions [130, 172]. Linear optics setups generally perform swaps with probability $p_s = 0.5$ [32, 55], while other setups can perform deterministic swaps ($p_s = 1$) at the cost of a slower speed of operation [147]. The cutoff time t_{cut} can be chosen by the user, as long as condition (2.4) is satisfied. Note that (2.4) depends on τ (which depends on the hardware available), F_{new} (which depends on the hardware and the choice of entanglement generation protocol), and F_{min} (which is specified by the final application).

The state of the repeater chain at the end of each time slot can be described using the age of every entangled link. In Figure 2.3 we show an example of the evolution of the state of a chain with cutoff $t_{cut} = 3$, over four time slots:

- In the first time slot ($t \in [0,1)$), all pairs of neighbors attempt entanglement generation, but it only succeeds between nodes two and three. No swaps can be performed, and the only link present is younger than the cutoff, so it is not discarded.
- In the second time slot ($t \in [1,2)$), the age of the link between nodes two and three increases by one. All pairs of neighbors (except nodes two and three) attempt entanglement generation, which succeeds between nodes four and five.
- In the third time slot ($t \in [2,3)$), the age of both existing links increases by one. All pairs of neighbors (except nodes two and three and nodes four and five) attempt entanglement generation, and only nodes five and six succeed. A swap can be performed at node five but they decide to wait.
- In the fourth time slot ($t \in [3,4)$), the age of every existing link increases by one. Nodes one and two and nodes three and four attempt entanglement generation but none of the pairs succeeds. A swap is successfully performed at node five, and a new link between nodes four and six is generated. This new link assumes the age of the oldest link involved in the swap operation. Lastly, the entangled link between nodes two and three is discarded, as its age reached the cutoff time.

2.3. OPTIMAL ENTANGLEMENT DISTRIBUTION POLICIES

As described above, nodes always attempt entanglement generation if there are qubits available. Cutoffs are always applied whenever an entangled state becomes too old. Since nodes are free to attempt swaps as soon as entangled links are available or some time later, they must agree on an entanglement distribution policy: a set of rules that indicate when to perform a swap. An optimal policy minimizes the average time required to generate end-to-end entanglement when starting from any state (i.e., from any combination of existing

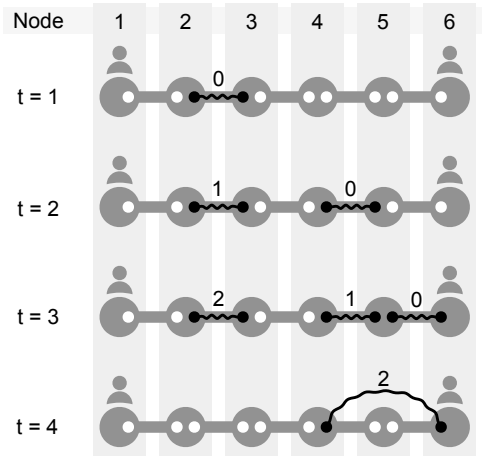

Figure 2.3: **Example of entangled link dynamics in a repeater chain.** Each row represents the state of the chain at the end of time slot t. Entangled links are represented as black solid lines, with occupied qubits as black circles. The number above each entangled link is the age of the link. We assume cutoff $t_{\text{cut}} = 3$.

links) and following said policy. In particular, it minimizes the mean entanglement delivery time, which is the average time required to generate end-to-end entanglement when starting from the state with no entangled links. We employ the mean entanglement delivery time as a performance metric.

In a global-knowledge policy, nodes have information about all the entangled links in the chain. In a local-knowledge policy, the nodes only need to know the state of the qubits they hold. An example of local policy is the swap-asap policy, in which each node performs a swap as soon as both entangled links are available.

We model the evolution of the state of the repeater chain as an MDP. We then formulate the Bellman equations [175] and solve them using value iteration and policy iteration to find global-knowledge optimal policies. More details and formal definitions are provided in Appendix 2.5.

Let us now describe the relation between the expected delivery time of an optimal policy, T_{opt}, and the variables of the system (n, p, p_{s}, and t_{cut}). Repeater chains with a larger number of nodes n yield a larger T_{opt}, since more entangled links need to be generated probabilistically. When p is small, more entanglement generation attempts are required to succeed, yielding a larger T_{opt}. Decreasing p_{s} also increases T_{opt}, since more attempts at entanglement swapping are required on average. When t_{cut} is small, all entangled states must be generated within a small time window and therefore T_{opt} is also larger. Figure 2.4 shows the expected delivery time of an optimal policy in a five-node chain. Interestingly, p_{s} has a much stronger influence on T_{opt} than p and t_{cut}: decreasing p_{s} from 1 to 0.5 in a five-node chain translates into an increase in T_{opt} of an order of magnitude. Similar behavior is observed for other values of n, as shown in Appendix 2.8.

To evaluate the advantages of an optimal policy, we use the swap-asap policy as a baseline. Early swaps can provide an advantage in terms of delivery time, since swapping earlier can free up qubits that can be used to generate backup entangled links, as displayed

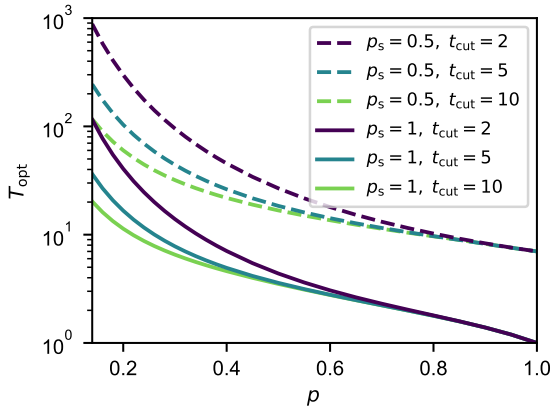

Figure 2.4: **The expected delivery time increases with lower p, p_s, and t_{cut}.** Expected delivery time of an optimal policy, T_{opt}, versus p in a five-node chain, for different values of cutoff ($t_{cut} = 2, 5, 10$). Solid lines correspond to deterministic swaps ($p_s = 1$) and dashed lines correspond to probabilistic swaps with $p_s = 0.5$.

in the first transition in Figure 2.5. However, the age of a swapped link may reach the cutoff time earlier than one of the input links consumed in the swap, as the swapped link assumes the age of the oldest input link. Following the example in Figure 2.5 and assuming $t_{cut} = 1$, if no swaps are performed, the links between nodes two and three and between three and four will exist for one more time slot, while the link between nodes four and five will be removed immediately since it reached the cutoff time. If both swaps are performed, the swapped link between nodes two and five will be removed immediately since it reached the cutoff time. Since we have arguments in favor of and against swapping early, it is not trivial to determine the scenarios in which the swap-asap policy is close to optimal. Next, we compare the expected delivery times of an optimal global-knowledge policy and the swap-asap policy.

Figure 2.6 shows the relative difference between the expected delivery times of an optimal global-knowledge policy, T_{opt}, and that of the swap-asap policy, T_{swap}, in a five-node chain. Increasing values of $(T_{swap} - T_{opt})/T_{opt}$ mean that the optimal policy is increasingly faster on average. Note that we restrict our analysis to the parameter regime $p \geq 0.3$ and $2 \leq t_{cut} \leq 6$ due to the very large computational cost of calculating the solution for smaller p and larger t_{cut} (for more details, see Appendix 2.5). Let us first focus on deterministic swaps (Figure 2.6a). The advantage provided by an optimal policy increases for decreasing p. When p is small, links are more valuable since they are harder to generate. Therefore, it is convenient to avoid early swaps, as they effectively increase the ages of the links involved and make them expire earlier. When t_{cut} is small, a similar effect happens: all entangled links must be generated within a small time window and early swaps can make them expire too soon. For larger t_{cut}, increasing the age of a link does not have a strong impact on the delivery time, since the time window is larger. Therefore, an optimal policy is increasingly better than swap-asap for decreasing t_{cut}. The maximum difference between expected delivery times in the parameter region explored is 5.25%.

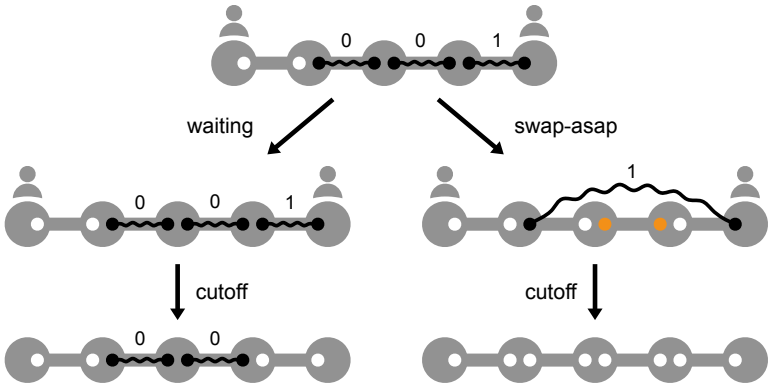

Figure 2.5: **Swap-asap policies free up qubits, but swapped links expire earlier.** Evolution of an example state when following a waiting policy versus the swap-asap policy during a single time slot. Entangled links are represented as solid black lines, with occupied qubits in black and free qubits in white. A waiting policy decides to not perform any swap, while the swap-asap policy decides to swap all three links. The swap frees up qubits (marked in orange) that can be used to resume entanglement generation either if the swap is successful, as in the picture, or not. After performing swaps, a cutoff $t_{\text{cut}} = 1$ is applied and links with age 1 are removed, causing the swapped link to expire.

Interestingly, probabilistic swaps (Figure 2.6b) yield an opposite behavior in the parameter region explored: optimal policies are increasingly better than swap-asap for increasing p and t_{cut} (except when $p \leq 0.4$ and $t_{\text{cut}} \leq 3$), and the relative difference in expected delivery time can be as large as 13.2% (achieved in a five-node chain with $p = 0.9$ and $t_{\text{cut}} = 6$). One reason for this may be the action that each policy decides to perform when the repeater chain is in a full state, which is a situation where each pair of neighboring nodes shares an entangled link (see state at the top of Figure 2.7). When swaps are deterministic, the optimal policy chooses to swap all links in a full state, since end-to-end entanglement will always be achieved. However, when swaps are probabilistic, an optimal policy generally chooses to perform two separate swaps (see Figure 2.7), similar to the nested purification scheme proposed in [22]. As an example, for $n = 5$, $p = 0.9$, $t_{\text{cut}} = 2$, and $p_{\text{s}} = 0.5$, the swap-asap policy yields an expected delivery time of $T = 9.35$. If, in full states, the swap at the third node is withheld, T drops to 8.34. The swap-asap policy is on average slower than this modified policy by 12.1%. The action chosen in full states has a stronger influence on T for increasing p. This is because full states are more frequent for large p: whenever a swap fails, a full state is soon recovered, since new entangled states are generated with high probability. As a consequence, an optimal policy is increasingly better than swap-asap for higher p when swaps are probabilistic. A similar effect happens for large t_{cut}. Note however that the effect of the action chosen in full states is practically irrelevant in four-node chains (see Appendix 2.8). Note also that the advantage of an optimal policy in terms of delivery time is not always monotonic in p and t_{cut} (see Appendix 2.8).

Optimal policies are also increasingly faster than swap-asap for increasing n, as shown in Figure 2.8. For example, for $p = 0.3$, $p_{\text{s}} = 0.5$, and $t_{\text{cut}} = 2$, the relative difference in expected delivery time is 1.7%, 5.9%, and 12.3%, for $n = 4$, 5, and 6, respectively. This is

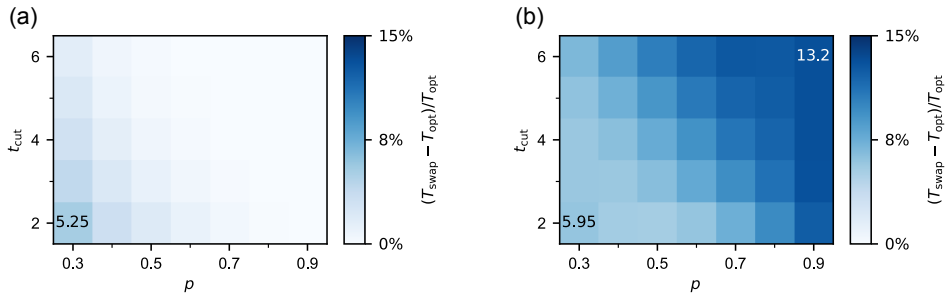

Figure 2.6: **In a five-node chain, an optimal policy performs increasingly better than swap-asap for lower/higher values of p and t_{cut} when swaps are deterministic/probabilistic.** Relative difference between the expected delivery times of an optimal policy, T_{opt}, and the swap-asap policy, T_{swap}, in a five-node chain, for different values of p and t_{cut}. (a) Deterministic swaps ($p_s = 1$). (b) Probabilistic swaps ($p_s = 0.5$).

in line with the fact that, when the number of nodes grows, there are increasingly more states in which the optimal action to perform is a strict subset of all possible swaps, as shown in Appendix 2.9. Note that, in three- and four-node chains, the relative difference in expected delivery time is generally below 1%.

2.4. DISCUSSION

Our work sheds light on how to distribute entanglement in quantum networks using a chain of intermediate repeaters with pre-configured cutoffs. We have shown that optimal global-knowledge policies can significantly outperform other policies, depending on the properties of the network. In particular, we have found and explained non-trivial examples in which performing swaps as soon as possible is far from optimal. We have also contributed a simple methodology to calculate optimal policies in repeater chains with cutoffs that can be extended to more realistic scenarios, e.g., asymmetric repeater chains, by modifying the transition probabilities of the MDP.

In this work, we have assumed that classical communication is instantaneous. Hence, our optimal policies may become sub-optimal in setups with non-negligible communication times, where decisions must be made using local information only. Nevertheless, our optimal policies still constitute a best-case policy against which to benchmark.

Note also that we have restricted our analysis to repeater chains with less than seven nodes. This is due to the exponentially large computational cost of solving the MDP for larger chains (see Appendix 2.12 for further details). However, each entanglement swap decreases the fidelity of the entangled links. Hence, a large number of swaps limits the maximum end-to-end fidelity achievable, making chains with a very large number of nodes impractical. Therefore, we consider the analysis of short chains to be more relevant.

An interesting extension of this work would be to explore different cutoff policies. For example, one could allow the nodes to decide when to discard entangled links, or one could optimize simultaneously over the cutoff and the swapping policy. This may lead to

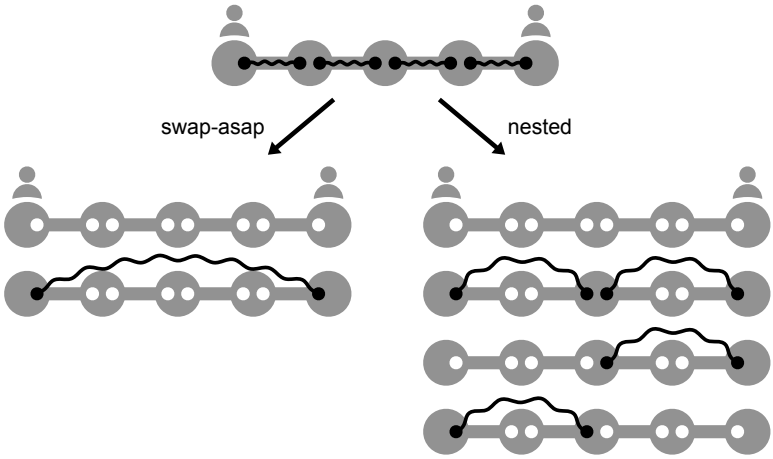

Figure 2.7: **All possible transitions after performing a swap-asap action or a nested action in a full state, depending on which swaps succeed.** In full states, every pair of neighbors shares an entangled link (solid black lines, with occupied qubits in black and free qubits in white). The swap-asap policy decides to swap all links, while the nested approach consists in swapping only at nodes 2 and 4. When swaps are probabilistic, the nested approach is generally optimal in terms of expected delivery time.

improved optimal policies.

As a final remark, note that we have employed the expected delivery time as the single performance metric. In some cases, the expected value and the variance of the delivery time distribution are within the same order of magnitude (some examples are shown in Appendix 2.10). Therefore, an interesting follow-up analysis would be to study the delivery time probability distribution instead of only the expected value. Additionally, we put fidelity aside by only requiring an end-to-end fidelity larger than some threshold value, via a constraint on the cutoff time. This constraint can be lifted to optimize the fidelity instead of the expected delivery time, or to formulate a multi-objective optimization problem to maximize fidelity while minimizing delivery time.

2.5. [APPENDIX] METHODS

We have formulated the problem of finding optimal entanglement distribution policies as an MDP where each state is a combination of existing entangled links and link ages. Let $\boldsymbol{s}$ be the state of the repeater chain at the beginning of a time slot. As previously explained, $\boldsymbol{s}$ can be described using the age of every entangled link. Mathematically, this means that $\boldsymbol{s}$ can be represented as a vector of size $\binom{n}{2}$:

$$\boldsymbol{s} = [g_1^2, g_1^3, \ldots, g_1^n;\ g_2^3, \ldots, g_2^n;\ \ldots;\ g_{n-1}^n],$$

where g_i^j is the age of the entangled link between nodes i and j (if nodes i and j do not share an entangled link, then $g_i^j = -1$). In each time slot, the nodes must choose and perform an action a. Mathematically, a is a set containing the indices of the nodes that

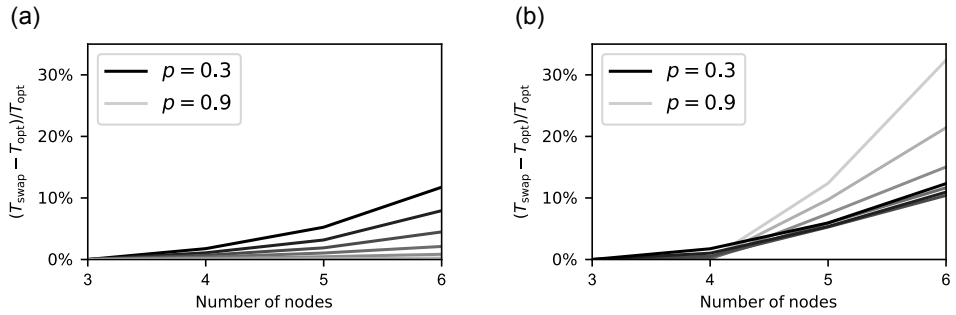

Figure 2.8: **An optimal policy performs increasingly better than swap-asap in longer chains.** Relative difference between the expected delivery times of an optimal policy, T_{opt}, and the swap-asap policy, T_{swap}, for $t_{cut} = 2$ and different values of p, as a function of the number of nodes n. Black lines correspond to $p = 0.3$, and the value of p increases in steps of 0.1 with increasing line transparency up to $p = 0.9$. (a) Deterministic swaps ($p_s = 1$). (b) Probabilistic swaps ($p_s = 0.5$).

must perform swaps (if no swaps are performed, $a = \emptyset$).

The state of the chain at the end of the time slot is $\boldsymbol{s}'$. Since entanglement generation and swaps are probabilistic, the transition from $\boldsymbol{s}$ to $\boldsymbol{s}'$ after performing a happens with some transition probability $P(\boldsymbol{s}'|\boldsymbol{s}, a)$. A policy is a function π that indicates the action that must be performed at each state, i.e.,

$$\pi\colon \boldsymbol{s} \in \mathcal{S} \rightarrow \pi(\boldsymbol{s}) \in \mathcal{A}\,,$$

where $\mathcal{S}$ is the state space and $\mathcal{A}$ is the action space. W.l.o.g., we only consider deterministic policies, otherwise a policy would be a probability distribution instead of a function (see Appendix 2.11 for further details).

Let us define $\boldsymbol{s}_0$ as the state where no links are present and $\mathcal{S}_{end}$ as the set of states with end-to-end entanglement, also called absorbing states. In general, the starting state is $\boldsymbol{s}_0$, and the goal of the repeater chain is to transition to a state in $\mathcal{S}_{end}$ in the fewest number of steps. When a state in $\mathcal{S}_{end}$ is reached, the process stops. Let us define the expected delivery time from state $\boldsymbol{s}$ when following policy π, $T_\pi(\boldsymbol{s})$, as the expected number of steps required to reach an absorbing state when starting from state $\boldsymbol{s}$. The expected delivery time is also called hitting time in the context of Markov chains (see Chapter 9 from [184]). A policy π is better than or equal to a policy π' if $T_\pi(\boldsymbol{s}) \leq T_{\pi'}(\boldsymbol{s})$, $\forall\, \boldsymbol{s} \in \mathcal{S}$. An optimal policy π^* is one that is better than or equal to all other policies. In other words, an optimal policy is one that minimizes the expected delivery time from all states. One can show that there exists at least one optimal policy in an MDP with a finite and countable set of states (see Section 2.3 from [176]). To find such an optimal policy, we employ the following set of equations, which are derived in Appendix 2.11:

$$T_\pi(\boldsymbol{s}) = 1 + \sum_{\boldsymbol{s}' \in \mathcal{S}} P(\boldsymbol{s}'|\boldsymbol{s}, \pi) \cdot T_\pi(\boldsymbol{s}'),\ \forall\, \boldsymbol{s} \in \mathcal{S}, \tag{2.5}$$

where $\mathcal{S}$ is the state space and $P(\boldsymbol{s}'|\boldsymbol{s}, \pi)$ is the probability of transition from state $\boldsymbol{s}$ to state $\boldsymbol{s}'$ when following policy π. Equations (2.5) are a particular case of what is generally known in the literature as the Bellman equations.

An optimal policy can be found by minimizing $T_\pi(\boldsymbol{s})$, $\forall\, \boldsymbol{s} \in \mathscr{S}$, using (2.5). To solve this optimization problem, we used value iteration and policy iteration, which are two different iterative methods whose solution converges to the optimal policy (both methods provided the same results). For more details, see Appendix 2.12, and for a general reference on value and policy iteration, see Chapter 4 from [175].

We provide an example of how to calculate the transition probabilities $P(\boldsymbol{s}'|\boldsymbol{s},\pi)$ analytically in Appendix 2.13, although this is generally impractical, since the size of the state space grows at least exponentially with n and polynomially with t_{cut} (as shown in Appendix 2.14, $|\mathscr{S}| = \Omega\big((t_{\text{cut}})^{n-2}\big)$). In the Supplementary Material from ref. [91], we discuss how to simplify the calculation of transition probabilities with a technique that we call *state bunching*.

As a validation check, we also implemented a Monte Carlo simulation that can run our optimal policies, providing the same expected delivery time that we obtained from solving the MDP.

2.6. [APPENDIX] DEPOLARIZATION OF WERNER STATES

In this appendix we show that the fidelity of a Werner state in which each qubit independently experiences a depolarizing process evolves as

$$F(t) = \frac{1}{4} + \Big(F(t-\Delta t) - \frac{1}{4}\Big) \mathrm{e}^{-\frac{\Delta t}{\tau}},$$

where t is the time, Δt is an arbitrary interval of time, and τ is a parameter that characterizes the exponential decay in fidelity of the whole entangled state due to the qubits being stored in noisy memories. Note that we assume independent noise on each qubit since, in our problem, they are stored in different nodes of the repeater chain.

The depolarizing channel [58, 138] is defined as

$$\mathscr{E}_i : \ \rho_i \ \rightarrow \ p\rho_i + (1-p)\frac{\mathbb{I}_2}{2}, \tag{2.6}$$

where ρ_i is a single-qubit state, $0 \leq p \leq 1$ (this p is not to be confused with the entanglement generation probability used in the main text of this chapter), and $\mathbb{I}_d$ is the d-dimensional identity. Let us assume that each qubit independenlty experiences a depolarizing channel while stored in memory for a finite time t_{dep}. During an interval of time t_{dep}, a Werner state ρ with fidelity F is therefore mapped to $(\mathscr{E}_1 \otimes \mathscr{E}_2)(\rho)$, where $\mathscr{E}_i$ is a depolarizing channel acting on the i-th qubit. Let us calculate this output state

explicitly:

$$\begin{aligned}(\mathscr{E}_1\otimes\mathscr{E}_2)(\rho) &= p^2\rho + p(1-p)\,\mathrm{Tr}_2(\rho)\otimes\frac{\mathbb{I}_2}{2} + p(1-p)\frac{\mathbb{I}_2}{2}\otimes\mathrm{Tr}_1(\rho) + (1-p)^2\frac{\mathbb{I}_4}{4}\\ &\overset{a}{=} p^2\rho + p(1-p)\frac{\mathbb{I}_2}{2}\otimes\frac{\mathbb{I}_2}{2} + p(1-p)\frac{\mathbb{I}_2}{2}\otimes\frac{\mathbb{I}_2}{2} + (1-p)^2\frac{\mathbb{I}_4}{4}\\ &= p^2\rho + \big(2p(1-p)+(1-p)^2\big)\frac{\mathbb{I}_4}{4}\\ &\overset{b}{=} p^2\frac{4F-1}{3}\left|\phi^+\right\rangle\!\left\langle\phi^+\right| + p^2\frac{1-F}{3}\mathbb{I}_4 + \big(2p(1-p)+(1-p)^2\big)\frac{\mathbb{I}_4}{4}\\ &\overset{c}{=} \frac{4F'-1}{3}\left|\phi^+\right\rangle\!\left\langle\phi^+\right| + \frac{1-F'}{3}\mathbb{I}_4,\end{aligned} \tag{2.7}$$

with the following steps:

a. We use the fact that the partial trace of a maximally entangled state is a maximally mixed state. As a consequence, $\mathrm{Tr}_i(\rho)=\frac{\mathbb{I}_2}{2}$, for any Werner state ρ.

b. We use the definition of Werner state: $\rho=\frac{4F-1}{3}\left|\phi^+\right\rangle\!\left\langle\phi^+\right| + \frac{1-F}{3}\mathbb{I}_4$.

c. We define $F'=\frac{1}{4}+p^2(F-\frac{1}{4})$.

The output state $(\mathscr{E}_1\otimes\mathscr{E}_2)(\rho)$ is a Werner state with fidelity F'. Then, the application of n successive transformations $\mathscr{E}_1\otimes\mathscr{E}_2$ produces a Werner state with fidelity

$$F^{(n)} = \frac{1}{4} + p^{2n}\Big(F-\frac{1}{4}\Big). \tag{2.8}$$

This can be shown by induction as follows. The base case is proven in (2.7): $F^{(1)} = \frac{1}{4}+p^2(F-\frac{1}{4})$. Next, if we assume that (2.8) is true for $n=k$, we can show that it also holds for $n=k+1$:

$$F^{(k+1)} = \frac{1}{4} + p^2\Big(F^{(k)}-\frac{1}{4}\Big) = \frac{1}{4} + p^2\Big(\frac{1}{4}+p^{2k}\Big(F-\frac{1}{4}\Big)-\frac{1}{4}\Big) = \frac{1}{4}+p^{2(k+1)}\Big(F-\frac{1}{4}\Big),$$

where we have used (2.7) in the first step.

The total time required for these operations is $\Delta t = n\,t_{\mathrm{dep}}$. Therefore, if the fidelity of the state at time $t-\Delta t$ was $F(t-\Delta t)$, the fidelity at t is given by

$$F(t) = \frac{1}{4} + p^{2\Delta t/t_{\mathrm{dep}}}\Big(F(t-\Delta t)-\frac{1}{4}\Big). \tag{2.9}$$

Finally, we map $p\in[0,1]$ to a new parameter $\tau\in[0,+\infty)$ as $p^2\equiv \mathrm{e}^{-t_{\mathrm{dep}}/\tau}$. Then, we obtain

$$F(t) = \frac{1}{4} + \Big(F(t-\Delta t)-\frac{1}{4}\Big)\mathrm{e}^{-\frac{\Delta t}{\tau}}. \tag{2.10}$$

2.7. [APPENDIX] CUTOFF AND THRESHOLD FIDELITY

In the design of a quantum repeater chain, we must select a cutoff time t_{cut} such that the fidelity of any end-to-end entangled link is larger than some threshold F_{min}. We show that this requirement is always satisfied when the cutoff time meets the following condition:

$$t_{\text{cut}} \leq -\tau \ln\left(\frac{3}{4F_{\text{new}}-1}\left(\frac{4F_{\text{min}}-1}{3}\right)^{\frac{1}{n-1}}\right), \tag{2.11}$$

where τ is a parameter that characterizes the exponential decay in fidelity of the whole entangled state due to the qubits being stored in noisy memories, F_{new} is the fidelity of newly generated entangled links, F_{min} is the minimum desired end-to-end fidelity, and n is the number of nodes in the chain.

First, we analyze the impact of a late entanglement swap on the fidelity of the output state. As shown in Appendix 2.6, the fidelity of a Werner state that experiences depolarizing noise independently on each qubit decays as

$$F(t) = \frac{1}{4} + \left(F(t-\Delta t) - \frac{1}{4}\right)e^{-\Delta t/\tau}, \tag{2.12}$$

over an interval of time time Δt. When two Werner states are used as input in an entanglement swap, the output state is a Werner state with fidelity

$$F_{\text{swap}}(F_1, F_2) = F_1 \cdot F_2 + \frac{(1-F_1)\cdot(1-F_2)}{3}, \tag{2.13}$$

where F_1 and F_2 are the fidelities of the input states [132].

Let us consider two Werner states with initial fidelities $F_1(t_0)$ and $F_2(t_0)$, respectively. On the one hand, if we perform a swap and then wait for some time t_{wait}, the final state is a Werner state with fidelity

$$F_{\text{swap-wait}}(t_0 + t_{\text{wait}}) = \frac{1}{4} + \left(F_1(t_0)F_2(t_0) - \frac{1}{4} + \frac{\big(1-F_1(t_0)\big)\big(1-F_2(t_0)\big)}{3}\right)e^{-t_{\text{wait}}/\tau}, \tag{2.14}$$

which can be obtained by applying (2.13) first and (2.12) next. On the other hand, if we wait for some time t_{wait} and then perform the swap, we obtain a Werner state with fidelity

$$\begin{aligned} F_{\text{wait-swap}}(t_0 + t_{\text{wait}}) &= F_1(t_0 + t_{\text{wait}})F_2(t_0 + t_{\text{wait}}) + \frac{\big(1-F_1(t_0 + t_{\text{wait}})\big)\big(1-F_2(t_0 + t_{\text{wait}})\big)}{3} \\ &= \frac{1}{4} + \left(F_1(t_0)F_2(t_0) - \frac{1}{4} + \frac{\big(1-F_1(t_0)\big)\big(1-F_2(t_0)\big)}{3}\right)e^{-2t_{\text{wait}}/\tau}, \end{aligned} \tag{2.15}$$

where we have used (2.12) in the second step and performed some basic algebra. Note that the factor that multiplies the exponential in (2.14) and (2.15) is nonnegative as long as $F_1(t_0), F_2(t_0) \geq \frac{1}{4}$ – if the initial fidelity is $\frac{1}{4}$, the initial state is a maximally mixed state.

By comparing (2.14) and (2.15), we find that an entangled link with larger fidelity is obtained if we first perform an entanglement swap and then wait for time t_{wait} rather than if we wait for time t_{wait} and then perform the swap, since

$$F_{\text{swap-wait}}(t_0 + t_{\text{wait}}) > F_{\text{wait-swap}}(t_0 + t_{\text{wait}}), \ \forall\, t_{\text{wait}} > 0. \tag{2.16}$$

Let us now consider a sequence of m entangled links that can be fused into a single long link after performing $m-1$ swaps. Each of the initial links has fidelity F_i, $i = 1, ..., m$. We want to calculate the final fidelity, assuming that all swaps are successful. For this, it is convenient to define a Werner state in terms of the Werner parameter x:

$$\rho = x\left|\phi^+\right\rangle\left\langle\phi^+\right| + \frac{1-x}{4}\mathbb{1}_4, \tag{2.17}$$

where $\left|\phi^+\right\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$ is a Bell state, and $\mathbb{1}_d$ is the d-dimensional identity. The Werner parameter x is defined in terms of the fidelity as $x = \frac{4F-1}{3}$. Equation (2.13) can be written in terms of the Werner parameter of each state:

$$x_{\text{swap}}(x_1, x_2) = x_1 x_2, \tag{2.18}$$

where x_{swap} is the Werner parameter of the output state after swapping two Werner states with Werner parameters x_1 and x_2. If we apply Equation (2.18) repeatedly to our sequence of m entangled links, assuming that all swaps happen simultaneously (i.e., with no decoherence happening in between swaps), we obtain a final state with Werner parameter

$$x_{\text{final}} = x_1 x_2 \dots x_m = \prod_{i=1}^{m} \frac{4F_i - 1}{3}. \tag{2.19}$$

Then, the final fidelity is given by

$$F_{\text{final}} = \frac{3x_{\text{final}} + 1}{4} = \frac{1}{4} + \frac{3}{4}\prod_{i=1}^{m} \frac{4F_i - 1}{3}. \tag{2.20}$$

A similar result was derived in [22], although assuming $F_i = F$, $\forall i$.

We are now ready to find a relationship between the cutoff time and the minimum fidelity in an n-node quantum repeater chain with cutoff time t_{cut}. For this, we need to identify the sequence of events that produces the end-to-end link with the lowest fidelity. First, note that all the entangled links that eventually form a single end-to-end link are created within a window of t_{cut} time slots. According to (2.16), delaying entanglement swaps has a negative impact on the final fidelity. Therefore, the sequence of events that produces end-to-end entanglement with the lowest fidelity must be one where all swaps are performed at the end of the time window, i.e., all swaps are performed when the oldest link reaches the cutoff time, just before it expires. If any of those swaps were performed earlier, the final fidelity would be larger. Such a sequence of events produces the lowest end-to-end fidelity when all the links are as old as possible, i.e., when their age is t_{cut}. If any of the links were younger, the end-to-end fidelity would be larger. Hence, the lowest end-to-end fidelity is achieved when all the links are generated simultaneously and all the swaps are performed when those links are t_{cut} time slots old. In this case, the fidelity of each link before swapping is given by (2.12):

$$F_{\text{old}} = \frac{1}{4} + \left(F_{\text{new}} - \frac{1}{4}\right)e^{-\frac{t_{\text{cut}}}{\tau}}, \tag{2.21}$$

where F_{new} is the fidelity of newly generated elementary links. The final fidelity after swapping all the links can be calculated using (2.20):

$$F_{\text{worst}} = \frac{1}{4} \cdot \left[1 + \frac{(4F_{\text{old}} - 1)^{n-1}}{3^{n-2}}\right]. \tag{2.22}$$

Finally, we impose that the worst-case end-to-end fidelity must be larger than the desired minimum fidelity F_{min}: $F_{\text{worst}} \geq F_{\text{min}}$. Solving for t_{cut} yields an explicit condition for the cutoff time:

$$t_{\text{cut}} \leq -\tau \ln\left(\frac{3}{4F_{\text{new}}-1}\left(\frac{4F_{\text{min}}-1}{3}\right)^{\frac{1}{n-1}}\right). \tag{2.23}$$

When this condition is satisfied, every sequence of events will lead to a large enough fidelity. Consequently, any policy that we implement on the repeater chain will also deliver entanglement with a large enough fidelity.

2.8. [APPENDIX] FURTHER COMMENTS ON THE OPTIMAL EXPECTED DELIVERY TIME

Here we provide the expected delivery time of optimal policies in three- and four-node repeater chains. Then, we compare optimal policies to the swap-asap policy in a four-node chain. We also show that, in longer chains, the relative difference in expected delivery time is not always monotonic with the probability of successful entanglement generation p.

Figure 2.9 shows the expected delivery time of an optimal policy, T_{opt}, in three- and four-node chains, versus p for different values of p_{s} and t_{cut}. The relation between T_{opt} and the rest of the variables is similar to that of the five-node chain discussed in the main text. When p is small, more entanglement generation attempts are required to succeed, yielding a larger T_{opt}. Decreasing p_{s} also increases T_{opt}, since more attempts at entanglement swapping are required on average. When t_{cut} is small, all entangled states must be generated within a small time window and therefore T_{opt} is also larger.

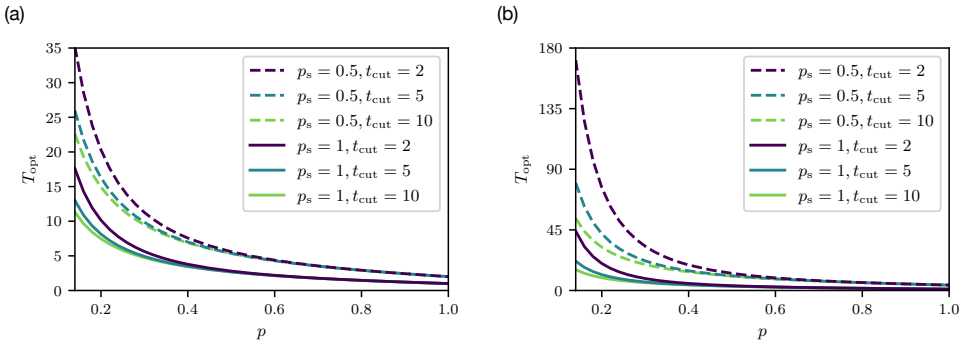

Figure 2.9: **The expected delivery time increases with lower p, p_{s}, and t_{cut}.** Expected delivery time of an optimal policy, T_{opt}, versus p for (a) $n = 3$ and (b) $n = 4$ and different values of cutoff ($t_{\text{cut}} = 2, 5, 10$). Solid lines correspond to deterministic swaps ($p_{\text{s}} = 1$) and dashed lines correspond to probabilistic swaps with $p_{\text{s}} = 0.5$.

In three-node chains, the swap-asap policy is always optimal since there is no reason to wait after both links have been generated. As the number of nodes increases, policies have more degrees of freedom that can be adjusted to get an improvement over the swap-asap policy. Figure 2.10 shows the advantage in expected delivery time of an optimal

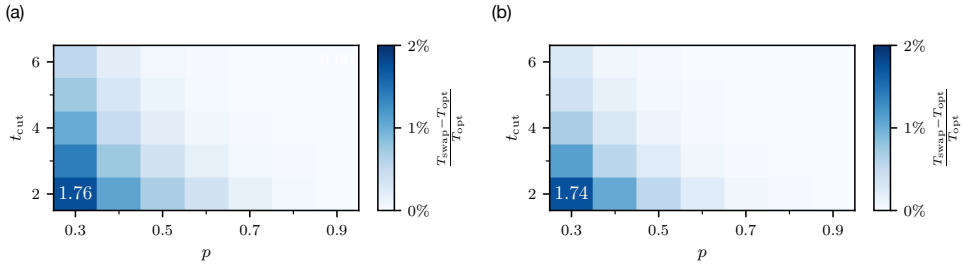

Figure 2.10: **The swap-asap policy is close to optimal in four-node chains.** Relative difference between the expected delivery times of an optimal policy, T_{opt}, and the swap-asap policy, T_{swap}, in a four-node chain, for different values of p and t_{cut}.

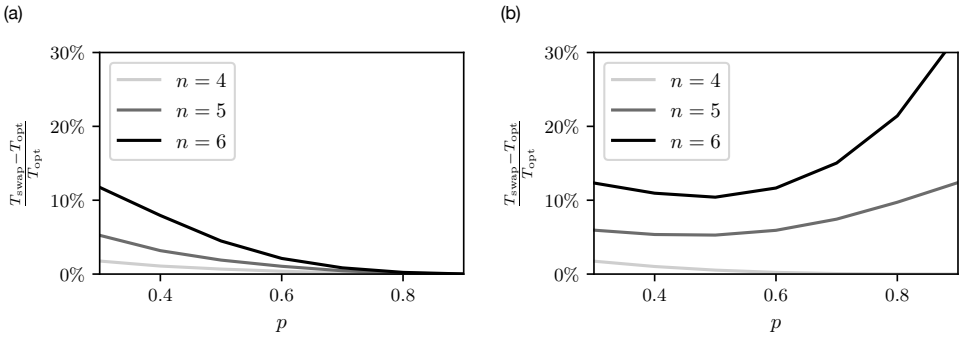

Figure 2.11: **The advantage provided by an optimal policy over swap-asap is not always monotonic with p.** Relative difference between the expected delivery times of an optimal policy, T_{opt}, and the swap-asap policy, T_{swap}, in an n-node chain with $t_{\text{cut}} = 2$, for different values of n and p.

policy versus the swap-asap policy in four-node chains. The swap-asap policy is no longer optimal, as it was in three-node chains, although the largest advantage observed is below 2%, meaning that the swap-asap policy is still close to optimal. The advantage over swap-asap increases up to 30% in five- and six-node chains, as shown in Figure 2.8 (main text) and in Figure 2.11.

Figure 2.11 shows the advantage of an optimal policy over swap-asap in terms of expected delivery time, versus p and for different number of nodes. When swaps are deterministic, the advantage is larger for smaller p. The reason is that links are harder to generate as p approaches zero, and therefore a fine-tuned policy that makes better use of those scarce resources is expected to be increasingly better than a greedy policy like swap-asap. When swaps are probabilistic and $n > 4$, the advantage is not monotonic in p anymore, as can clearly be seen for $n = 6$, $t_{\text{cut}} = 2$, and $p_{\text{s}} = 0.5$. On the one hand, the advantage increases when p approaches zero, due to swap-asap making an inefficient use of the links, which become a scarce resource. On the other hand, when p approaches one, the advantage also increases, due to the effect of full states, as discussed in the main text.

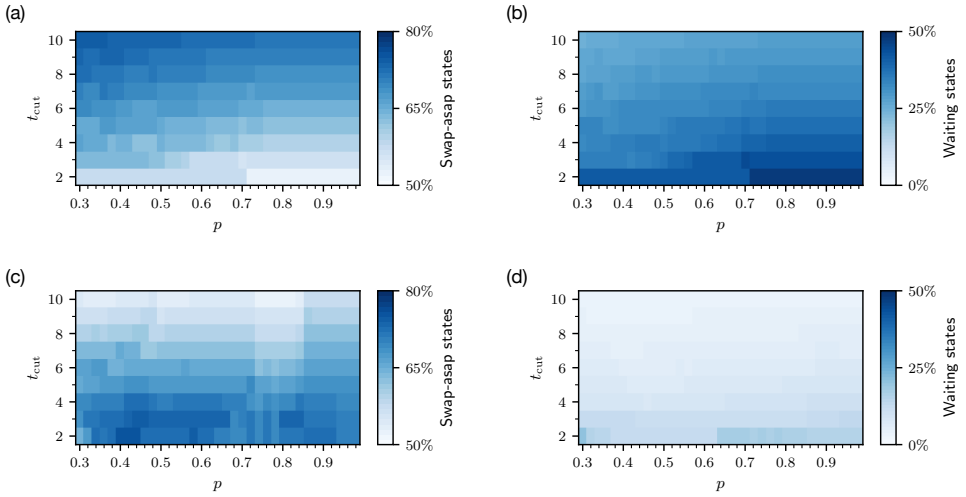

Figure 2.12: **An optimal policy acts as the swap-asap policy in a large number of states.** Percentage of states in which the optimal policy found by our solver decides to (a,c) perform all possible swaps or (b,d) not perform any swap, in a five-node repeater chain with (a-b) $p_s = 1$ or (c-d) $p_s = 0.5$. We only consider states in which at least one swap can be performed.

2.9. [APPENDIX] ACTIONS CHOSEN BY OPTIMAL POLICIES

Figure 2.12 shows the percentage of states in which an optimal policy decides to perform all possible swaps (acting as the swap-asap policy) or not perform any swap at all, in a five-node repeater chain. For this analysis, we only consider states in which at least one swap can be performed. Although there seem to be some clear trends, these results cannot be used to determine how close the swap-asap policy is to being optimal, since:

(i) We found one of the possibly many optimal policies, so the swap-asap policy may be closer to a different optimal policy. This also explains why the plots in Figure 2.12 are not monotonic.

(ii) Even if there is only one state in which two policies differ, this state could have a large impact on the expected delivery time, as explained in the example of full states in the main text.

In Figure 2.13 we plot the same quantities for increasing number of nodes. The percentage of states in which the optimal policy decides to perform all possible swaps, acting as the swap-asap policy, decreases with increasing n. This agrees with the fact that the advantage provided by an optimal policy in terms of expected delivery time over the swap-asap policy increases with increasing n, as shown in the main text. However, the data from Figure 2.13 alone should not be used to draw any conclusions, since arguments (i) and (ii) also apply to these plots.

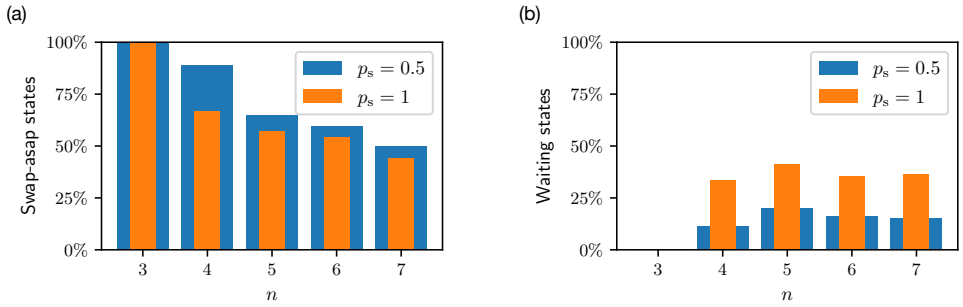

Figure 2.13: **The percentage of states in which the optimal policy acts as the swap-asap policy decreases in longer chains.** Percentage of states in which the optimal policy found by our solver decides to (a) perform all possible swaps or (b) not perform any swap, in a repeater chain with $p = 0.3$ and $t_{\text{cut}} = 2$. We only consider states in which at least one swap can be performed.

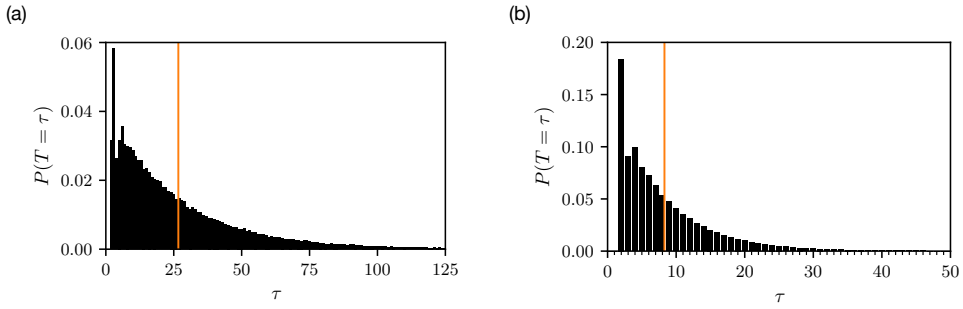

Figure 2.14: **The delivery time distribution can be heavy-tailed.** Delivery time distribution after simulating an optimal policy in a five-node repeater chain with $p_s = 0.5$, $t_{\text{cut}} = 2$, and (a) $p = 0.5$ or (b) $p = 0.9$. The number of samples is 10^5. Solid orange lines correspond to the expected delivery time of the optimal policy.

2.10. [APPENDIX] DELIVERY TIME DISTRIBUTION

Here, we show two examples of repeater chains in which the entanglement delivery time distribution is heavy-tailed. Figure 2.14 shows the delivery time distribution in a five-node chain with $p_s = 0.5$, $t_{\text{cut}} = 2$, and $p = 0.5$ (Figure 2.14a) or $p = 0.9$ (Figure 2.14b). The results shown here have been calculated by repeatedly simulating the optimal policy in a repeater chain (source code available at https://github.com/AlvaroGI/optimal-homogeneous-chain). As shown in the figure, the distribution is heavy-tailed for some combinations of parameters. In those cases, the average value does not provide an accurate description of the whole distribution.

2.11. [APPENDIX] EXPECTED TIME TO REACH AN ABSORBING STATE

In this appendix, we show that the expected time required to reach an absorbing state in a discrete Markov decision process (MDP) starting from state $\boldsymbol{s}$ and following policy π, $T_\pi(\boldsymbol{s})$, satisfies

$$T_\pi(\boldsymbol{s}) = 1 + \sum_{\boldsymbol{s}' \in \mathcal{S}} P(\boldsymbol{s}'|\boldsymbol{s},\pi) \cdot T_\pi(\boldsymbol{s}'),$$

where $\mathcal{S}$ is the state space and $P(\boldsymbol{s}'|\boldsymbol{s},\pi)$ is the probability of transition from state $\boldsymbol{s}$ to state $\boldsymbol{s}'$ when following policy π. We also discuss the difference between deterministic and stochastic policies.

Let $t_\pi(\boldsymbol{s})$ be the time required to reach an absorbing state starting from state $\boldsymbol{s}$ in one realization of the process, and let

$$T_\pi(\boldsymbol{s}) \equiv \mathbb{E}[t_\pi(\boldsymbol{s})] = \sum_{m=0}^{\infty} m \Pr[t_\pi(\boldsymbol{s}) = m] \tag{2.24}$$

be its expected value. The time required to reach an absorbing state starting from $\boldsymbol{s}$ can be calculated as the time required to go from state $\boldsymbol{s}$ to any state $\boldsymbol{s}'$ plus the time required to go from $\boldsymbol{s}'$ to an absorbing state. Since the Markov chain is discrete, we can write this as

$$\Pr[t_\pi(\boldsymbol{s}) = m] = \sum_{\boldsymbol{s}' \in \mathcal{S}} P(\boldsymbol{s}'|\boldsymbol{s},\pi) \Pr\big[t_\pi(\boldsymbol{s}') = m-1\big]. \tag{2.25}$$

The recursive relation for $T_\pi(\boldsymbol{s})$ can be derived as follows:

$$\begin{aligned}
T_\pi(\boldsymbol{s}) &\overset{a}{=} \sum_{m=0}^{\infty} m \Pr[t_\pi(\boldsymbol{s}) = m] && (2.26)\\
&\overset{b}{=} \sum_{m=0}^{\infty} m \sum_{\boldsymbol{s}' \in \mathcal{S}} P(\boldsymbol{s}'|\boldsymbol{s},\pi) \Pr\big[t_\pi(\boldsymbol{s}') = m-1\big] && (2.27)\\
&= \sum_{\boldsymbol{s}' \in \mathcal{S}} P(\boldsymbol{s}'|\boldsymbol{s},\pi) \sum_{m=0}^{\infty} m \Pr\big[t_\pi(\boldsymbol{s}') = m-1\big] && (2.28)\\
&= \sum_{\boldsymbol{s}' \in \mathcal{S}} P(\boldsymbol{s}'|\boldsymbol{s},\pi) \sum_{m=-1}^{\infty} (m+1) \Pr\big[t_\pi(\boldsymbol{s}') = m\big] && (2.29)\\
&= \sum_{\boldsymbol{s}' \in \mathcal{S}} P(\boldsymbol{s}'|\boldsymbol{s},\pi) \sum_{m=0}^{\infty} (m+1) \Pr\big[t_\pi(\boldsymbol{s}') = m\big] && (2.30)\\
&= \sum_{\boldsymbol{s}' \in \mathcal{S}} P(\boldsymbol{s}'|\boldsymbol{s},\pi) \sum_{m=0}^{\infty} m \Pr\big[t_\pi(\boldsymbol{s}') = m\big] + \sum_{\boldsymbol{s}' \in \mathcal{S}} P(\boldsymbol{s}'|\boldsymbol{s},\pi) \sum_{m=0}^{\infty} \Pr\big[t_\pi(\boldsymbol{s}') = m\big] && (2.31)\\
&\overset{c}{=} \sum_{\boldsymbol{s}' \in \mathcal{S}} P(\boldsymbol{s}'|\boldsymbol{s},\pi) \sum_{m=0}^{\infty} m \Pr\big[t_\pi(\boldsymbol{s}') = m\big] + 1 && (2.32)\\
&\overset{d}{=} \sum_{\boldsymbol{s}' \in \mathcal{S}} P(\boldsymbol{s}'|\boldsymbol{s},\pi) T_\pi(\boldsymbol{s}') + 1, && (2.33)
\end{aligned}$$

with the following steps:

a. We apply Equation (2.24).

b. We apply Equation (2.25).

c. We employ the normalization of the probability distributions: $\sum_{m=0}^{\infty} \Pr\left[t_\pi(\boldsymbol{s}') = m\right] = 1$ and $\sum_{\boldsymbol{s}' \in \mathscr{S}} P(\boldsymbol{s}'|\boldsymbol{s}, \pi) = 1$.

d. We use Equation (2.24) again.

In the previous derivation, we have implicitly assumed that the policy is deterministic: at each state $\boldsymbol{s}$, the action chosen is always $\pi(\boldsymbol{s})$. It can be shown that, in an MDP with a finite and countable set of states, there exists at least one optimal policy that is deterministic (see Section 2.3 from [176]). Therefore, since we are solving a finite MDP, we only need to consider deterministic policies. Optimal random policies can be built by combining several deterministic optimal policies, provided that there is more than one.

When considering stochastic policies, π is no longer a mapping from a state to an action but a mapping from a state to a probability distribution over the action space. The previous derivation remains valid for stochastic policies, although in that case the transition probabilities must be written as

$$P(\boldsymbol{s}'|\boldsymbol{s}, \pi) = \sum_{a \in \mathscr{A}} \pi(a|\boldsymbol{s}) P(\boldsymbol{s}'|\boldsymbol{s}, a), \tag{2.34}$$

where $\mathscr{A}$ is the action space and $\pi(a|\boldsymbol{s})$ is the probability of choosing action a in state $\boldsymbol{s}$ when following policy π.

2.12. [APPENDIX] DYNAMIC PROGRAMMING ALGORITHMS

To find optimal policies, we formulate a Markov decision process that results in the Bellman equations, as explained in the main text. These equations can be solved using a dynamic programming algorithm, such as value iteration and policy iteration. Both algorithms start with arbitrary values of $T_\pi(\boldsymbol{s})$ (for some policy π and $\forall \boldsymbol{s} \in \mathscr{S}$, where $\mathscr{S}$ is the state space) and they iteratively update the policy π and the values $T_\pi(\boldsymbol{s})$, $\forall \boldsymbol{s} \in \mathscr{S}$. The updated policy is guaranteed to converge to an optimal policy π^* in a finite number of iterations in policy iteration and an infinite number of iterations in value iteration (see Sections 4.3 and 4.4 from [175]). Note that there might be multiple optimal policies, although this approach finds only one of them. In practice, the algorithms stop when the updated values differ by not more than some $\varepsilon > 0$ from the values in the previous iteration. All our results have been calculated using $\varepsilon = 10^{-7}$. In this work, we have applied both value iteration and policy iteration, which provided the same results (our specific implementations can be found at https://github.com/AlvaroGI/optimal-homogeneous-chain). For a detailed explanation of both algorithms, see Sections 4.3 and 4.4 from [175].

In terms of computational cost, policy iteration is generally faster since less iterations are required. To the best of our knowledge, there are no known tight bounds on the number of iterations until convergence. However, the computational complexity of a single iteration in policy iteration is $\mathscr{O}(|\mathscr{A}||\mathscr{S}|^2 + |\mathscr{S}|^3)$, where $\mathscr{A}$ is the action space and $\mathscr{S}$ is the state space, which can be prohibitive for some combinations of parameters [99].

The computational complexity of one iteration in value iteration is $\mathcal{O}(|\mathcal{A}||\mathcal{S}|^2)$ [99]. In our problem, the complexity of each iteration increases exponentially with increasing number of nodes and polynomially with increasing cutoff time (see Appendix 2.14), and the number of iterations increases with decreasing probability of entanglement generation and decreasing probability of successful swap, since the estimate of the values is worse when these probabilities are small. Consequently, to study long chains with large cutoffs and small probabilities of successful entanglement generation and swap, one may need to employ approximate methods, such as deep reinforcement learning, which can find sub-optimal but good enough policies at a lower computational cost.

2.13. [APPENDIX] MARKOV DECISION PROCESS EXAMPLE

Here we provide an example of how to formulate the Markov decision process (MDP) for a three-node repeater chain with cutoff $t_{\text{cut}} = 1$. Specifically, we calculate each term in the Bellman equations, which can then be used to find an optimal policy, as explained in the main text.

We start by listing all the states in which the chain can be found. The sequence of events during each time slot is the following:

1. First, the ages of all entangled links are increased by 1.
2. Second, entanglement generation is attempted between every pair of neighbors with qubits available.
3. Third, entanglement swaps can be performed.
4. Lastly, links whose age is equal to t_{cut} are removed. End-to-end links are not removed.

Since the cutoff is 1, the ages of all links are at most 1. All possible states are listed in Figure 2.15.

Let us now find the equation for $T_\pi(\boldsymbol{s}_0)$, the expected delivery time from state $\boldsymbol{s}_0$, which is given by

$$T_\pi(\boldsymbol{s}_0) = 1 + \sum_{\boldsymbol{s}' \in \mathcal{S}} P(\boldsymbol{s}'|\boldsymbol{s}_0, \pi) \cdot T_\pi(\boldsymbol{s}'),$$

as explained in the main text and derived in 2.11. For clarity, let us abuse notation and write T_i to denote $T_\pi(\boldsymbol{s}_i)$. We can find each term by considering each possible scenario separately:

- With probability $(1-p)^2$, no links are successfully generated and the state remains $\boldsymbol{s}_0$. Swaps and cutoffs do not apply to this state. This contributes with a term $(1-p)^2 T_0$.
- With probability $p(1-p)$, only one of the links is generated and the state becomes either $\boldsymbol{s}_1$ or $\boldsymbol{s}_2$. Swaps and cutoffs do not apply to these states. This contributes with $p(1-p)T_1 + p(1-p)T_2$.

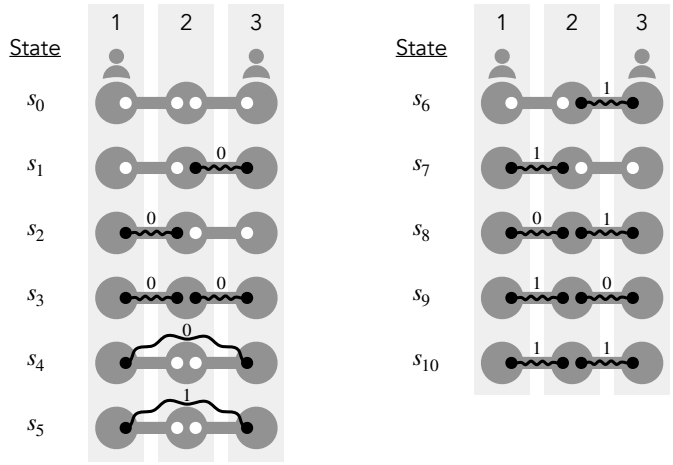

Figure 2.15: **There are eleven possible states in a three-node repeater chain with cutoff $t_{\text{cut}} = 1$.** Nodes are labeled 1 to 3 from left to right.

- With probability p^2, both links are generated and the state becomes $\boldsymbol{s}_3$. Then, a swap can be performed and the last term splits into two contributions:
  - If the policy decides to perform a swap in node 2, i.e., $\pi(\boldsymbol{s}_3) = \{2\}$, the state at the end of the time slot will be $\boldsymbol{s}_4$ if the swap is successful, and $\boldsymbol{s}_0$ if the swap fails. These scenarios contribute with $p^2 \mathbb{1}_{\pi(\boldsymbol{s}_3)=\{2\}} \big(p_{\text{s}} T_4 + (1-p_{\text{s}}) T_0\big)$, where $\mathbb{1}_A$ is the indicator function that takes value 1 if A is true and value 0 otherwise.
  - If the policy decides to not perform the swap, i.e. $\pi(\boldsymbol{s}_3) = \emptyset$, the state remains $\boldsymbol{s}_3$. The contribution is then $p^2 \mathbb{1}_{\pi(\boldsymbol{s}_3)=\emptyset} T_3$.

Now, we can write all the terms above in a single equation:

$$T_0 = 1 + (1-p)^2 T_0 + p(1-p) T_1 + p(1-p) T_2 + p^2 \mathbb{1}_{\pi(\boldsymbol{s}_3)=\{2\}} \big(p_{\text{s}} T_4 + (1-p_{\text{s}}) T_0\big) + p^2 \mathbb{1}_{\pi(\boldsymbol{s}_3)=\emptyset} T_3. \tag{2.35}$$

Note that $T_4 = T_5 = 0$, since $\boldsymbol{s}_4$ and $\boldsymbol{s}_5$ are absorbing states. Rearranging terms, we obtain

$$T_0 = 1 + \Big[(1-p)^2 + p^2 \mathbb{1}_{\pi(\boldsymbol{s}_3)=\{2\}} (1-p_{\text{s}})\Big] T_0 + \Big[p(1-p)\Big] T_1 + \Big[p(1-p)\Big] T_2 + \Big[p^2 \mathbb{1}_{\pi(\boldsymbol{s}_3)=\emptyset}\Big] T_3. \tag{2.36}$$

Let us now find the equation for $\boldsymbol{s}_1$. At the beginning of the time slot, the age of the link is increased by 1 and the state becomes $\boldsymbol{s}_6$. After that:

- With probability $(1-p)$, no links are successfully generated. Then, the only existing link is removed since it is 1 time slot old. This contributes with a term $(1-p) T_0$.
- With probability p, the remaining link is generated and the state becomes $\boldsymbol{s}_8$. Then, a swap can be performed:
  - If the policy decides to perform a swap in node 2, i.e., $\pi(\boldsymbol{s}_8) = \{2\}$, the state at the end of the time slot will be $\boldsymbol{s}_5$ if the swap is successful, and $\boldsymbol{s}_0$ if the swap fails. These scenarios contribute with $p \mathbb{1}_{\pi(\boldsymbol{s}_8)=\{2\}} \big(p_{\text{s}} T_5 + (1-p_{\text{s}}) T_0\big)$.

– If the policy decides to not perform the swap, i.e. $\pi(\boldsymbol{s}_8) = \emptyset$, the state remains $\boldsymbol{s}_8$, and the link with age 1 is removed afterwards. The state becomes $\boldsymbol{s}_2$ and the contribution is then $p\mathbb{1}_{\pi(\boldsymbol{s}_8)=\emptyset} T_2$.

Combining all terms, we obtain

$$T_1 = 1 + \Big[(1-p) + p\mathbb{1}_{\pi(\boldsymbol{s}_8)=\{2\}}(1-p_s)\Big] T_0 + \Big[p\mathbb{1}_{\pi(\boldsymbol{s}_8)=\emptyset}\Big] T_2, \tag{2.37}$$

where we have used that $T_5 = 0$.

Due to the symmetry of the problem,

$$T_2 = T_1, \tag{2.38}$$

so we do not need to derive a new equation for T_2.

Lastly, we find the equation for $\boldsymbol{s}_3$. At the beginning of the time slot, the age of each link is increased by 1 and the state becomes $\boldsymbol{s}_{10}$, in which no more links can be generated. After that, a swap can be performed:

– If the policy decides to perform a swap in node 2, i.e., $\pi(\boldsymbol{s}_{10}) = \{2\}$, the state at the end of the time slot will be $\boldsymbol{s}_5$ if the swap is successful, and $\boldsymbol{s}_0$ if the swap fails. These scenarios contribute with $\mathbb{1}_{\pi(\boldsymbol{s}_{10})=\{2\}}\big(p_s T_5 + (1-p_s) T_0\big)$.

– If the policy decides to not perform the swap, i.e. $\pi(\boldsymbol{s}_{10}) = \emptyset$, the state remains $\boldsymbol{s}_{10}$, and both links are removed afterwards, when cutoffs are applied. The state becomes $\boldsymbol{s}_0$ and the contribution is then $\mathbb{1}_{\pi(\boldsymbol{s}_{10})=\emptyset} T_0$.

The equation then reads

$$T_3 = 1 + \Big[\mathbb{1}_{\pi(\boldsymbol{s}_{10})=\{2\}}(1-p_s) + \mathbb{1}_{\pi(\boldsymbol{s}_{10})=\emptyset}\Big] T_0, \tag{2.39}$$

where we have used that $T_5 = 0$.

We can write Equations (2.36), (2.37), (2.38), and (2.39) as

$$\begin{cases} T_0 = 1 + \Big[(1-p)^2 + p^2\mathbb{1}_{\pi(\boldsymbol{s}_3)=\{2\}}(1-p_s)\Big] T_0 + 2\Big[p(1-p)\Big] T_1 + \Big[p^2\mathbb{1}_{\pi(\boldsymbol{s}_3)=\emptyset}\Big] T_3, \\ T_1 = 1 + \Big[(1-p) + p\mathbb{1}_{\pi(\boldsymbol{s}_8)=\{2\}}(1-p_s)\Big] T_0 + \Big[p\mathbb{1}_{\pi(\boldsymbol{s}_8)=\emptyset}\Big] T_1, \\ T_3 = 1 + \Big[\mathbb{1}_{\pi(\boldsymbol{s}_{10})=\{2\}}(1-p_s) + \mathbb{1}_{\pi(\boldsymbol{s}_{10})=\emptyset}\Big] T_0. \end{cases} \tag{2.40}$$

An optimal policy π^* can be found by minimizing T_0, T_1, and T_3 in this system of equations. This can be done, e.g., using iterative algorithms such as value and policy iteration, as discussed in the main text. In this case, it can be shown that the swap-asap policy is optimal, i.e., $\pi^*(\boldsymbol{s}_3) = \pi^*(\boldsymbol{s}_8) = \pi^*(\boldsymbol{s}_{10}) = \{2\}$. This also makes sense intuitively: once both links are generated, waiting provides no advantage in terms of delivery time over performing the swap immediately. For this policy, the system of equations becomes

$$\begin{cases} T_0 = 1 + \Big[(1-p)^2 + p^2(1-p_s)\Big] T_0 + 2\Big[p(1-p)\Big] T_1, \\ T_1 = 1 + \Big[(1-pp_s)\Big] T_0, \\ T_3 = 1 + \Big[(1-p_s)\Big] T_0, \end{cases} \tag{2.41}$$

which yields an optimal expected delivery time of

$$T_0 = \frac{1+2p(1-p)}{1-(1-p)^2-p^2(1-p_s)-2p(1-p)(1-pp_s)}.$$

As a final remark, note that the expected delivery times from states $\boldsymbol{s}_6$ to $\boldsymbol{s}_{10}$ were not necessary to compute T_0. In fact, states $\boldsymbol{s}_6$ to $\boldsymbol{s}_{10}$ cannot exist at the beginning of a time slot, since the links that exist at the beginning of a time slot are always younger than t_{cut} (i.e., their age is 0) or are end-to-end links. This is the reason why we do not need to optimize over T_6 to T_{10}.

2.14. [APPENDIX] SCALING OF THE NUMBER OF STATES

In this appendix, we find a lower bound to the number of states in the Markov decision process discussed in the main text, and show that it scales as $\Omega\big((t_{\text{cut}})^{n-2}\big)$. Then, we compare this lower bound to the exact number of states for some combinations of parameters.

We start by calculating the lower bound. Let us define $\mathscr{S}(l)$ as the set of states in which only l entangled links are present. By definition,

$$\mathscr{S} = \bigcup_l \mathscr{S}(l).$$

The sets $\mathscr{S}(l)$ do not overlap. Therefore,

$$|\mathscr{S}| = \sum_l |\mathscr{S}(l)|. \tag{2.42}$$

The first term is given by

$$|\mathscr{S}(0)| = 1, \tag{2.43}$$

since there is only one state without any entangled links.

Since any two nodes could potentially share an entangled link, there are

$$k = \binom{n}{2} - 1 = \frac{n^2-n-2}{2}$$

possible links in a repeater chain with n nodes (note that we subtract 1 from the combinatorial number because there is no need to represent end-to-end links). When one of those links exists, its age can be anything from 0 to t_{cut}. Therefore, the total number of different states with only one link is given by

$$|\mathscr{S}(1)| = \frac{n^2-n-2}{2}(t_{\text{cut}}+1) \geq \frac{n^2-n-2}{2} t_{\text{cut}}. \tag{2.44}$$

Let us now consider a chain with two links where both are connected to the same node i. From node i towards one end of the chain, there are $i-1$ nodes. From i towards

the other end of the chain, there are $n-i$ nodes. Then, the number of states with two links where both are connected to node i is given by

$$|\mathscr{S}_i(2)| = (i-1)(n-i)(t_{\text{cut}}+1)^2,$$

where the last factor accounts for all possible ages of both links. We can find a lower bound to $|\mathscr{S}(2)|$ by considering only the states in which both links are connected to the same node i, i.e.,

$$\begin{aligned}
|\mathscr{S}(2)| &\geq \sum_{i=2}^{n-1} |\mathscr{S}_i(2)| \\
&= \sum_{i=2}^{n-1} (i-1)(n-i)(t_{\text{cut}}+1)^2 \\
&= \sum_{j=1}^{n-2} j(n-j-1)(t_{\text{cut}}+1)^2 \\
&= (t_{\text{cut}}+1)^2 \sum_{j=1}^{n-2} \left(-j^2+(n-1)j\right) \\
&= (t_{\text{cut}}+1)^2 \frac{n(n-1)(n-2)}{6} \\
&\geq \frac{n(n-1)(n-2)}{6} t_{\text{cut}}^2,
\end{aligned} \tag{2.45}$$

where we used the identities $\sum_{k=1}^{m} k = \frac{m(m+1)}{2}$ and $\sum_{k=1}^{m} k^2 = \frac{m(m+1)(2m+1)}{6}$ to simplify the sum.

Next, we compute a lower bound for $|\mathscr{S}(l)|$, $l>2$. Let us consider only states with l adjacent links, i.e., states that have $l-1$ nodes that hold 2 entangled links. The number of such states is lower bounded by the different ways in which we can pick those $l-1$ nodes from a total of $n-2$ nodes (end-nodes cannot have two links) and the possible ages of the l links. Therefore, we can bound $|\mathscr{S}(l)|$ as follows:

$$|\mathscr{S}(l)| \geq \binom{n-2}{l-1}(t_{\text{cut}}+1)^l \geq \binom{n-2}{l-1} t_{\text{cut}}^l, \quad l>2. \tag{2.46}$$

Finally, using Equations (2.42) to (2.46), we find a lower bound for $|\mathscr{S}|$:

$$\begin{aligned}
|\mathscr{S}| &= \sum_{l=0}^{2} |\mathscr{S}(l)| + \sum_{l=3}^{n-1} |\mathscr{S}(l)| \\
&\geq \sum_{l=0}^{2} |\mathscr{S}(l)| + \sum_{l=3}^{n-1} \binom{n-2}{l-1} t_{\text{cut}}^{l} \\
&\geq \sum_{l=0}^{2} |\mathscr{S}(l)| + t_{\text{cut}} \sum_{k=0}^{n-2} \binom{n-2}{k} t_{\text{cut}}^{k} - t_{\text{cut}} - (n-2)\, t_{\text{cut}}^{2} \\
&\overset{a}{\geq} \sum_{l=0}^{2} |\mathscr{S}(l)| + t_{\text{cut}} (t_{\text{cut}}+1)^{n-2} - t_{\text{cut}} - (n-2)\, t_{\text{cut}}^{2} \\
&\geq \sum_{l=0}^{2} |\mathscr{S}(l)| + t_{\text{cut}}^{n-1} - t_{\text{cut}} - (n-2)\, t_{\text{cut}}^{2} \\
&\geq 1 + \frac{n^2-n-2}{2} t_{\text{cut}} + \frac{n(n-1)(n-2)}{6} t_{\text{cut}}^{2} + t_{\text{cut}}^{n-1} - t_{\text{cut}} - (n-2)\, t_{\text{cut}}^{2}
\end{aligned}$$

where, in step a, we have used the binomial sum:

$$\sum_{k=0}^{n} \binom{n}{k} x^{k} = (1+x)^{n}.$$

After some algebra, we find

$$|\mathscr{S}| \geq 1 + \frac{n^2-n-4}{2} t_{\text{cut}} + \frac{(n^2-n-6)(n-2)}{6} t_{\text{cut}}^{2} + t_{\text{cut}}^{n-1}. \tag{2.47}$$

From the previous result, we conclude that the scaling of the number of states is

$$|\mathscr{S}| = \Omega\big((t_{\text{cut}})^{n-1}\big).$$

Figure 2.16 shows the exact number of states versus the cutoff time and the number of nodes, together with the lower bound (2.47). In these plots, the exact number of states corresponds to the size of the state space explored by our policy iteration algorithm.

DATA AND CODE AVAILABILITY

The data shown in this chapter can be found at [92]. Our code can be found in the following GitHub repository: https://github.com/AlvaroGI/optimal-homogeneous-chain.

AUTHOR CONTRIBUTIONS

ÁGI and GV defined the problem, the model, and the MDP formulation. ÁGI and GV, with support from LS, coded iterative methods to solve the MDP. ÁGI analyzed the results and was the main writer of the main text and appendices, which were also published as ref. [91]. SW provided active feedback at every stage of the project.

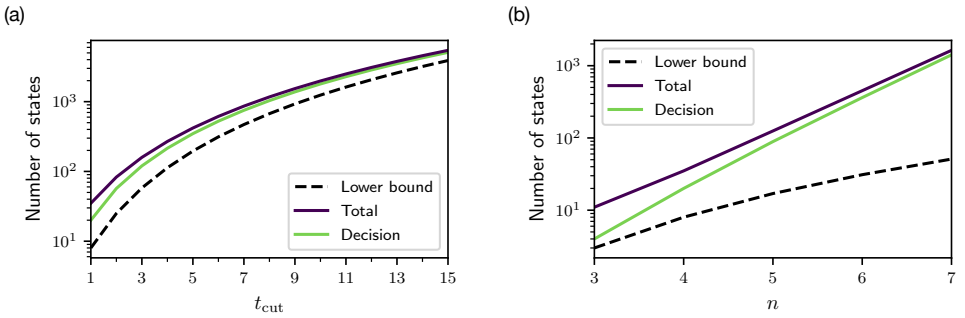

Figure 2.16: **The number of states scales at least exponentially with increasing n and polynomially with increasing t_{cut}.** (a) Number of states versus the cutoff time in a four-node chain, and (b) versus the number of nodes in a chain with cutoff $t_{\text{cut}} = 1$. Solid lines correspond to the number of states found by our policy iteration algorithm (note that the number of states only depends on n and t_{cut}). The purple solid line is the total number of states and the green line is the number of states in which a decision can be made (i.e., states in which at least one swap can be performed). The dashed line corresponds to the lower bound (2.47) to the total number of states.

3

Quantum Circuit Switching with One-Way Repeaters in Star Networks

Álvaro G. Iñesta, Hyeongrak Choi, Dirk Englund, and Stephanie Wehner

The kind-hearted person held the elevator door for him [...].
Out of kindness, she forced him to quicken his pace,
humiliating him.
— Ignatius Farray

Distributing quantum states reliably among distant locations is a key challenge in the field of quantum networks. One-way quantum networks address this by using one-way communication and quantum error correction. Here, we analyze quantum circuit switching as a protocol to distribute quantum states in one-way quantum networks. In quantum circuit switching, pairs of users can request the delivery of multiple quantum states from one user to the other. After waiting for approval from the network, the states can be distributed either sequentially, forwarding one at a time along a path of quantum repeaters, or in parallel, sending batches of quantum states from repeater to repeater. Since repeaters can only forward a finite number of quantum states at a time, a pivotal question arises: is it advantageous to send them sequentially (allowing for multiple requests simultaneously) or in parallel (reducing processing time but handling only one request at a time)? We compare

This chapter has been published separately in ref. [89].

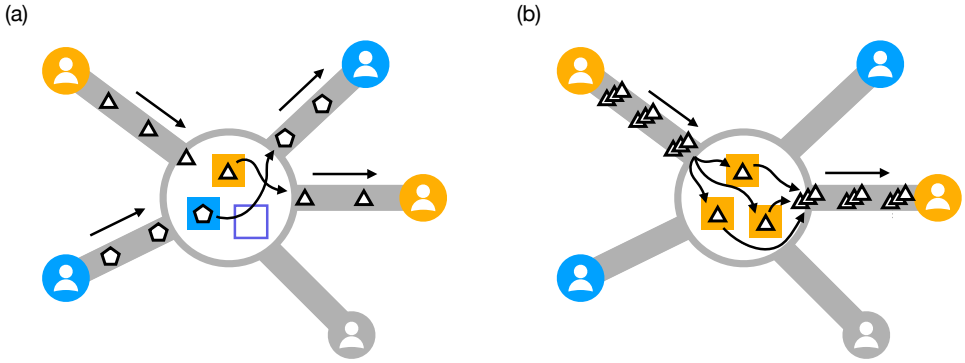

Figure 3.1: **In QCS, packets can be distributed (a) sequentially or (b) in parallel.** Illustration of a QCS scheme with (a) sequential and (b) parallel distribution of packets, in a star network with five users and a single central repeater (big circle). The repeater contains $k = 3$ forwarding stations (squares), therefore it can forward at most $k = 3$ quantum data packets (triangles and pentagons) at a time. In sequential distribution of packets, a single forwarding station is reserved to meet each of the requests submitted by the pairs of users in orange and blue. In parallel distribution, all three stations are used in parallel to meet a single request at a time.

both approaches in a quantum network with a star topology. Using tools from queuing theory, we show that requests are met at a higher rate when packets are distributed in parallel, although sequential distribution can generally provide service to a larger number of users simultaneously. We also show that using a large number of quantum repeaters to combat channel losses limits the maximum distance between users, as each repeater introduces additional processing delays. These findings provide insight into the design of protocols for distributing quantum states in one-way quantum networks.

3.1. INTRODUCTION

In the field of quantum networking, the efficient delivery of quantum information between distant users remains a central challenge [163]. A common strategy is to employ quantum repeaters to facilitate quantum communication among remote network nodes [132, 194]. Many repeater architectures require two-way communication and long-lived quantum memories, as they rely on the distribution of entanglement via heralded generation [9, 16] and entanglement swaps [55, 164, 209]. Conversely, the so-called *third-generation quantum repeaters* use quantum error correction to send quantum data using only one-way communication [18, 132, 133, 135]. Here, we focus on one-way quantum networks, which employ third-generation repeaters to distribute quantum states.

In one-way quantum networks, each quantum state that must be delivered to a remote location (e.g., a data qubit or half of an entangled pair) is encoded and sent over the network as a *quantum data packet*, which consists of the encoded state and some additional metadata (such as the packet destination) [54]. In classical networks, data packets are commonly distributed across the network according to a packet switching or a circuit switching protocol [119]. In packet switching [8, 51], users can send data packets at will, which are then forwarded from one router to the next one whenever possible.

Routers may store the packets until they are able to forward them. In circuit switching (see, e.g., [28]), users first request service to the network. Later, the network reserves one or multiple paths (also called circuits) where routers reserve dedicated resources to meet that request, and there is in principle no need to store data in intermediate routers. In classical networks, packet switching has found more success than circuit switching since it makes better use of the resources in large networks with conflicting routes and high traffic loads. In quantum networks, quantum data packets can also be distributed according to a packet switching [54] or a circuit switching protocol. Quantum circuit switching has been studied in the context of two-way quantum repeaters [4], but not in the context of one-way quantum networks, to the best of our knowledge.

The main challenge when designing a one-way protocol to distribute quantum states is decoherence: quantum states have a limited lifetime and cannot be stored indefinitely in intermediate quantum repeaters. This means that the protocol must ensure rapid delivery of the quantum data packets. Additionally, some multi-party applications may need to consume multiple quantum states simultaneously (e.g., to implement verifiable blind quantum computations, multiple qubits must be sent from a client to a server, where they must coexist while some operations are performed [49, 115]). In those cases, the quantum states must be distributed within some time window – otherwise, by the time the last state is distributed, the quality of the first state would have decayed too much due to decoherence. Consequently, one-way protocols must prevent large variations in the time between the delivery of successive quantum data packets to ensure all of them are delivered within a specific time window. We expect the pre-allocation of resources in circuit switching to enable the protocol to meet these timing constraints required to successfully distribute quantum states over a network, regardless of network traffic. In this work, we introduce *quantum circuit switching* (QCS) for one-way quantum networks, and we investigate how to allocate resources for each request to distribute quantum states.

Each quantum repeater has limited capabilities, namely, it can only forward a finite number of packets, k, at a time (see Figure 3.1). We then say that the repeater has k *forwarding stations*, which operate independently and can forward one packet at a time. When multiple requests need to use the same repeater, the circuit switching protocol can either (*i*) reserve one forwarding station per request and process up to k requests simultaneously (Figure 3.1a), or (*ii*) reserve all k forwarding stations to meet one request at a time (Figure 3.1b). For simplicity, we omit the intermediate case where more than one but less than k stations are reserved for each request. In the first approach, quantum data packets are distributed *sequentially*, while in the second approach packets are distributed in *parallel* (requiring some form of frequency multiplexing).

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

situations is it actually advantageous to distribute quantum packets in parallel instead of sequentially? That is, *when does parallel distribution provide shorter mean sojourn times?* Addressing this resource-allocation question early on is crucial for the design of efficient protocols for distributing quantum states within one-way quantum networks. Here, we answer the question assuming a star network: a simplified topology where all users are connected via a number of quantum repeaters to a central repeater (similar to a quantum switch architecture [185, 186]).

Our main contributions are the following:

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

Our main findings are the following:

- In star networks with sequential distribution of packets, we can increase the number of forwarding stations k to increase the number of users supported by the network (i.e., the maximum number of users that allow requests to be met within a finite amount of time), which scales as $\sim \sqrt{k}$. This is not possible in parallel distribution: the number of users is capped and cannot be increased indefinitely by adding more forwarding stations per repeater.
- Parallel distribution generally provides smaller mean sojourn times than sequential distribution.
- There exists a trade-off between the number of users and the physical distances between them: too many users and too long distances overload the system and yield infinite waiting times.
- When there is a large number of users, adding more intermediate repeaters to combat channel losses and minimize the probability of packet loss does not allow the users to be located further away. In fact, adding more repeaters limits the size of the network to smaller scales, since each repeater introduces additional delays.

In Section 3.2, we explain the problem setup: the quantum network and quantum data packets model, the requests model, and the quantum circuit switching scheme. In 3.3, we present our main results: we analyze the operating regimes of both packet distribution strategies and compare their performance. We do this in a variety of scenarios, considering different quantum repeater architectures based on the financial budget available. Lastly, in 3.4, we discuss the limitations and the implications of our work.

Table 3.1: **Parameters of a quantum network running a QCS protocol.**

Physical topology (star network)	
u	Number of users
L	Distance between each user and the central repeater
N	Number of repeaters between each user and the central repeater
Hardware	
k	Number of forwarding stations per repeater
t_{fwd}	Forwarding time per forwarding station and quantum data packet
p	Probability of successful packet delivery from user to user
c	Speed of light in the physical channels
Requests	
n	Number of quantum data packets per request
w	Request time window
λ_0	Request submission rate per pair of users
Quantum Circuit Switching	
m	Number of packets of the same request distributed concurrently ($m = 1$ for sequential distribution and $m = k$ for parallel distribution)

3.2. PROBLEM SETUP

In this section, we present the problem setup. In 3.2.1, we discuss our model of quantum network and quantum data packets. In 3.2.2, we explain how the nodes submit requests and how to measure the performance of the network in meeting those requests. Lastly, we formalize the concept of quantum circuit switching in 3.2.3. We provide a summary of all the parameters introduced in this section in Table 3.1.

3.2.1. QUANTUM NETWORKS AND QUANTUM DATA PACKETS

We consider a quantum network with *third-generation quantum repeaters* [18, 132, 135, 136]. These type of repeaters use quantum error-correction to distribute quantum information using one-way communication, as opposed to first- and second-generation repeaters, which use heralded entanglement generation and two-way communication [136]. To forward quantum data, a repeater must first decode an incoming logical state (generally shared as a collection of photons) and correct any errors, then reencode the state again into multiple physical qubits, and finally send the encoded state to the next repeater.

We make the following assumptions about the quantum network:

- We consider a quantum network with a *star topology*, where u users are connected to a central repeater (see Figure 3.2). This is the simplest case of network where paths between users intersect, leading to shared resources and routing conflicts.

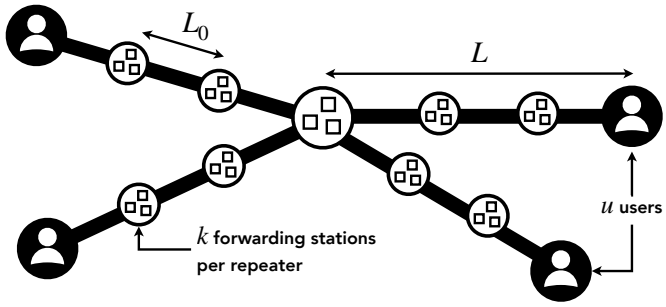

Figure 3.2: **Illustration of a star quantum network.** u users are at distance L from a central repeater. There are N repeaters between each user and the central repeater ($N = 2$ in the figure), and each of them has k forwarding stations (squares). The spacing between adjacent nodes is $L_0 = L/(N+1)$.

- For simplicity, we assume that all users are at the same physical distance L from the central repeater (see Figure 3.2). Moreover, there are N repeaters between each user and the central repeater. All repeaters are placed at distance $L_0 = L/(N+1)$ from each other.
- Users can encode and decode quantum data packets, according to some quantum code. A *quantum data packet* consists of a logical qubit and a classical header containing some metadata, as proposed in ref. [54]. The logical qubit consists of multiple physical qubits, generally in the form of photons. The classical header provides relevant information about the routing of the data packet (e.g., the destination of the packet).
- Each repeater has k *forwarding stations* that can decode and encode according to the same quantum code. Each station can receive/send physical qubits from/to any node. Moreover, each station can only forward (i.e., decode, reencode, and send) one quantum data packet at a time, which takes time t_{fwd}. After forwarding a packet, the station can be immediately used to forward another packet from any source node to any destination node (i.e., there is no downtime).
- Physical qubits can suffer from noise in the encoding and decoding circuits and in the physical channels. We assume that each packet can be successfully decoded at destination with probability p (this includes the case in which the encoded qubits suffered no errors and also the case in which they suffered a correctable number of errors). With probability $1-p$, a number of uncorrectable errors will be detected at destination and the packet will be discarded. We consider the probability of a logical error being unnoticed to be negligible, since we assume the main source of errors to be photon loss in the channels, which can be detected.

3.2.2. REQUESTS AND PERFORMANCE MEASURES

In many quantum network applications, multiple copies of a quantum state need to be distributed over a short interval of time. For example, verifiable blind quantum

computing requires multiple qubits to be sent from a client to a server, where they must coexist while some operations are performed [49, 115]. Another example is verifiable quantum secret sharing [46], where the parties involved want to verify that the dealer successfully distributed a quantum secret among them. This verification requires the dealer to distribute multiple ancillary quantum states to each party that will be consumed simultaneously (note that other proposals perform the verification stage sequentially [120]). In these examples, a number of quantum states must be sent from one network node to another over a short interval of time due to decoherence – otherwise, when the last state is distributed, the quality of the state that was distributed first would have decayed too much and would not be useful anymore.

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

We make the following additional assumptions about the requests:

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

$$\lambda = \binom{u}{2}\lambda_0 = \frac{u(u-1)}{2}\lambda_0, \tag{3.1}$$

  where u is the number of users.

The performance of protocols for quantum state distribution is typically measured with the quality of the distributed states and/or the distribution rate (see, e.g., [6, 91, 102, 140]). In our model, quantum data packets are either successfully distributed or a failure flag is raised, and therefore there is no need to study the quality of the state. The main quantity of interest in our problem is therefore the rate at which (n, w)-requests are met. In particular, to measure the performance of the protocol, we propose the *mean sojourn time*: the average time passed since a request is submitted until it is met (i.e., until the n-th quantum packet is successfully delivered within the time window w). The sojourn time of a request can be computed as the sum of two main contributions:

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

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

$$\mathbb{E}\left[T_{\text{sojourn}}\right] = \mathbb{E}\left[T_{\text{wait}}\right] + \mathbb{E}\left[T_{\text{service}}\right]. \tag{3.2}$$

In this work, we implicitly refer to the steady-state mean values when we discuss the mean sojourn time, the mean waiting time, and the mean service time. The steady state

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

We now summarize how to compute each term from (3.2):

- To compute the mean waiting time, we model the system as an M/G/s queue – this is a queue model where request arrivals follow a Poisson distribution and therefore are Markovian (M), the service time follows a general distribution (G), and a maximum of s requests can be processed simultaneously. This formulation allows us to derive analytical solutions and approximations for the mean waiting time in a variety of situations, although in some cases Monte Carlo methods are required. See Appendices 3.5 and 3.6 for further details.

- The service time distribution is hard to compute analytically. The reason is that calculating the number of packets that need to be sent until n are successfully received over a time window w is a challenging task [49]. When packets are sent in batches of m (with $m \leq k$, since repeaters can only forward at most k packets simultaneously) and each packet can be lost with probability p, the number of batches until success is denoted by the random variable $B_{n,w,p,m}$. The mean service time can be computed in terms of the mean value of $B_{n,w,p,m}$:

$$\mathbb{E}[T_{\text{service}}] = \frac{2L}{c} + t_{\text{fwd}}\left(2N + \mathbb{E}\left[B_{n,w,p,m}\right]\right), \tag{3.3}$$

  where c is the speed of light in the communication channels. The first term corresponds to the time required for a packet (or a batch of packets) to travel from one user to another. The second term accounts for the total delay introduced by the forwarding stations, which depends on the number of repeaters, N, and the expected number of batches required, $\mathbb{E}\left[B_{n,w,p,m}\right]$. In Appendix 3.5.2, we provide a derivation of (3.3), and we also explain how to compute the probability distribution of $B_{n,w,p,m}$ (in some cases, numerical approximations are required).

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

$$\rho = \frac{\lambda_0 u(u-1)m}{2k}\mathbb{E}[T_{\text{service}}]. \tag{3.4}$$

It can be shown that, if and only if $\rho > 1$, the system is overloaded and the sojourn times go to infinity (see Chapter 14 from [184]). This happens when requests are submitted at a higher rate than they are serviced. For example, if the number of users is too large, the number of requests waiting to be serviced will grow indefinitely and the sojourn times will go to infinity over time. Therefore, service will only be possible if the load remains below one.

3.2.3. QUANTUM CIRCUIT SWITCHING

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

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

1. If we reserve only one forwarding station per request, packets must be sent sequentially, i.e., $m = 1$, since each station can only forward one packet at a time (Figure 3.1a). We call this strategy *sequential distribution* of packets.

2. An alternative is to reserve multiple forwarding stations, such that multiple packets can be routed in parallel. We call this approach *parallel distribution* of packets. For simplicity, we assume that QCS with parallel distribution reserves all k stations for each request (Figure 3.1b), i.e., $m = k$. In practice, this can be done with some form of frequency multiplexing [43].

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

3.3. SEQUENTIAL VS PARALLEL DISTRIBUTION OF QUANTUM DATA PACKETS

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

1. *Small budget*. We consider the more affordable one-way quantum repeater architecture from ref. [18], which only requires two matter qubits and a single-photon emitter per forwarding station. The encoding used by this repeater is not fault-tolerant and therefore can only be used over short inter-repeater distances ($L_0 \sim 1$ km). Over such a short distance, the scheme is near-deterministic, i.e., we can assume packets are successfully delivered with probability $p \approx 1$. Since we consider a tight financial budget, we also assume that there is a single central repeater that is directly connected to the users (i.e., $N = 0$), which means that the size of the network cannot be larger than $L \sim 1$ km.

2. *Large budget*. In this case, we employ a more expensive type of forwarding station: the all-photonic proposal from ref. [139]. This architecture can encode quantum states using large distance codes, which allows for distribution of packets over longer inter-repeater distances ($L_0 > 1$ km). Since the financial budget is larger than before, we can place N repeaters in between each user and the central repeater. The probability of successful delivery depends on the number of repeaters, N, and the size of the network, L:

$$p(L,N) = 10^{-\frac{\alpha_{\text{eff}}(L_0)}{10} 2L}, \tag{3.5}$$

where $L_0 = L/(N+1)$ is the inter-repeater distance, and $\alpha_{\text{eff}}(L_0)$ is an effective attenuation coefficient that depends on the type of encoding used and on the repeater efficiency. Here, we assume $\alpha_{\text{eff}}(L_0) \approx 10^{-6}(277L_0^2 + 29L_0^4)$ dB/km, which corresponds to forwarding stations that employ the [[48, 6, 8]] generalized bicycle code [141] and have a 90% efficiency (which incorporates photon-source and detector efficiencies, on-chip loss, and coupling losses into a single parameter). This encoding and efficiency was also used as an example in ref. [139] – see Appendix 3.8 for further details. To make good use of the budget available, we choose N following the strategy from ref. [139], where the authors propose maximizing the number of repeaters per kilometer divided by the probability p, i.e., they optimize the cost function $(2N+1)/(Lp)$. For the [[48,6,8]] code and a fixed distance L, the value of N that minimizes the cost is the one that yields $p \approx 0.7$ (e.g., the optimal solutions for $L = 7.5, 13, 18, 30$ km are $N = 0, 1, 2, 5$).

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

In the following subsections, we analyze the performance of a QCS protocol in the previous use cases. In Subsection 3.3.1 we analyze the maximum number of users that the system can support before sojourn times go to infinity. Then, in 3.3.2, we compare the performance of sequential and parallel distribution, in terms of the mean sojourn time, when the budget is small (use case 1) and when it is large (use case 2). Lastly, in 3.3.3, we focus on use case 2 and study the interplay between the number of users and the distances between them, considering a fixed number of repeaters N. In Appendix 3.7, we motivate the parameter values chosen in the examples from this section.

3.3.1. CRITICAL NUMBER OF USERS

Before looking at the performance of sequential and parallel distribution of packets, the first question we ask is: *how many users can be supported by the quantum network?* That is, we want to know the maximum number of users that can be serviced before the sojourn times go to infinity – what we call the *critical number of users*, u_{crit}. We find u_{crit} by ensuring that the load of the system (3.4) is below 1, which yields

$$u_{\text{crit}} = \left\lfloor \frac{1}{2} + \frac{1}{2}\sqrt{1 + \frac{8k}{\lambda_0 m \mathbb{E}[T_{\text{service}}]}} \right\rfloor, \tag{3.6}$$

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

In the small-budget situation (use case 1), $p = 1$ and $N = 0$. Then, the mean service time from (3.3) takes a closed form:

$$\mathbb{E}[T_{\text{service}}]_{p=1,N=0} = \frac{2L}{c} + t_{\text{fwd}} \left\lceil \frac{n}{m} \right\rceil. \tag{3.7}$$

This expression can be derived using (3.3) and the fact that $\mathbb{E}[B_{n,w,1,m}] = \lceil n/m \rceil$ (for further details on the latter, see Appendix 3.5.2). From (3.6) and (3.7), one can show that, for a small budget, the critical number of users is larger when packets are distributed sequentially rather than in parallel. This can be seen in Figure 3.3, which shows u_{crit} for increasing number of forwarding stations. It is possible to provide service to a larger number of users (i.e., increase u_{crit}) by increasing the number of forwarding stations k, but only when packets are distributed sequentially. In particular, the critical number of users scales as $\sqrt{k}$. When packets are distributed in parallel, increasing the number of stations beyond $k = n$ does not allow for an increased number of users. This is a consequence of the parallel distribution model: we assume that all the forwarding stations are used simultaneously for a single request. When packet distribution is deterministic ($p = 1$), only n forwarding stations are needed to distribute n packets in parallel, and increasing k beyond this number will not provide any added benefit – those extra forwarding stations will remain unused.

When the financial budget is large (use case 2), the service time follows a nontrivial distribution and the mean value cannot be computed with (3.7) anymore. Depending on the values of p and w, we were able to compute the mean service time analytically or required Monte Carlo sampling (see Appendix 3.5 for further details). Nevertheless, the scaling of u_{crit} with k remains similar to the behavior shown in Fig. 3.3 (we provide the same plot for a large budget and different combinations of N, L, w in Appendix 3.9). We observed that, for most parameter regimes, sequential distribution can still provide service to a larger number of users than parallel distribution. However, this is not true in the parameter regime where resources are scarce, namely, when: (i) packet distribution is not deterministic ($p < 1$; specifically in our use case, $p \approx 0.7$), (ii) applications require many states to be distributed over a short period of time or users have short-lived quantum memories (small w compared to n), and (iii) there are only a few forwarding stations per repeater (small k). In a situation that meets these three conditions, many states must be successfully delivered over a short time window. Since each packet has a nonzero probability of failure, this process can take a very long time when packets are

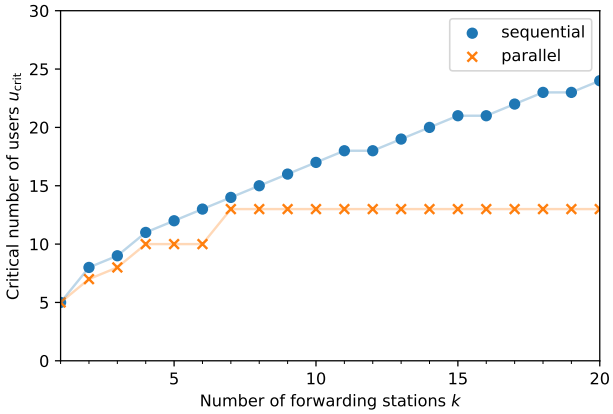

Figure 3.3: **A network with more forwarding stations k can support more users, but only when distribution of packets is sequential.** Critical number of users, u_{crit}, vs number of forwarding stations, k, for QCS with sequential (blue dots) and parallel (orange crosses) distribution of packets, in the small budget use case (see (3.6) and (3.7)). The critical number of users is the maximum number of users the system can support before the sojourn time goes to infinity. In sequential distribution, $u_{\text{crit}}(k)$ scales as $\sqrt{k}$. In parallel distribution, $u_{\text{crit}}(k) = \text{constant}, \forall k > n$. Parameters used in this figure: $N = 0$, $p = 1$, $n = 7$, $w = 10$, $\lambda_0 = 10^{-4}$ μs^{-1}, $c = 0.2$ km/μs, $t_{\text{fwd}} = 100$ μs.

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

In the large-budget scenario, the asymptotic behavior in k is similar to the small-budget case. When $p < 1$, the mean service time is lower bounded by (3.7), and therefore the critical number of users for $p = 1$ (small budget) is an upper bound for $p < 1$ (large budget). In particular, this means that u_{crit} cannot scale faster than $\sqrt{k}$ with sequential distribution, and it converges to a constant as $k \to \infty$ with parallel distribution. As mentioned earlier, we provide some graphical examples in Appendix 3.9.

3.3.2. MEAN SOJOURN TIME

Now we focus on the scenarios in which both sequential and parallel distribution can provide service within a finite mean sojourn time (MST). The MST can be computed according to (3.2), which requires calculating the mean waiting time and the mean service time in advance. In Appendix 3.5 we provide analytical formulas and numerical methods to compute them.

Figure 3.4 shows the relative difference in MST between sequential and parallel distribution, for the small-budget use case (Figure 3.4a) and the large-budget use case (Figure 3.4b). In both use cases, there is a region where both sequential and parallel distributions are possible. This region is shaded blue/red when sequential/parallel distribution is faster. When the number of users is too large, service is not possible. We indicate

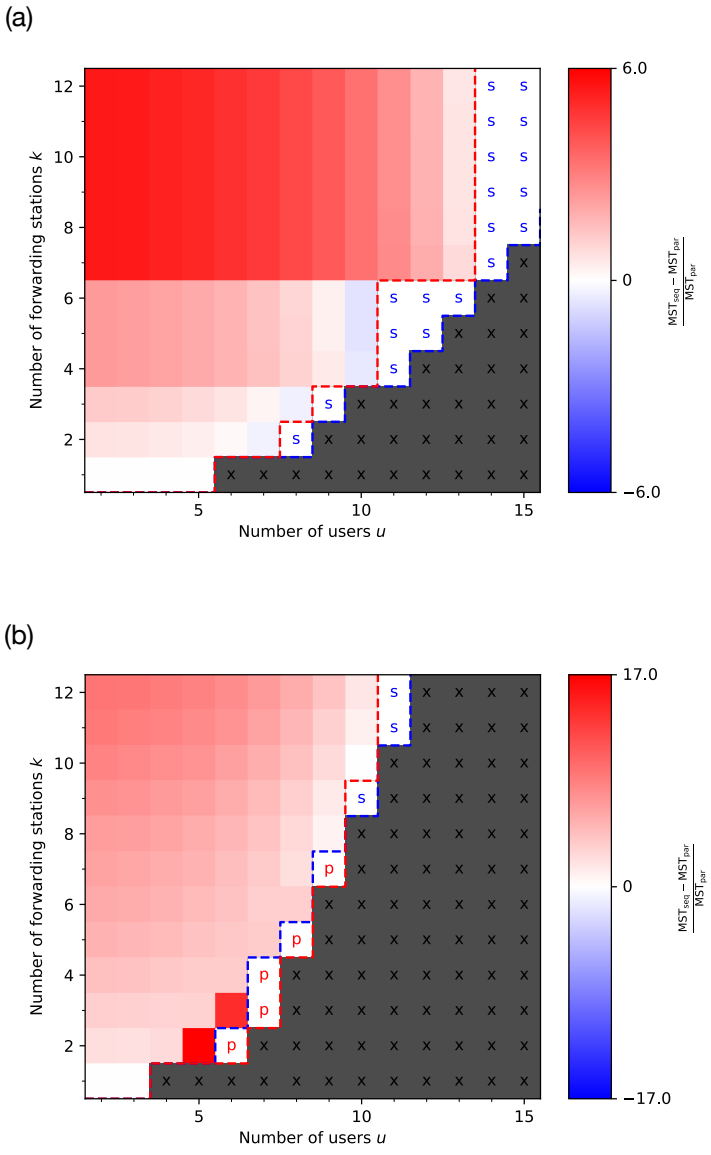

Figure 3.4: **Parallel distribution of packets is generally faster.** Relative difference in mean sojourn time (MST) between sequential and parallel packet distribution, for different numbers of users u and forwarding stations k. **(a)** Small-budget use case ($p = 1$; $w \geq n$; $N = 0$, $L = 1$ km) and **(b)** large-budget use case ($p \approx 0.7$) with $N = 0$, $L = 7.5$ km, and $w = 8$. Sequential/parallel distribution provides lower MST in blue/red regions. In regions with an 's'/'p', only sequential/parallel distribution can provide service (i.e., yield finite MST). In dark regions with an 'x', no service is possible. Parameters used in this figure: $n = 7$, $\lambda_0 = 10^{-4}$ μs^{-1}, $c = 0.2$ km/μs, $t_{\text{fwd}} = 100$ μs. MSTs in (a) calculated with (3.7). MSTs in (b) calculated with Monte Carlo sampling with 10^4 samples (the standard error in the relative difference in MST was below 0.5 for every combination of parameters).

with an 's'/'p' those regions where service can only be provided by sequential/parallel distribution. In dark regions with an 'x', service is not possible at all. We draw the following main observations from the figure:

- In the small-budget use case (Figure 3.4a), increasing the number of users (for fixed k) leads to a region where only sequential distribution can provide service. If the number of users keeps increasing, no service is possible at all. In the large-budget case (Figure 3.4b), we observe a similar behavior, except that there is also a region where only parallel distribution (instead of sequential) is possible. This is the same phenomenon that was discussed in the previous section: when $p < 1$, w is close to n, and k is small, parallel distribution supports more users. In Appendix 3.10, we show other examples in which there is a region where only parallel distribution is possible.

- For a fixed number of forwarding stations k, parallel distribution is faster than sequential distribution when the number of users is small. As we increase the number of users (for fixed k), the advantage of parallel over sequential distribution generally decreases. For some values of k, sequential distribution eventually becomes slightly better. This happens because the main feature of sequential distribution is that multiple requests can be processed simultaneously, and this is particularly beneficial when there is a large number of requests (in our model, a large number of users implies a large number of requests). In some other cases, parallel distribution is always better (e.g., in the small-budget example from Figure 3.4a with $k > 6$). When parallel distribution has a larger critical number of users (e.g., in the large-budget case from Fig 3.4b with $k = 2, 3$), the advantage in MST over sequential distribution can actually increase dramatically as the number of users increases (e.g., increasing u from 4 to 5 when $k = 2$, in Figure 3.4b). The reason is that the MST of sequential distribution diverges as we get close to a region where only parallel distribution is possible. This can be seen more clearly in Appendix 3.10, where we plot the MST vs the number of users for fixed k.

- The difference in MST converges to a constant value as $k \to \infty$. In fact, the MST should converge to a constant as $k \to \infty$ for both sequential and parallel distribution. Intuitively, once we have enough forwarding stations to meet all incoming requests, there is no benefit in increasing the number of stations per repeater. In Appendix 3.5.3, we discuss this more formally.

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

3.3.3. MANY USERS OVER LONG DISTANCES

Next, we investigate the effect of increasing the number of users and the distances between them. Here, we assume the same all-photonic forwarding stations from ref. [139] as in the large-budget use case. This means that the probability of successful packet delivery depends on the distance between users and on the number of repeaters according to (3.5).

However, as opposed to the large-budget use case, we now consider a fixed but arbitrary number of repeaters per user, N, to be placed between the user and the central repeater – that is, we do not choose N according to any cost function optimization. Each repeater has a fixed number of forwarding stations k. The question we address here is: *how far apart can the users be to provide service to them?*

Specifically, we want to analyze the critical distance L_{crit} that overloads the system, i.e., the value of L that makes the load from (3.4) equal to 1. Combining (3.3) and (3.4) and enforcing $L \geq 0$, we find that the critical distance is given by

$$L_{\text{crit}} = \max\left(0, \frac{ck}{\lambda_0 u(u-1)m} - \frac{ct_{\text{fwd}}}{2}\left(2N + \mathbb{E}\left[B_{\text{n,w,p}(L_{\text{crit}},\text{N}),\text{m}}\right]\right)\right). \tag{3.8}$$

Recall that m is the number of packets that are sent in each batch ($m = 1$ for sequential distribution and $m = k$ for parallel distribution). Note that (3.8) is a transcendental equation, since p is a function of L_{crit}, and it must be solved numerically.

The first observation from (3.8) is that L_{crit} decreases (and eventually drops to zero) for increasing u, regardless of the number of repeaters N. The second term is always negative, so the critical distance can be upper bounded as

$$L_{\text{crit}} \leq \frac{ck}{\lambda_0 u(u-1)m}. \tag{3.9}$$

The scaling with the number of users is therefore upper bounded by $\sim u^{-2}$. This unveils a tradeoff between the number of users and the distances between them. When service is provided to a large number of users, the number of incoming requests will increase and this will put a higher load on the system. If the distances between users are large, service times will increase, since packets will need to travel further away, which will also increase the load. Consequently, we must decrease u to increase L and vice versa. Figure 3.5 shows an example where we can observe this effect.

We have shown that L and u cannot be scaled up simultaneously when N is fixed, but *what if we add more intermediate repeaters to boost the probability of successful packet delivery? Would this allow us to place many users further apart?* Intuitively, one may think that this is the case, since a larger number of repeaters N provides a larger success probability p (see (3.5)). Increasing p means that we will need to send less (batches of) packets per request, i.e., $\mathbb{E}[B_{n,w,p,m}]$ will be smaller, and consequently the service time (3.3) should decrease and the critical distance (3.8) should increase. However, this intuition is not always correct, since each additional repeater introduces an additional delay t_{fwd}, which directly impacts the service time and the critical distance: both (3.3) and (3.8) include a linear term in N. These linear terms yield an interesting behavior: the delays introduced by the repeaters accumulate and the service time eventually increases (and the critical distance decreases) when increasing the number of repeaters N, despite the benefit from a larger p. If there are too many repeaters, these forwarding delays may overload the system, preventing requests from being met within a finite amount of time.

When the number of users is small, increasing the number of repeaters is actually beneficial: it allows us to increase the distances between users (i.e., larger N provides larger L_{crit}), as can be seen in Figure 3.5. When the number of users is large, service is only possible if the number of repeaters is small. That is, when increasing the number of users,

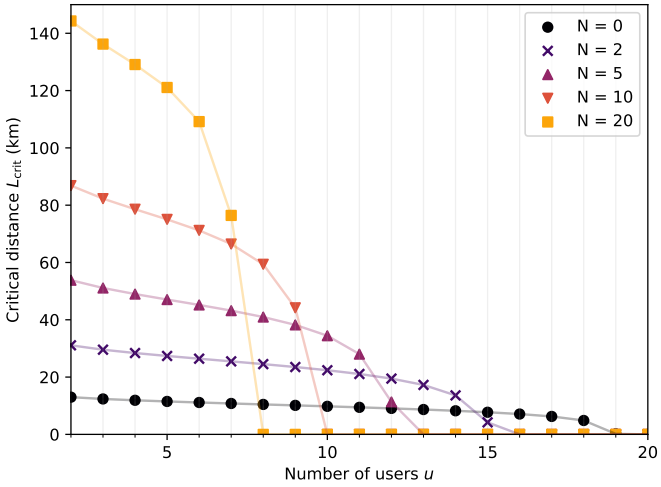

Figure 3.5: **Networks with many users cannot cover long distances.** Critical distance L_{crit} vs number of users u, for different numbers of repeaters N. Parameters used in this figure: sequential distribution, $n = 7$, $w \to \infty$, $k = 12$, $\lambda_0 = 10^{-4}\ \mu\text{s}^{-1}$, $c = 0.2$ km/μs, $t_{\text{fwd}} = 100\ \mu$s.

L_{crit} drops to zero quicker for larger N. As a consequence, for intermediate values of u, there is an finite value of N that maximizes the critical distance. For instance, when there are ten users in the example from Figure 3.5, using $N = 5$ repeaters allows the users to be further apart than if we used $N = 10$ repeaters. Note that Figure 3.5 assumes sequential distribution of packets, although the previous discussion remains general and also applies to parallel distribution, as shown in Appendix 3.11.

From this analysis we draw two main conclusions:

1. There exists a tradeoff in the implementation of a QCS protocol: the number of users and the distances between them cannot be scaled up simultaneously. This tradeoff happens for any number of repeaters N and any function $p(L, N)$ (see (3.9)), which means that the tradeoff will exist even with unlimited resources and ideal hardware.

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

3.4. OUTLOOK

This chapter lays the groundwork for further exploration and refinement of quantum circuit switching (QCS) protocols in the context of one-way quantum networks. We have explored two fundamental resource-allocation strategies: in the first one, quantum data packets are distributed sequentially, and, in the second one, they are distributed in batches. We concluded that sequential distribution can generally provide service to

a larger number of users, although parallel distribution generally meets requests at a higher rate. We also found a tradeoff between the number of users supported by the QCS protocol and the size of the network.

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

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

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

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

3.5. [APPENDIX] - ANALYTICAL CALCULATION OF THE MEAN SOJOURN TIME

In this Appendix, we show how to calculate the mean waiting time and the mean service time of a QCS protocol in a star network (the mean sojourn time can be obtained by adding both contributions). For that, we model the system as an M/G/s queue. In this queue model, new requests arrive following a Poisson distribution, i.e., request arrivals are Markovian (M). Incoming requests are placed in a common queue. Requests in the queue are processed according to a first-in-first-out policy. Processing a request takes some service time, which follows a general distribution (G), and a maximum of s requests can be processed simultaneously. When quantum data packets are distributed sequentially ($m = 1$), each of the k forwarding stations is dedicated to meet one request and therefore $s = k$ (see Figure 3.6a). When packets are distributed in parallel ($m = k$), all forwarding stations are reserved to meet one request at a time and therefore $s = 1$ (see Figure 3.6b).

In Sections 3.5.1 and 3.5.2, we show how to compute waiting and service times, respectively, in the M/G/s model for QCS. These results are summarized in Tables 3.2 and 3.3. We conclude this Appendix discussing the limit when the number of forwarding stations goes to infinity (Section 3.5.3).

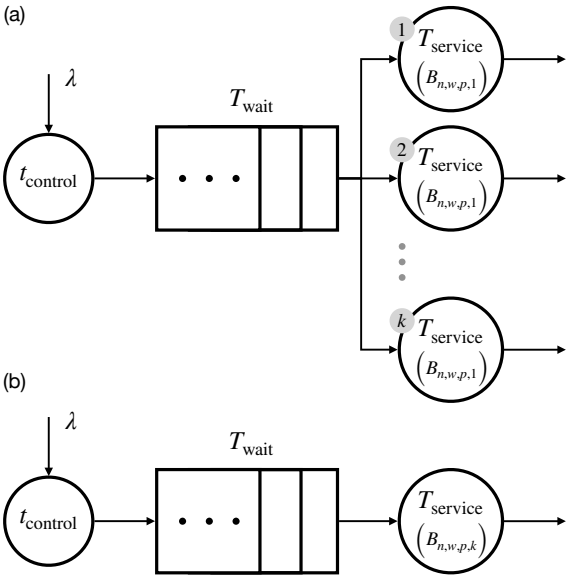

Figure 3.6: **QCS in a star network can be modeled as an M/G/s queue.** Illustration of the queue model of a QCS scheme with (a) sequential distribution of packets ($s = k$) and (b) parallel distribution of packets ($s = 1$), in a star network with k forwarding stations per repeater. New requests are submitted at rate λ. After some control time t_{control}, the request is placed in a first-in first-out queue. After some waiting time T_{wait}, the request leaves the queue and begins being processed. Processing the request takes time T_{service}. Sequential distribution (a) can process up to k requests in parallel, while parallel distribution (b) can only process one at a time. The service time distribution is given by the distribution of the number of batches of packets that must be sent until the request is met, $B_{n,w,p,m}$ (see Appendix 3.5.2).

3.5.1. WAITING TIME

Calculating the waiting time poses a major problem: no general analytical solution is known for the waiting time of an M/G/s queue. Only for $s = 1$, a closed-form solution is known (see, e.g., Chapter 14 from [184]):

$$\mathbb{E}\left[T_{\text{wait}}^{\text{M/G/1}}\right] = \frac{\lambda \mathbb{E}\left[\left(T_{\text{service}}^{\text{M/G/1}}\right)^2\right]}{2\left(1-\lambda \mathbb{E}\left[T_{\text{service}}^{\text{M/G/1}}\right]\right)}, \tag{3.10}$$

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

Multiple approximations and bounds for the waiting time distribution have been found for $s > 1$ (see, e.g., [77]). One of the most well-known approximations is the early formula from ref. [113]:

$$\mathbb{E}\left[T_{\text{wait}}^{\text{M/G/}s}\right] \approx \frac{C_{\text{service}}^2 + 1}{2} \mathbb{E}\left[T_{\text{wait}}^{\text{M/M/}s}\right], \tag{3.11}$$

Table 3.2: **How to compute the mean waiting time in QCS in star networks**, depending on the type of packet distribution (sequential or parallel) and the number of forwarding stations per repeater (k). We assume that the first two moments of the service time distribution are known. See Appendix 3.5.1 for further details.

	Sequential ($s = k$)	Parallel ($s = 1$)
$k = 1$:	Closed-form - (3.10)	Closed-form - (3.10)
$k > 1$:	Approximation - (3.11)	Closed-form - (3.10)

where $C^2_{\text{service}} \equiv \text{Var}\left[T^{\text{M/G/}s}_{\text{service}}\right] / \mathbb{E}\left[T^{\text{M/G/}s}_{\text{service}}\right]^2$ is the squared coefficient of variation of the service time distribution, and M/M/s is an auxiliary queuing system with a single queue, Markovian arrivals (M), exponentially distributed service times (M) with mean $\mu \equiv \mathbb{E}\left[T^{\text{M/G/}s}_{\text{service}}\right]$, and capacity to process s requests simultaneously. The mean waiting time of the auxiliary queue can be computed as follows (see Chapter 14 from [184]):

$$\mathbb{E}\left[T^{\text{M/M/}s}_{\text{wait}}\right] = \frac{\lambda^s}{s!s\mu^{s+1}z^2}\left(\frac{\lambda^s}{s!\mu^s z} + \sum_{i=0}^{s-1}\frac{\lambda^i}{i!\mu^i}\right)^{-1}, \tag{3.12}$$

with $z \equiv 1-\lambda/(s\mu)$. Refs. [77] and [198] provide empirical evidence that the approximation (3.11) works well when the service time distribution has a small squared coefficient of variation C^2_{service}. We observed empirically that, in the use cases studied in this chapter, C^2_{service} is indeed small (see Appendix 3.6) and therefore we expect this approximation to be accurate. Note also that the approximation assumes the first two moments of the service time distribution are known.

To study our QCS protocol, we can employ the closed-form solution (3.10) when (i) packets are distributed sequentially and there is a single forwarding station per repeater ($k = 1$) and (ii) when packets are distributed in parallel. When packets are distributed sequentially with $k > 1$, we can only use approximations, such as (3.11), or Monte Carlo sampling. In practice, the use of both the closed-form solution and the approximation are restricted to situations in which we can efficiently compute the first two moments of the service time. We discuss this in the next section.

3.5.2. SERVICE TIME

As introduced in the main text, $B_{n,w,p,m}$ is the number of batches of quantum data packets that must be sent from user to user until n packets are successfully distributed within a time window w, assuming each packet has a success probability p and each batch contains m packets. The service time can be computed as

$$T_{\text{service}} = \frac{2L}{c} + t_{\text{fwd}}(2N+1) + t_{\text{fwd}}\left(B_{n,w,p,m} - 1\right), \tag{3.13}$$

where L is the distance between each user and the central repeater, c is the speed of light in the physical channels, t_{fwd} is the forwarding time, and N is the number of repeaters

Table 3.3: **How to compute the first two moments of $B_{n,w,p,m}$.** Once these moments are known, the first two moments of the service time can be trivially computed with (3.14) and (3.15). The method used to compute $B_{n,w,p,m}$ depends on the type of packet distribution (sequential or parallel), the window size (w), and the probability of successful packet delivery (p). See Appendix 3.5.2 for further details.

	Sequential ($m = 1$)	Parallel ($m = k$)
$p = 1$:	Closed-form - (3.17) and (3.18)	Closed-form - (3.21) and (3.22)
$w \to \infty$:	Closed-form - (3.19) and (3.20)	Analytical - (3.31), (3.32), and (3.33)
$p < 1$ and $w < \infty$:	Analytical but expensive to evaluate - [49]	Monte Carlo sampling

between each user and the central repeater. The first term corresponds to the time required for the first packet (or batch of packets) to travel from one user to another, without considering forwarding delays. The second term corresponds to the delay introduced by the forwarding stations: each station takes time t_{fwd} to forward a packet, and the packet (or batch of packets) is forwarded at $2N+1$ intermediate repeaters. Therefore, the first two terms account for the total time since the first packet (or batch of packets) is sent from one user until it is received by the other user. The third term accounts for the time it takes to receive the remaining $B_{n,w,p,m} - 1$ (batches of) packets: a packet (or batch of packets) is sent every t_{fwd} units of time, to ensure it will arrive at the next repeater when the previous packet is already processed and leaving the repeater. The mean service time is then given by

$$\mathbb{E}[T_{\text{service}}] = \frac{2L}{c} + t_{\text{fwd}}(2N + \mathbb{E}\left[B_{n,w,p,m}\right]), \tag{3.14}$$

since $B_{n,w,p,m}$ is the only random variable involved. As shown in Section 3.5.1, the second moment of the service time is also required to compute the expected waiting time $\mathbb{E}[T_{\text{wait}}]$. The second moment can be computed as

$$\begin{aligned}\mathbb{E}\left[T^2_{\text{service}}\right] = &\left(\frac{4L^2}{c^2} + \frac{8L}{c}\, t_{\text{fwd}} N + 4 t^2_{\text{fwd}} N^2\right) \\ &+ \left(\frac{4L}{c}\, t_{\text{fwd}} + 4 t^2_{\text{fwd}} N\right) \mathbb{E}\left[B_{n,w,p,m}\right] + t^2_{\text{fwd}} \mathbb{E}\left[B^2_{n,w,p,m}\right].\end{aligned} \tag{3.15}$$

From (3.14) and (3.15), we conclude that the service and the waiting times can be efficiently computed if we can efficiently compute the first two moments of $B_{n,w,p,m}$. Next, we discuss how to compute them.

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

$$B_{n,w,p,m} = \inf\left\{ x : \sum_{j=x-w+1}^{x} S_j \geq n \right\}. \tag{3.16}$$

It is possible to calculate the probability distribution of $B_{n,w,p,m}$ analytically when there is no multiplexing ($m = 1$) [49]. When $m = 1$ and $p = 1$, the solution is trivial since we deterministically need n batches to successfully deliver n packets (there is only one per batch). In that case, we have

$$\mathbb{E}[B_{n,w,1,1}] = n, \tag{3.17}$$

$$\mathbb{E}\left[\left(B_{n,w,1,1}\right)^2\right] = n^2. \tag{3.18}$$

As shown in ref. [49], in the case of $m = 1$ and an infinite window ($w \to \infty$), $B_{n,w,p,1}$ follows a negative binomial distribution, which has the following first two moments:

$$\mathbb{E}[B_{n,\infty,p,1}] = \frac{n}{p}, \tag{3.19}$$

$$\mathbb{E}\left[\left(B_{n,\infty,p,1}\right)^2\right] = \frac{n(1-p)+n^2}{p^2}. \tag{3.20}$$

For every other combination of values of n, p, and w, and with $m = 1$, there exists a nontrivial analytical solution, as shown in ref. [49]. However, computing the solution involves inverting a matrix whose size scales as $\mathcal{O}(w^{n-1})$. Hence, in practice we need to employ approximate methods (such as Monte Carlo sampling) for arbitrary values of n, w, and p.

In the multiplexed case ($m > 1$), there is no known analytical solution yet for every combination of n, w, p, and m, to the best of our knowledge. Next, we solve the problem for two cases: (i) $p = 1$ and (ii) $w \to \infty$. In every other case, we employed Monte Carlo sampling to estimate the probability distribution of $B_{n,w,p,m}$ when needed. First, the case $p = 1$ is again trivial. When packets are always successfully delivered, we need $\lceil n/m \rceil$ batches to deliver n packets. Hence,

$$\mathbb{E}[B_{n,w,1,m}] = \left\lceil \frac{n}{m} \right\rceil, \tag{3.21}$$

$$\mathbb{E}\left[\left(B_{n,w,1,m}\right)^2\right] = \left\lceil \frac{n}{m} \right\rceil^2. \tag{3.22}$$

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

$$\Pr\left[B_{n,\infty,p,m} > b\right] \overset{a}{=} \Pr\left[\sum_{i=1}^{b} S_i < n\right] \tag{3.23}$$

$$\overset{b}{=} \sum_{s_b=0}^{m} \Pr\left[\sum_{i=1}^{b-1} S_i < n - s_b \mid S_b = s_b\right] \Pr\left[S_b = s_b\right] \tag{3.24}$$

$$\overset{c}{=} \sum_{s_b=0}^{m} \Pr\left[\sum_{i=1}^{b-1} S_i < n - s_b\right] \Pr\left[S_b = s_b\right] \tag{3.25}$$

$$\overset{d}{=} \sum_{s_1,\ldots,s_b=0}^{m} \Pr\left[\sum_{i=1}^{b} s_i < n\right] \prod_{i=1}^{b} \Pr\left[S_i = s_i\right] \tag{3.26}$$

$$\overset{e}{=} \sum_{s_1,\ldots,s_b=0}^{m} \Pr\left[\sum_{i=1}^{b} s_i < n\right] \prod_{i=1}^{b} \binom{m}{s_i} p^{s_i} (1-p)^{m-s_i} \tag{3.27}$$

$$= \sum_{s_1,\ldots,s_b=0}^{m} \Pr\left[\sum_{i=1}^{b} s_i < n\right] p^{\sum_{i=1}^{b} s_i} (1-p)^{mb-\sum_{i=1}^{b} s_i} \prod_{i=1}^{b} \binom{m}{s_i} \tag{3.28}$$

$$= (1-p)^{mb} \sum_{s_1,\ldots,s_b=0}^{m} \Pr\left[\sum_{i=1}^{b} s_i < n\right] \left(\frac{p}{1-p}\right)^{\sum_{i=1}^{b} s_i} \prod_{i=1}^{b} \binom{m}{s_i} \tag{3.29}$$

$$\overset{f}{=} (1-p)^{mb} \sum_{s_1=0}^{z_1} \sum_{s_2=0}^{z_2} \cdots \sum_{s_b=0}^{z_b} \left(\frac{p}{1-p}\right)^{\sum_{i=1}^{b} s_i} \prod_{i=1}^{b} \binom{m}{s_i}. \tag{3.30}$$

$$\tag{3.31}$$

with the following steps:

a. We define S_i as the number of successes in batch i. Completing the process in more than b batches ($B_{n,\infty,p,m} > b$) corresponds to obtaining less than n successful attempts in the first b batches ($\sum_{i=1}^{b} S_i < n$).

b. We use the law of total probability.

c. The number of successes in each batch are independent (i.e., S_i are i.i.d.).

d. We apply the previous two steps recursively for every S_i.

e. S_i follows a binomial distribution with m attempts and probability of success p.

f. We define $z_i \equiv \min\left(m, n-1-\sum_{j=1}^{i-1} s_j\right)$.

The survival function (3.31) can then be used to calculate the first two moments of $B_{n,\infty,p,m}$ as follows:

$$\mathbb{E}[B_{n,\infty,p,m}] = \sum_{b=0}^{\infty} \Pr\left[B_{n,\infty,p,m} > b\right], \tag{3.32}$$

$$\mathbb{E}[B_{n,\infty,p,m}^2] = \sum_{b=1}^{\infty} b^2 \left(\Pr[B_{n,\infty,p,m} > b-1] - \Pr[B_{n,\infty,p,m} > b]\right), \tag{3.33}$$

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

3.5.3. LIMITS WHEN $k \to \infty$

To conclude this Appendix, we show that the mean sojourn time converges to a finite value when the number of forwarding stations, k, goes to infinity. For that, we only need to show that the mean waiting time and the mean service time also converge to a constant.

Let us start with the service time. Both $\mathbb{E}[T_{\text{service}}]$ and $\mathbb{E}[T^2_{\text{service}}]$ are linear in $\mathbb{E}[B_{n,w,p,m}]$ and $\mathbb{E}[B^2_{n,w,p,m}]$ (see (3.14) and (3.15)). Next, we prove that these expected values go to a constant when $k \to \infty$:

- When packets are distributed sequentially, $m = 1$, and both $B_{n,w,p,m}$ and T_{service} are independent of k. Hence, the first two moments are constant in k.

- When packets are distributed in parallel, $m = k$. When $k \to \infty$, the number of packets in each batch of the window problem goes to infinity, and therefore the probability of successfully distributing a fixed number of packets n in a single batch goes to 1 (assuming $p > 0$, otherwise success is never declared). More specifically, the probability that the number of successes in the first batch, S_1, is larger than or equal to n goes to 1. This can be proven by showing that this probability is lower bounded by a value that converges to 1:

$$\begin{aligned}
\lim_{m\to\infty} \Pr[S_1 \geq n] &= 1 - \lim_{m\to\infty} \Pr[S_1 < n] \\
&= 1 - \lim_{m\to\infty} \sum_{x=0}^{n-1} \Pr[S_1 = x] \\
&= 1 - \lim_{m\to\infty} \sum_{x=0}^{n-1} \binom{m}{x} p^x (1-p)^{m-x} \\
&= 1 - \lim_{m\to\infty} \sum_{x=0}^{n-1} \frac{1}{x!} \left(\frac{p}{1-p}\right)^x \frac{m!}{(m-x)!} (1-p)^m \\
&> 1 - \lim_{m\to\infty} \sum_{x=0}^{n-1} \frac{1}{x!} \left(\frac{p}{1-p}\right)^x m^x (1-p)^m \\
&= 1,
\end{aligned}$$

  where we have used the following: (i) S_i follows a binomial distribution with m attempts and probability of success p, (ii) $m!/(m-x)! < m^x$, and (iii) $\lim_{m\to\infty} m^x q^m = 0$, for $x \geq 0$ and $0 < q \leq 1$. Combining the previous result with (3.16), we obtain $\mathbb{E}[B_{n,w,p,m}] \to 1$ and $\mathbb{E}[B^2_{n,w,p,m}] \to 1$ when $k \to \infty$. Since $\mathbb{E}[T_{\text{service}}]$ and $\mathbb{E}[T^2_{\text{service}}]$ are linear in $\mathbb{E}[B_{n,w,p,m}]$ and $\mathbb{E}[B^2_{n,w,p,m}]$ (see (3.14) and (3.15)), they also converge to constant values.

The waiting time also converges to a constant as $k \to \infty$:

- When packets are distributed sequentially, an infinite number of forwarding stations means that we can simultaneously process as many requests as desired. Intuitively, this means that every incoming request will be immediately processed and the waiting time will be zero. We leave the formal proof as future work.

- When packets are distributed in parallel, only one request is processed at a time. The mean waiting time can be computed using (3.10), which only depends on the first two moments of the service time. Since we have shown that these converge to constant values as $k \to \infty$, it is trivial to show that (3.10) also converges to a constant (assuming $\lambda \mathbb{E}[T_{\text{service}}^{\text{M/G/1}}] < 1$; otherwise, the system is overloaded and the waiting times go to infinity, as discussed in the main text).

3.6. [APPENDIX] - SQUARED COEFFICIENT OF VARIATION OF THE SERVICE TIME

As discussed in Appendix 3.5.1, there is no known general solution for the expected waiting time of an M/G/s queue, with $s > 1$. Approximations such as (3.11) work well when the squared coefficient of variation of the service time, C^2_{service}, takes small values – in [77], the authors consider large values of C^2_{service} to be in the order of $\gtrsim 10 - 100$. In this Appendix, we empirically show that C^2_{service} takes small values in systems in which the service time is linear in the number of batches until success of a window problem (this is the case for our QCS protocols).

First, we write the service time as

$$T_{\text{service}} = x + yB_{n,w,p,m}, \tag{3.34}$$

where x and y are non-negative constants and $B_{n,w,p,m}$ is the number of batches until success, as defined in (3.16). In our work, $x = 2L/c + 2t_{\text{fwd}}N$, which accounts for the travel time of a quantum data packet from user to user and for the processing delays introduced by the repeaters, and $y = t_{\text{fwd}}$, which accounts for the delays in between (batches of) packets. The first and second moments of the service time are

$$\mathbb{E}[T_{\text{service}}] = x + y\mathbb{E}\left[B_{n,w,p,m}\right], \tag{3.35}$$

$$\mathbb{E}\left[T^2_{\text{service}}\right] = x^2 + 2xy\mathbb{E}\left[B_{n,w,p,m}\right] + y^2\mathbb{E}\left[B^2_{n,w,p,m}\right]. \tag{3.36}$$

For a service time of the form (3.34), the squared coefficient of variation is upper bounded by the squared coefficient of variation of $B_{n,w,p,m}$:

$$\begin{aligned}
C^2_{\text{service}} &= \frac{\text{Var}[T_{\text{service}}]}{\mathbb{E}[T_{\text{service}}]^2} \\
&= \frac{\mathbb{E}\left[T^2_{\text{service}}\right]}{\mathbb{E}[T_{\text{service}}]^2} - 1 \\
&= \frac{x^2 + 2xy\mathbb{E}\left[B_{n,w,p,m}\right] + y^2\mathbb{E}\left[B^2_{n,w,p,m}\right]}{\left(x + y\mathbb{E}\left[B_{n,w,p,m}\right]\right)^2} - 1 \\
&= \frac{x^2 + 2xy\mathbb{E}\left[B_{n,w,p,m}\right] + y^2\mathbb{E}\left[B_{n,w,p,m}\right]^2\left(1 + C^2_{n,w,p,m}\right)}{\left(x + y\mathbb{E}\left[B_{n,w,p,m}\right]\right)^2} - 1
\end{aligned}$$

$$\leq \left(1 + C^2_{n,w,p,m}\right) \frac{x^2 + 2xy\mathbb{E}\left[B_{n,w,p,m}\right] + y^2\mathbb{E}\left[B_{n,w,p,m}\right]^2}{\left(x + y\mathbb{E}\left[B_{n,w,p,m}\right]\right)^2} - 1$$
$$= C^2_{n,w,p,m}, \tag{3.37}$$

where we have used the fact that $C^2_{n,w,p,m} \equiv \mathrm{Var}\left[B_{n,w,p,m}\right] / \mathbb{E}\left[B_{n,w,p,m}\right]^2 \geq 0$.

As a consequence of the bound (3.37), showing that $B_{n,w,p,m}$ has a small coefficient of variation is enough to show that the service time also has a small coefficient of variation. Figure 3.7 shows $C^2_{n,w,p,m}$ for some parameter regimes of n, w, p, and m that are relevant in our work. This figure provides empirical evidence that $C^2_{n,w,p,m}$ is below 1 in the parameter regimes explored in this work. Consequently, according to [77], the mean waiting time should be well approximated by (3.11). Note that there are other interesting features in Figure 3.7 (e.g., $C^2_{n,w,p,m}$ is non-monotonic in n and m), although we only focus on the order of magnitude of $C^2_{n,w,p,m}$ and therefore a detailed study of the behavior of $C^2_{n,w,p,m}$ is out of the scope of this work.

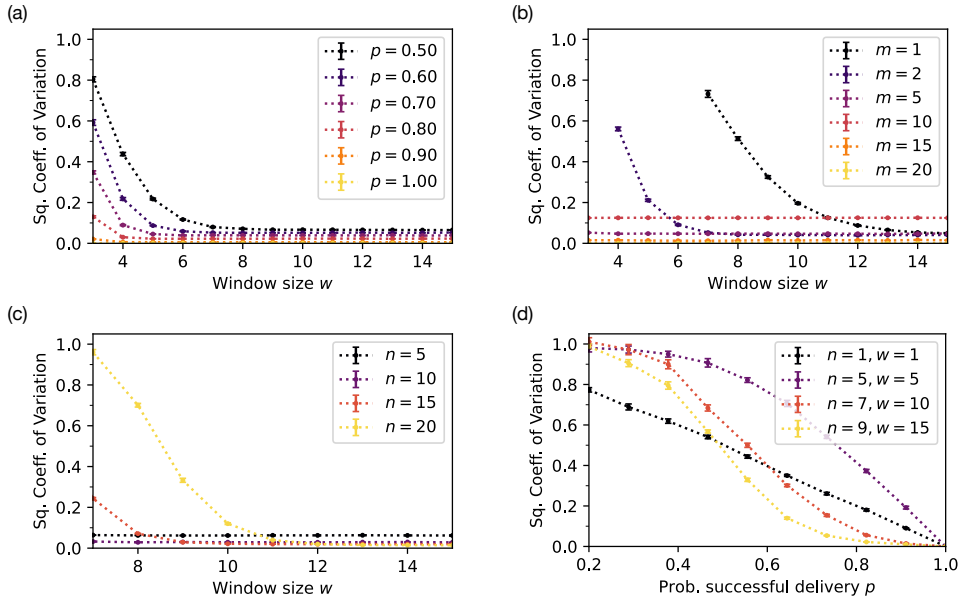

Figure 3.7: **The squared coefficient of variation of the window problem is below 1 for the relevant parameter regimes.** Squared coefficient of variation of $B_{n,w,p,m}$, $C^2_{n,w,p,m} \equiv \mathrm{Var}\left[B_{n,w,p,m}\right] / \mathbb{E}\left[B_{n,w,p,m}\right]^2$. In the parameter regions explored, $C^2_{n,w,p,m}$ decreases with increasing window size w and with increasing probability of success p. The behavior with n and m is nontrivial although the values remain below 1 (in the parameter regions explored). Each data point is the average over 10 simulations (error bars correspond to the standard error). Each simulation consisted in the estimation of $C^2_{n,w,p,m}$ over 10^3 samples of $B_{n,w,p,m}$ collected via Monte Carlo sampling. Parameters used in each subfigure: (a) $n = 7$, $m = 3$; (b) $n = 7$, $p = 0.7$; (c) $p = 0.7$, $m = 3$; (d) $m = 1$.

3.7. [APPENDIX] - PARAMETER VALUES

In Table 3.4, we provide a summary of all the parameters involved in the model, including the specific values that we use in our examples. Next, we motivate the choice of these numerical values for our examples.

For the number of users u, the number of repeaters N, and the number of forwarding stations per repeater k, we employ values that are reasonable for early quantum networks, on the order of magnitude of $1-10$. Additionally, the specific values we choose provide illustrative examples in which the system exhibits interesting and/or useful behavior (e.g., if we used the same parameter values but the number of users was much larger than 20, the mean sojourn times would diverge to infinity and no service would be possible). The physical size of the network, L, is in the order of magnitude of $1-10$ km. A short distance (1 km) is required for the forwarding stations based on the proposal from ref. [18], while all-photonic repeaters are expected to work at internode distances of $1-10$ km [136, 139].

The forwarding time of each station t_{fwd} is 100 μs in all our examples. We obtain this value if we consider that: (i) approximately ten gates (order of magnitude) are needed to forward a quantum data packet (e.g., the encoding and decoding circuits of the five-qubit perfect code consist of at most 8 gates each [112]), and (ii) a quantum gate takes around 1-50 μs (this is the case for qubits realized with color centers, such as nitrogen vacancies – see, e.g., refs. [100, 147]). The probability of success p depends on L, on N, and also on the hardware used to implement the forwarding stations, as discussed in the main text. We assume the physical channels between users are optical fibers in which the speed of light is approximately 0.2 km/μs [110]. As discussed in the main text, we consider negligible control time since this would not affect our analysis.

Regarding the requests, increasing the number of requested states, n, would increase the expected service time. We tested different values of n and did not find any different behavior from the system. In our examples, we use $n = 7$. We chose this relatively small value of n because a large n would increase the runtime of the Monte Carlo sampling when the analytical solutions cannot be applied. The request window w must be as large as n (otherwise sequential distribution is not possible and we can only distribute packets in parallel). We consider values of w that are close to n (7, 8, and 10) and also $w \to \infty$. We do not consider large values of w since the results converge quickly to the infinite window case due to large values of p being used (in most of our examples, $p \geq 0.7$). Lastly, the value of $\lambda_0 = 10^{-4}$ μs^{-1} used in our examples was chosen to illustrate interesting behavior, for the other parameter values chosen. If λ_0 is small, the system can process requests seamlessly and saturates at a larger number of users (i.e., the critical number of users increases). If λ_0 is large, the opposite happens. Note that, when $\lambda_0 = 10^{-4}$ μs^{-1}, each pair of users submits a request every 10^4 μs (on average): during this time, a single forwarding station can forward 100 quantum data packets.

Table 3.4: **Parameters of a quantum network running a QCS protocol.** In our analysis, we consider a star topology where user nodes are connected via a central repeater, although our definitions and methods remain general. The second column provides the parameter values used in our examples.

	Physical topology	
u	$2-20$	Number of users
L	$1-30$ km	Distance between each user and the central repeater
N	$0-5$	Number of repeaters between each user and the central repeater
	Hardware	
k	$1-15$	Number of forwarding stations per repeater
t_{fwd}	100 μs	Forwarding time per repeater and quantum data packet
p	-	Probability of successful packet delivery from user to user
c	0.2 km/μs	Speed of light in the physical channels
t_{control}	0	Control time
	Requests	
n	7	Number of entangled pairs per request
w	≥ 7	Request time window
λ_0	10^{-4} μs^{-1}	Request submission rate per pair of users

3.8. [APPENDIX] - ATTENUATION IN ALL-PHOTONIC QUANTUM REPEATERS

In this work, the probability of successfully delivering a quantum data packet, p, is assumed to depend on the physical implementation of the forwarding stations, the distance between repeaters (L_0), and the number of intermediate repeaters between users. In some of the use cases discussed in the main text, we consider the all-photonic forwarding stations proposed in ref. [139]. In this Appendix, we provide more details about how this choice of hardware determines the dependence of p on L_0. In particular, we explain how we calculate the effective attenuation coefficient $\alpha_{\text{eff}}(L_0)$ from (3.5).

In our examples, we consider repeaters that employ the [[48, 6, 8]] generalized bicycle code [141], which was also used as an example in ref. [139]. Moreover, we include photon-source and detector efficiencies, on-chip loss, and coupling losses into a single parameter: the forwarding station efficiency (or transmittance) η_r. We assume $\eta_r = 0.9$. This value was also used as an example in ref. [139]. In forwarding stations with efficiency $\eta_r = 0.9$ that use the [[48,6,8]] code, the effective attenuation coefficient can be approximated by

$$\alpha_{\text{eff}}(L_0) \approx 10^{-6}(277L_0^2 + 29L_0^4)\text{dB/km}. \tag{3.38}$$

We obtained this expression by fitting a fourth order polynomial to the data provided in ref. [139] (see Fig. 3.8).

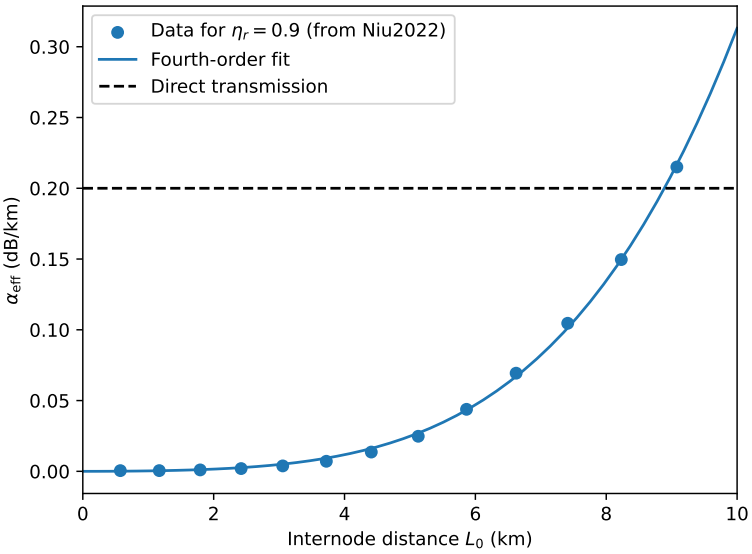

Figure 3.8: **Effective attenuation coefficient vs internode distance.** We fit α_{eff} versus L_0 for $\eta_r = 0.9$ and the [[48, 6, 8]] code to the data from Fig 4b from ref. [139] (blue dots, raw data extracted with WebPlotDigitizer [158]). Fourth order fitting with Plotly: $\alpha_{\text{eff}} \approx 10^{-6}(277L_0^2 + 29L_0^4)$ dB/km (solid line). Direct transmission over optical fiber experiences an attenuation of 0.2 dB/km (dashed line).

3.9. [APPENDIX] - CRITICAL NUMBER OF USERS WITH PROBABILISTIC PACKET DELIVERY

In this Appendix, we provide more examples of the behavior of the critical number of users, u_{crit}, with increasing number of forwarding stations, k, in the large-budget use case from Section 3.3. Figure 3.9 shows u_{crit} vs k for $w = 7$ and $w \to \infty$, for $N = 0$ and $N = 5$ (corresponding to $L = 7.5$ and $L = 30$ km, respectively). In these cases, the probability of successful packet delivery is $p \approx 0.7$. As in the small-budget scenario, u_{crit} increases with increasing k for sequential distribution of packets. Conversely, u_{crit} reaches a maximum value when packets are distributed in parallel, i.e., we cannot increase u_{crit} indefinitely by increasing k, for parallel distribution.

As discussed in the main text, parallel distribution supports more users than sequential distribution only when the window size is small (i.e., close to the number of states requested, n) and when there are few forwarding stations per repeater (small k). In Figure 3.9, we have $n = 7$, and we observe this behavior when $w = 7$ but not for $w \to \infty$. We also observe that, in the parameter regimes explored, this behavior vanishes when increasing the size of the network: for $L = 7.5$ km, parallel distribution can support more

users than sequential distribution for $w = 7$ and small k; but for $L = 30$ km, parallel distribution cannot support more users for any values of w and k. For more examples, see the Appendix of our paper [89].

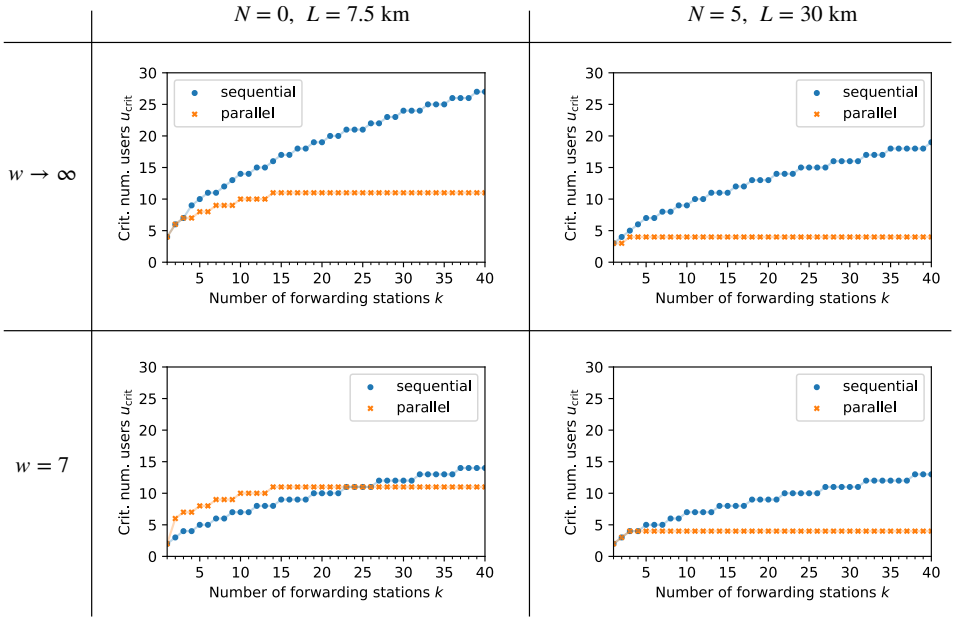

Figure 3.9: **Scaling of critical number of users with the number of forwarding stations, for probabilistic quantum data packet delivery.** Critical number of users, u_{crit}, vs number of forwarding stations, k, for QCS with sequential (blue dots) and parallel (orange crosses) distribution of packets. Star network with $N = 0$ and $N = 5$ (left to right) repeaters between each user and the central repeater, and distance $L = 7.5$ and $L = 30$ km (left to right) between each user and the central repeater. These combinations of N and L are cost efficient (they minimize $N/(Lp)$) and yield a probability of successful packet delivery of $p \approx 0.7$. Nodes request $n = 7$ quantum data packets to be successfully delivered within a time window $w \to \infty$ and $w = 7$ (top to bottom). When the window size is close to n and only a few forwarding stations per node are available (e.g., bottom left subplot, small k), parallel distribution supports more users – otherwise, sequential distribution supports more users. Other parameters used in this figure: $\lambda_0 = 10\ \mu\text{s}^{-4}$, $t_{\text{fwd}} = 100\ \mu$s. Results calculated using (3.6) and (3.3). For $w \to \infty$, $\mathbb{E}[B_{n,w,p,m}]$ (required to compute (3.3)) was computed using the analytical solution from Appendix 3.5.2 – in the other cases, the probability distribution of $B_{n,w,p,m}$ was estimated with 10^6 Monte Carlo samples.

3.10. [APPENDIX] - MEAN SOJOURN TIME WITH ALL-PHOTONIC REPEATERS

Here, we provide some additional examples of the behavior of the mean sojourn time (MST) in terms of the number of users, u, and the number of forwarding stations per repeater, k.

Figure 3.10 shows the MST for sequential and parallel distribution of packets, for increasing number of users and fixed k. In Figure 3.10a ($p = 1$, $N = 0$, $L = 1$ km, as in

Figure 3.4a, and $k = 7$), sequential distribution can support one more user than parallel distribution (up to 14). Nevertheless, parallel distribution provides a lower MST, although the advantage decreases as the number of users increases, since we approach the divergence. In Figure 3.10b ($p \approx 0.7$, $N = 0$, $L = 7.5$ km, as in Figure 3.4b, and $k = 2$), parallel distribution can support more users (up to 6, while the MST with sequential distribution diverges after 5 users). In this case, the advantage in MST provided by parallel distribution actually increases with increasing number of users, since the divergence happens earlier when packets are distributed sequentially.

In Figure 3.11, we provide the difference in MST between sequential and parallel distribution vs the number of users and the number of forwarding stations, for the large-budget use case from the main text. In this use case, we consider the all-photonic forwarding stations from ref. [139]. The number of intermediate repeaters between each user and the central repeater, N, is chosen to minimize the cost $N/(Lp)$. As explained in the main text, the optimal solution is $p \approx 0.7$: for $L = 7.5$ and $L = 30$ km, we have $N = 0$ and $N = 5$, respectively. In Figure 3.11, we provide the results for window sizes $w = 7$ and $w \to \infty$ (more examples can be found in the Appendix of our paper [89]). As discussed in the main text and in Appendix 3.9, parallel distribution can support more users than sequential distribution only when the resources are scarce, i.e., when w is small compared to n and k is small, and when the distances are also small. In the figure, this effect happens for $w = 7$ and $L = 7.5$ km.

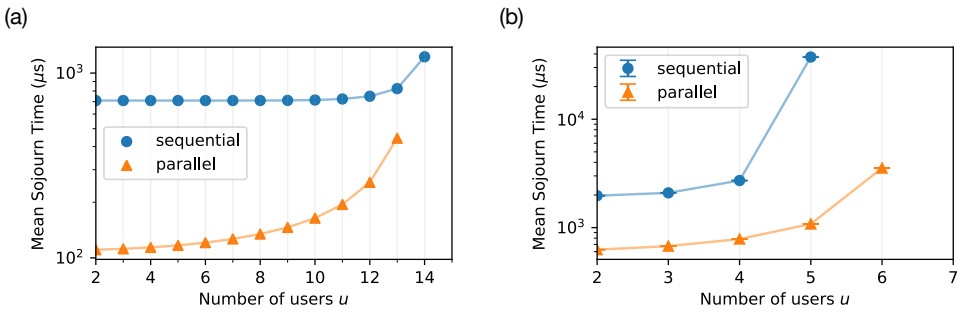

Figure 3.10: **Parallel distribution of packets is generally faster, and sometimes it supports more users.** MST with sequential (blue) and parallel (orange) packet distribution, for increasing number of users u. **(a)** Small-budget use case ($p = 1$; $N = 0$, $L = 1$ km) with $k = 7$ and **(b)** large-budget use case ($p \approx 0.7$) with $N = 0$, $L = 7.5$ km, and $w = 8$, and with $k = 2$. Parameters used in this figure: $n = 7$, $\lambda_0 = 10\ \mu s^{-4}$, $t_{\text{fwd}} = 100\ \mu s$. MST in (a) calculated with (3.7). MST in (b) calculated using a discrete-event simulation and Monte Carlo sampling with 10^5 samples (the error bars show the standard error).

3.11. [APPENDIX] - MANY USERS OVER LONG DISTANCES: ADDITIONAL EXAMPLES

In this Appendix, we compare sequential vs parallel distribution of packets in terms of the critical distance. As discussed in Section 3.3.3, increasing the number of repeaters only increases the critical distance when the number of users is small. Figure 3.5 shows

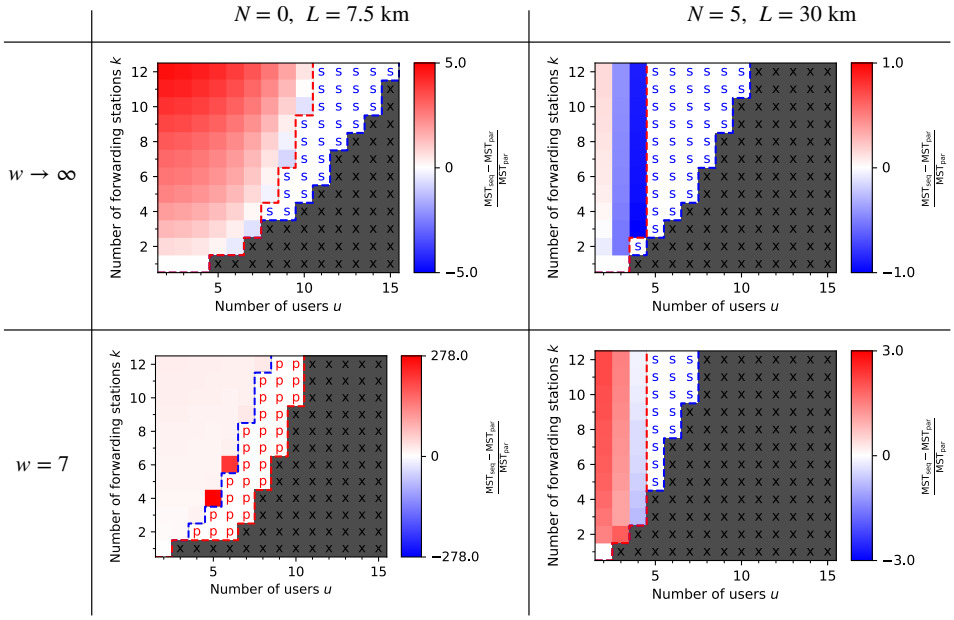

Figure 3.11: **Parallel distribution of packets is generally faster.** Relative difference in mean sojourn time (MST) between sequential and parallel packet distribution, for different numbers of users u and forwarding stations k, in the large-budget use case ($p \approx 0.7$) with $N = 0$ and $N = 5$ ($L = 7.5$ and $L = 30$ km, respectively; left to right subplots), and $w \to \infty$ (top) and $w = 7$ (bottom). Sequential/parallel distribution provides lower MST in blue/red regions. In regions with an 's'/'p', only sequential/parallel distribution can provide service (i.e., yield finite MST). In dark regions with an 'x', no service is possible. Parameters used in this figure: $n = 7$, $\lambda_0 = 10\ \mu s^{-4}$, $t_{fwd} = 100\ \mu s$. For $w \to \infty$, MST calculated with (3.7). Otherwise, MST calculated with Monte Carlo sampling with 10^7 samples (the standard error in the relative difference in MSTs was below 0.5 for every combination of parameters).

an example of this phenomenon when using sequential distribution of packets (the same example is shown in Figure 3.12a for convenience). In Figure 3.12b, we show that the same conclusions are observed when packets are distributed in parallel. Interestingly, when the number of users is fixed, sequential distribution (Figure 3.12a) allows for larger distances between them (i.e., it provides larger L_{crit}) than parallel distribution (Figure 3.12b). We also tested different combinations of parameters ($w = 7, 8, 10$ and $k = 6, 12, 18$) and observed very similar qualitative and quantitative behavior.

CODE AVAILABILITY

The code used to generate all the plots in this chapter can be found in the following GitHub repository: https://github.com/AlvaroGI/quantum-circuit-switching.

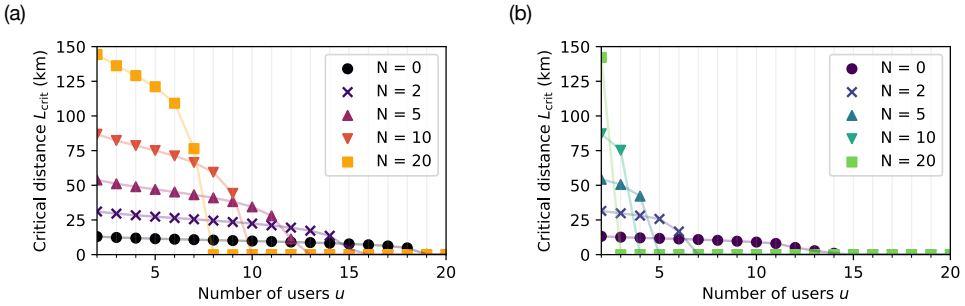

Figure 3.12: **Networks with sequential distribution of packets can cover longer distances than with parallel distribution.** Critical distance, L_{crit}, vs number of users, u, for different numbers of repeaters N in a star network. (a) Sequential distribution of packets; (b) parallel distribution of packets. Parameters used in this figure: $n = 7$, $w \to \infty$, $k = 12$, $\lambda_0 = 10^{-4}$ μs^{-1}, $c = 0.2$ km/μs, $t_{\text{fwd}} = 100$ μs. Results calculated using (3.8), where $\mathbb{E}\left[B_{n,w,p,m}\right]$ (required to solve (3.8)) was computed using the analytical solution from Appendix 3.5.2.

AUTHOR CONTRIBUTIONS

ÁGI and HC conceived the project. ÁGI, with input from HC, conducted the analysis and was the main writer of the paper [89]. SW and DE provided active feedback throughout the project.

4

Continuous Distribution of Entanglement in Multi-User Networks

Álvaro G. Iñesta and Stephanie Wehner

There is no way to reward performance
if there's no good way to measure performance.
— Tynan Sylvester

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

This chapter has been published separately in ref. [94].

4.1. INTRODUCTION

Quantum networks are expected to enable multi-party applications that are provably impossible by using only classical information. These applications range from basic routines, such as quantum teleportation [14, 72], to more complex tasks, such as quantum key distribution [12, 59] and entanglement-assisted distributed sensing [74, 202]. Some of these applications may operate in the *background* (e.g., a quantum key distribution subroutine that is continuously generating secret key), as opposed to *sporadic* applications that are executed after the users actively trigger them. Most quantum network applications consume shared entanglement as a basic resource. *Entanglement distribution protocols* are used to generate and share multipartite entanglement among remote parties. There are two main approaches to distribute entanglement among the nodes [35, 88]:

- Protocols for *on-demand distribution of entanglement* distribute entangled states only after some nodes request them. The request may involve some quality-of-service requirements (e.g., a minimum quality of the entanglement). This type of protocol typically involves solving a routing problem and scheduling a set of operations on a subset of nodes [22, 35, 91, 186, 188].

- Protocols for *continuous distribution of entanglement* (CD protocols) continuously distribute entangled states among the nodes. These entangled states can be consumed by the nodes whenever they need them. This allows *background applications* to continuously operate and consume entanglement in the background. In this work, we focus on CD protocols that provide entanglement to background applications.

On-demand distribution is generally more efficient, since entanglement is only produced when it is needed. This makes on-demand distribution more suitable for quantum networks where the quantum resources are limited (e.g., networks with a small number of qubits per node). As a consequence, previous work, both theoretical [22, 35, 91, 186, 188] and experimental [16, 83, 87, 130, 147, 172, 181], has mostly focused on this type of protocol in quantum networks with a simple topology or with very limited number of qubits per node.

On-demand distribution requires a scheduling policy that tells the nodes when to perform each operation based on specific demands. If the number of nodes involved in the generation of entanglement is large, the scheduling policies become more complex. In contrast, the continuous distribution of entanglement does not necessarily require an elaborate application-dependent schedule. Therefore, CD protocols are expected to allocate resources faster and prevent traffic congestion in large quantum networks. Here, we focus on the performance evaluation of CD protocols. Specifically, we consider protocols that distribute bipartite entanglement among remote nodes. We refer to shared bipartite entanglement as an *entangled link*. We focus on entangled links because this is a basic resource needed in many quantum network applications [10, 12, 59, 115], where nodes generally need many copies of a bipartite entangled state with high enough quality. Even when multipartite entanglement is required, it can be generated using entangled links [29, 107, 127, 145].

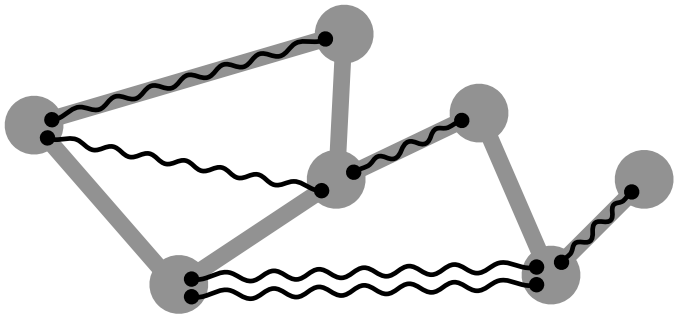

Figure 4.1: **Illustration of a seven-node quantum network.** The nodes are represented as gray circles, and physical channels connecting neighboring nodes are represented as gray lines. Entangled links are represented as black lines connecting two occupied qubits (small black circles). The physical topology is static, while the entangled links are continuously created, discarded, and consumed.

We consider a quantum network with n nodes. Some pairs of nodes are *physical neighbors*: they are connected by a physical channel, such as optical fibers [174, 206] or free space [169, 181]. This is depicted in Figure 4.1. To generate long-distance entanglement, we assume the nodes can perform the following basic operations: (i) heralded generation of entanglement between physical neighbors [9, 16], which successfully produces an entangled link with probability p_{gen} and otherwise raises a failure flag; (ii) entanglement swaps [55, 164, 209], which consume one entangled link between nodes A and B and another entangled link between nodes B and C to generate a single link between A and C with probability p_{s}; (iii) removal of any entangled link that has existed for longer than some cutoff time t_{cut} to prevent the existence of low-quality entanglement in the network [43, 103, 116, 160, 161]; and (iv) consumption of entangled links in background applications at some constant rate p_{cons}. Note that the choice of cutoff time is determined by the minimum fidelity required by the applications, F_{app}. We allow for multiple entangled links to be shared simultaneously between the same pair of nodes (see Figure 4.1).

Evaluating the performance of a CD protocol is a fundamentally different problem to evaluating the performance of on-demand protocols, since each type of protocol serves a different purpose. In on-demand protocols, one generally wants to maximize the rate of entanglement distribution among a specific set of end nodes and the quality of the entanglement (or some combined metric, such as the secret key rate [73]). By contrast, the goal of a CD protocol is (i) to distribute entanglement among the nodes such that it can be continuously consumed in background applications and (ii) to ensure that some entanglement is available for sporadic applications. To quantify the performance of a CD protocol, we need metrics that take these goals into account. A simple approach is to analyze the configuration of entangled links that a CD protocol can achieve. This configuration is time-dependent due to the dynamic nature of the entangled links. Most previous work aimed at describing the connectivity of large-scale quantum networks disregards the time-dependence of the system. As a consequence, previous results do not depend explicitly on parameters that determine the evolution of the entanglement, such as the coherence time. For example, in Refs. [25] and [24], the authors study a graph in which the edges are entangled links that exist at a specific instant. Some authors have

described the connectivity of a quantum network using percolation theory [2, 41, 47, 48, 128, 144, 200], which also disregards the time-evolution of the entangled states, and often assumes specific topologies and some form of pre-shared entanglement. Another line of related work is the use of pre-shared entanglement for on-demand applications [105, 149].

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

Our main contributions are the following:

- We define metrics to evaluate the performance of CD protocols in heterogeneous quantum networks with an arbitrary topology, namely, we define the *virtual neighborhood size* and the *virtual node degree*. These metrics provide information about the number of nodes that are able to continuously run background applications and about the number of existing backup entangled links to run sporadic applications.
- We provide analytical and numerical tools to compute the performance metrics.
- We provide a mathematical framework to maximize the virtual neighborhood size of every node in a heterogeneous network, while providing some minimum quality-of-service requirements (e.g., a minimum number of backup links). We do this via the concept of Pareto optimality.
- We study the relation between the steady-state performance of the entanglement distribution protocol and the application requirements (minimum fidelity and link consumption rate) in a quantum network with a tree topology.

Our main findings are the following:

- The expected virtual neighborhood size rapidly drops to zero when the entanglement consumption rate increases beyond the entanglement generation rate.
- In a quantum network with a tree topology and with high entanglement generation rate, the consumption rate has a stronger effect on the virtual neighborhood size than the minimum fidelity required by the applications. In other words, background applications that require a high consumption rate affect the CD protocol performance more than applications that require a high fidelity.
- The set of protocol parameters that maximize the virtual neighborhood size is node-dependent. Consequently, in heterogeneous networks with an arbitrary topology we need to solve a multi-objective optimization problem.

The structure of the chapter is as follows. In Section 4.2, we define the network model (physical topology, quantum operations, and quantum resources). In Section 4.3, we provide an example of a CD protocol. In Section 4.4, we formally define the virtual neighborhood and the virtual node degree. We apply these definitions to evaluate the performance of a CD protocol using analytical and numerical methods. As an example,

we analyze a CD protocol in a quantum network with a tree topology. In Section 4.5, we discuss the implications and limitations of our work.

4.2. NETWORK MODEL

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

We consider a quantum network with *n nodes* (see Figure 4.1). Nodes can store quantum states in the form of qubits, and they can manipulate them as we describe below. Additionally, some nodes are connected by a *physical channel* over which they can send quantum states. Qubits can be realized with different technologies, such as nitrogen vacancy (NV) centers [16, 83, 87, 147, 161], trapped ions [130, 172], or neutral atoms [195], while physical channels can be realized with optical fibers [174, 206] or free space [169, 181].

PHYSICAL TOPOLOGY. Two nodes are *physical neighbors* if they share a physical channel. The *physical node degree* d_i of node i is the number of its physical neighbors. The set of nodes and physical channels constitute the *physical topology* of the quantum network. Early quantum networks are expected to have simple physical topologies, such as a chain where each node is connected to two other nodes [22, 44, 91] and a star topology where all nodes are only connected to a central node [185, 186]. More advanced networks are expected to display a more complex physical topology, such as a dumbbell structure with a backbone connecting two metropolitan areas.

The definitions and methods we develop in this work are general and apply to an arbitrary physical topology, which can be described using an *adjacency matrix* A (element A_{ij} is 1 if nodes i and j are physical neighbors and 0 otherwise). To illustrate how our methods can be valuable and effective, we apply them to a quantum network with a tree topology as an example. In a tree, any node can be reached from any other node by following exactly one path. This topology is particularly relevant as it has been shown that it requires a reduced number of qubits per node to avoid traffic congestion [40].

Definition 4.1. *A* (d,k)-tree *network is an undirected unweighted graph where nodes are distributed in* k *levels, with* d^l *nodes in level* $l \in 0,1,\ldots,k-1$. *Each node in level* l *is connected to* d *nodes in the* $(l+1)$*-th level, and is only connected to one node in the* $(l-1)$*-th level.*

The total number of nodes in a (d,k)-tree is $n = (d^k - 1)/(d-1)$, and the network diameter is $2k$. A (2,3)-tree network is depicted in Figure 4.2.

ENTANGLEMENT DISTRIBUTION. The aim of a CD protocol is to distribute shared bipartite entangled states, which we call entangled links. Ideally, entangled links are maximally entangled states. However, entanglement generation and storage are generally noisy processes. Consequently, we assume that entangled links are Werner states [197]:

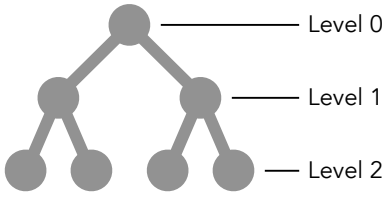

Figure 4.2: **(2,3)-tree network.** Each node is represented as a gray circle and is connected to two other nodes in a lower level.

maximally entangled states that have been subjected to a depolarizing process, which is a worst-case noise model [58]. Werner states can be written as

$$\rho = \frac{4F-1}{3}\left|\phi^+\right\rangle\!\left\langle\phi^+\right| + \frac{1-F}{3}\mathbb{I}_4, \tag{4.1}$$

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

$$F(\rho, \left|\phi\right\rangle) := \left\langle\phi\right|\rho\left|\phi\right\rangle. \tag{4.2}$$

We consider nodes that operate as first or second generation quantum repeaters [136]: physical neighbors generate entangled links via *heralded entanglement generation* using two-way signaling. This operation produces an entangled link with probability p_{gen} and otherwise raises a failure flag [9, 16]. The fidelity of newly generated links, F_{new} is generally a function of p_{gen}. For example, in the single-photon protocol [84], $F_{\text{new}} = 1 - \lambda p_{\text{gen}}$, for some $0 \leq \lambda \leq 1$ (as discussed in ref. [49], the value of λ can be tuned by performing a batch of entanglement attempts as a single entanglement generation step [148]).

Long-distance entanglement between physically non-neighboring nodes can be generated using *entanglement swapping* [55, 164, 209], which consumes an entangled link between nodes A and B, with fidelity F_{AB}, and another one between B and C, with fidelity F_{BC}, to produce a link between A and C, with fidelity $F_{\text{AC}} \leq F_{\text{AB}}, F_{\text{BC}}$. This operation succeeds with probability p_{swap} (when it fails, both input links are lost and nothing is produced). Note that entanglement swapping also requires two-way classical signaling. See Appendix 4.6 for further details on entanglement swapping.

QUANTUM APPLICATIONS. The main goal of a CD protocol is to provide a continuous supply of entanglement for nodes to run applications without the need for explicitly demanding entanglement. We assume that each pair of nodes that share entanglement is continuously running quantum applications in the background, consuming entangled links at a rate p_{cons}. For simplicity, we assume $0 \leq p_{\text{cons}} \leq 1$. Since we will assume time to be slotted (see Section 4.3), a consumption rate between zero and one can be interpreted as the probability that, in each time slot, two nodes that share some entangled links consume one link. We consider entanglement purification [57, 81, 188] as an application and therefore we omit it in our model (purification at the physical link level can be included in our model by modifying p_{gen} and F_{new} accordingly; see Appendix 4.6 for further details).

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

MITIGATING DECOHERENCE. The operations involving entangled links and the storage in memory have a negative impact on the quality of the links. Each entanglement swap produces a link with a lower fidelity than the input links [132]. To prevent the fidelity from dropping too low, we must limit the *maximum swap distance*, defined as the maximum number of short-distance links that can be combined into longer distance entanglement via swaps. We denote this maximum number of links as M. Two nodes can only share entanglement if they are at most M physical links away.

Additionally, the fidelity of entangled links stored in memory decreases over time due to couplings to the environment [38, 58], making old links unusable for applications that require high fidelity states. A simple technique to alleviate the effects of noisy storage consists in imposing a cutoff time t_{cut}: any link that has been stored for longer than the cutoff time must be discarded [160].

To ensure that the fidelity of every entangled link is above F_{app} in a network where new links are generated with fidelity F_{new}, it is enough to choose the values of t_{cut} and M such that [91]

$$t_{\text{cut}} \leq -T \ln\left(\frac{3}{4F_{\text{new}}-1}\left(\frac{4F_{\text{app}}-1}{3}\right)^{\frac{1}{M}}\right), \tag{4.3}$$

where T is a parameter that characterizes the exponential decay in fidelity of the whole entangled state due to the qubits being stored in noisy memories (see Section 2.7 for a derivation of (4.3)). In our analysis, we choose the largest cutoff that satisfies (4.3). For further details on the noise model, see Appendix 4.6.

LIMITED QUANTUM RESOURCES. Nodes have a limited number of qubits. These qubits can be used for communication (short coherence times) or for storage (long coherence times) [11, 114]. Here, we assume a simplified setup where every qubit can be used for entanglement generation and for storage of an entangled link. Intuitively, nodes with a larger number of physical neighbors should have more resources available, to establish entanglement with many neighbors simultaneously. We assume that the maximum number of qubits that node i can store is $d_i r$, where d_i is the physical node degree of node i and $r \in \mathbb{N}$ is a hardware-dependent parameter that limits the maximum number of qubits per node.

We make an additional simplifying assumption: each qubit can only generate entanglement with a fixed neighboring node. The physical motivation behind this assumption is the lack of optical switches in the node. This assumption allows us to uniquely identify each qubit using a three-tuple address (i, j, m). The first index, $i \in \{0, ..., n-1\}$, corresponds to the node holding the qubit. The second index, $j \in \{0, ..., n-1\}$, is the node with which the qubit can generate entanglement ($i \neq j$). The third index, $m \in \{0, ..., r-1\}$, is used to distinguish qubits that share the same indices i and j. A graphical example is shown in Figure 4.3.

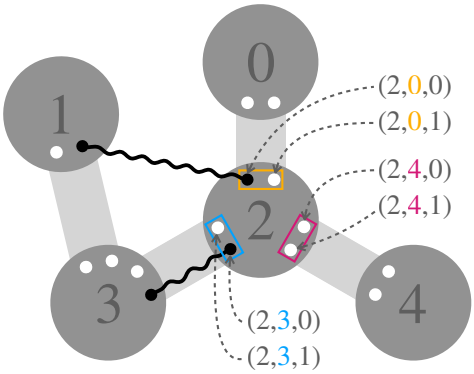

Figure 4.3: **Qubit addresses.** Each qubit is identified by a qubit address consisting of three values (i, j, m): i is the node holding the qubit, j is the neighboring node that can generate entanglement with that qubit, and m is used to distinguish qubits with the first two indices i and j. In this example, each node has two qubits per physical neighbor, i.e., $r = 2$.

4.3. PROTOCOL FOR CONTINUOUS DISTRIBUTION OF ENTANGLEMENT

The operations discussed above – entanglement generation, swaps, entanglement consumption, and application of cutoffs – are performed following a specific protocol for continuous distribution of entanglement (CD protocol). Here we consider a basic CD protocol that we will use to test our performance optimization tools. We assume a synchronous protocol: time is divided into non-overlapping time slots and each operation is allocated within a time slot. This is a common assumption in the field of quantum networking (see, e.g., Refs. [91, 171]), since nodes generally have to agree to perform synchronized actions for heralded entanglement generation. In what follows, we focus on the Single Random Swap (SRS) protocol, which is described in Algorithm 1. In this protocol, (i) entanglement generation is attempted sequentially on every physical link; (ii) swaps are performed using links chosen at random; and (iii) every pair of nodes that shares an entangled link consumes one link per time step with probability p_{cons}. The protocol has a single parameter, $q \in [0, 1]$, which determines how many nodes must perform a swap at each time step (if $q = 0$, no swaps are performed; if $q = 1$, every node must perform a swap if possible; if $0 < q < 1$, a random subset of nodes may perform swaps). In step 3.2 of the SRS protocol, the condition $A_{jk} = 0$, ensures that swaps will generally not connect physical neighbors.

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

In Table 4.1 we provide a summary of the network and protocol parameters. In the next section we present our performance metrics and how to use them to tune the protocol

Table 4.1: **Parameters of the quantum network.** The number of nodes is given by the size of the adjacency matrix A. When considering a (d,k)-tree topology, the adjacency matrix A can be replaced by d and k. The cutoff time t_{cut} is given by p_{gen}, F_{new}, F_{app}, and M via (4.3).

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

parameter(s) for an optimal performance. Note that our methods can be applied to any other (synchronous and non-synchronous) CD protocol.

4.4. PERFORMANCE EVALUATION

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

Definition 4.2. *In a quantum network, the* virtual neighborhood *of node i, $V_i(t)$, is the set of nodes that share an entangled link with node i at time t. Two nodes are* virtual neighbors *if they share at least one entangled link. The* virtual neighborhood size *is denoted as $v_i(t) := |V_i(t)|$.*

Definition 4.3. *In a quantum network, the virtual node degree of node i, $k_i(t)$, is the number of entangled links connected to node i at time t.*

The virtual neighborhood size and the virtual node degree combined are useful metrics to evaluate the performance of a CD protocol. The size of the virtual neighborhood

Algorithm 1 - SRS entanglement generation protocol.

Inputs:

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

Outputs:

- Quantum network with updated configuration of links.

Algorithm:

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

of node i corresponds to the number of nodes that can run background applications together with node i. Since our model includes consumption of entanglement in such applications, the virtual degree provides information about how many resources are left to run sporadic applications.

The definitions above are similar to the notions of node neighborhood and node degree in classical graph theory. However, the configuration of entangled links changes over time, and therefore performance metrics from graph theory are ill-suited for this problem, as they generally do not include this type of time-dependence. In contrast to those metrics, $\nu_i(t)$ and $k_i(t)$ are not random variables but stochastic processes, i.e., the value at each time slot is a random variable.

When consuming entanglement at a constant rate, the steady state of the system is of particular interest since it will provide information about the performance of the protocols in the long term. In Appendix 4.7, we show that, when running the SRS protocol (Algorithm 1), the network undergoes a transient state and then reaches a unique steady-state regime (the proof also applies to similar CD protocols that use heralded entanglement generation, entanglement swaps, and cutoffs). In what follows, we will focus on evaluating the performance of the protocol during the steady state via the steady-state expected value of the virtual neighborhood size, $\nu_i \equiv \lim_{t\to\infty} \mathbb{E}\big[\nu_i(t)\big]$, and the virtual node degree, $k_i \equiv \lim_{t\to\infty} \mathbb{E}\big[k_i(t)\big]$.

Next, in Subsection 4.4.1, we analyze the behavior of ν_i and k_i in the absence of swaps. In 4.4.2, we analyze the relationship between these metrics and the protocol parameter q in a tree-like network (although our methods are general and apply to any arbitrary topology) and we find the optimal q that maximizes the virtual neighborhood size of the nodes in the lowest level of the tree. In 4.4.3, we provide a mathematical framework, based on Pareto optimization, to provide a good quality of service in heterogeneous networks.

4.4.1. NO SWAPS

To gain some intuition about the dynamics of the network and to set a benchmark, we consider the SRS protocol with $q = 0$, i.e., no swaps. In the absence of swaps, only physical neighbors can share entanglement, and the virtual neighborhood size and the virtual node degree of node i in the steady state are given by

$$\nu_i \equiv \lim_{t\to\infty} \mathbb{E}\big[\nu_i(t)\big] = d_i \frac{1 - \frac{1-p_{\text{cons}}}{1-p_{\text{gen}}}\lambda^r}{1 - \frac{p_{\text{cons}}}{p_{\text{gen}}}\lambda^r}, \tag{4.4}$$

$$k_i \equiv \lim_{t\to\infty} \mathbb{E}\big[k_i(t)\big] = d_i\, p_{\text{gen}} \frac{r + \frac{p_{\text{cons}}(1-p_{\text{cons}})}{p_{\text{gen}}-p_{\text{cons}}}(\lambda^r - 1)}{p_{\text{gen}} - p_{\text{cons}}\lambda^r}, \tag{4.5}$$

where $\lambda \equiv \frac{p_{\text{cons}}(1-p_{\text{gen}})}{p_{\text{gen}}(1-p_{\text{cons}})}$; p_{gen} is the probability of successful entanglement generation at the physical link level; p_{cons} is the link consumption rate; d_i is the physical node degree of node i; and r is the number of qubits available at node i per physical neighbor. (4.4) and (4.5) are derived in Appendix 4.8 using general random walks. Note that in the derivation we assume large enough cutoffs, such that links are consumed with a high enough probability before reaching the cutoff time.

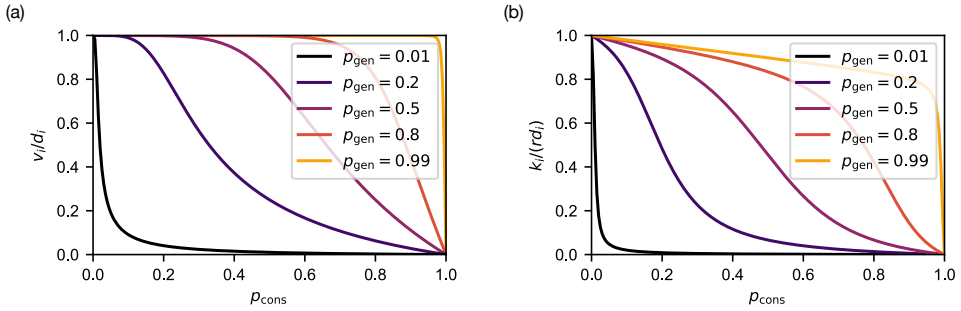

Figure 4.4: **Larger consumption rates decrease the virtual neighborhood size and the virtual node degree.** Expected virtual neighborhood size (a) and virtual node degree (b) in the steady state in a quantum network with no swaps, with cutoff $t_{cut} = 10/p_{cons}$ time steps, and with five qubits per node per physical channel ($r = 5$). Both quantities are normalized by the physical degree of node i, d_i. The curves were calculated using (4.4) and (4.5).

In the absence of swaps, both v_i and k_i are proportional to the physical node degree d_i but independent of the rest of the physical topology. This allows us to study these performance metrics without assuming any specific physical topology. Figure 4.4 shows the analytical solution for v_i and k_i when each node has five qubits per physical neighbor ($r = 5$). The figure shows a transition from large to small virtual neighborhood size when increasing p_{cons} beyond p_{gen}. When the consumption rate is smaller than the generation rate, the size of the virtual neighborhood saturates and converges to the number of physical neighbors. When p_{cons} increases beyond p_{gen}, the virtual neighborhood size goes to zero. A similar behavior is observed for the virtual degree, which takes larger values for $p_{cons} < p_{gen}$. The same behavior is observed for different values of r, as shown in Appendix 4.8.

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

In Appendix 4.8, we use simulations to show that $\mathbb{E}[v_i(t)]$ and $\mathbb{E}[k_i(t)]$ indeed converge to the steady-state values predicted by our analytical calculations as t goes to infinity.

4.4.2. HOMOGENEOUS SET OF USERS

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

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

when they are successful). In what follows, we show how to optimize the SRS protocol in a quantum network with a (2,3)-tree topology (although our methods apply to any quantum network and any CD protocol). This tree network is particularly interesting because it corresponds to a dumbbell network, which could be used to model users (level-2 nodes) in two metropolitan areas (level-1 nodes) connected by a central link via the level-zero node. If we assume distances of the order of 10 km, the communication time over optical fibers is of the order of 1 ms. Hence, the time step must be at least of the order of 1 ms. For demonstration purposes, we assume a coherence time of $T = 2000$ time steps, which is of the order of 1 s. As a reference, state-of-the-art coherence times lie between milliseconds (e.g., $T \approx 11.6$ ms in the NV centers experiment from ref. [147]) and seconds (e.g., $T \approx 50$ s in the trapped-ion experiment from ref. [82]). Additionally, also for demonstration purposes, we assume probabilistic entanglement generation, deterministic swaps, maximum swap distance $M = 4$ (such that every node can share links with every other node), and background applications that can be executed with low fidelity links ($F_{\text{app}} = 0.6$). We analyzed the system by simulating the evolution of the network over time and using Monte Carlo sampling. For further details about how we find the steady state and how we compute expectation values from simulation data see Appendix 4.9.

Figure 4.5 shows v_i and k_i for three different nodes. Due to the symmetry of the topology, every node in the same level of the tree has the same statistical behavior. Therefore, we can describe the behavior of the whole tree network by looking at one node per level. When no swaps are performed ($q = 0$), the virtual neighborhood size v_i (Figure 4.5a) is upper bounded by the number of physical neighbors d_i ($d_i = 2, 3$, and 1, for nodes in level 0, 1, and 2, respectively). Increasing q leads to an increase in v_i, which reaches a maximum value before decreasing again. If too many swaps are performed (q close to 1), then v_i decreases, since each swapping operation consumes two links and produces only one. The maximum virtual neighborhood size, $\max_q v_i$, is achieved at a different value of q for each node. The virtual node degree k_i (Figure 4.5b) behaves qualitatively in a similar way for every node: it is maximized at $q = 0$ and, as we perform more swaps (increasing q), more links are swapped and fewer links remain in the system. A similar behavior was observed for larger trees and for probabilistic swaps (see Appendix 4.10).

In some cases, we may be only interested in providing a good service to a subset of nodes U, the *user nodes*. The users run applications but also perform swaps to support the entanglement distribution among other pairs of users. The only purpose of the rest of the nodes (*repeater nodes*) is to aid the users to meet their needs. In the literature, users that consume entanglement, but do not perform swaps to help other nodes, are generally called *end nodes*. Here we assume every node is a user or a repeater node. When some nodes are users and some are repeaters, the performance metrics of repeater nodes become irrelevant and we want to maximize v_i, $\forall\, i \in U$. When the set of users is homogeneous (i.e., all user nodes have the same properties), the statistical behavior of all users is the same and we can formulate a single-objective optimization problem where we want to maximize v_i for a single $i \in U$. For example, in a tree quantum network, users are generally the nodes at the lowest level [40]. In the example from Figure 4.5, the lowest-level nodes are the level-2 nodes (green line with crosses). If the level-2 nodes are the only users, the performance of the protocol is optimized for $q \approx 0.65$, which maximizes their

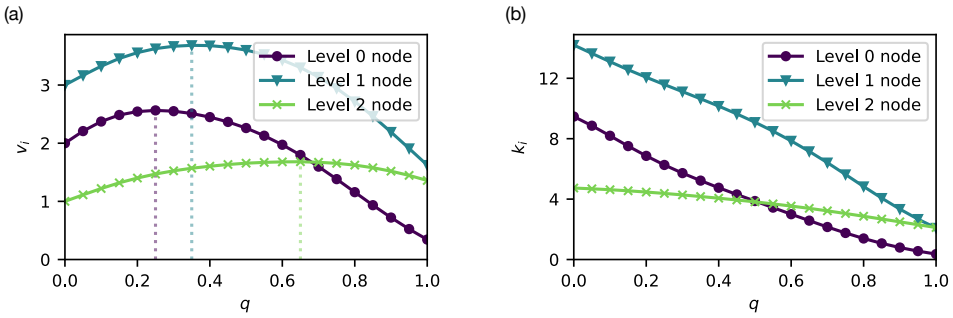

Figure 4.5: **The virtual neighborhood size of every node cannot be maximized simultaneously.** Expected virtual neighborhood size (a) and virtual node degree (b) in the steady state in a (2,3)-tree network running the SRS protocol vs the protocol parameter q. The value of q that maximizes the virtual neighborhood size, indicated by the dotted lines, is node-dependent. The virtual node degree decreases monotonically with increasing q, since more links are consumed in swaps when q is large. Other parameter values used in this experiment: $p_{\text{gen}} = 0.9$, $F_{\text{new}} = 0.888$, $p_{\text{swap}} = 1$, $r = 5$, $T = 2000$ time steps, $M = 4$, $p_{\text{cons}} = p_{\text{gen}}/4 = 0.225$, $F_{\text{app}} = 0.6$, $t_{\text{cut}} = 56$ time steps (given by (4.3)). Results obtained using a network simulation and Monte Carlo sampling with 10^6 samples. Error bars are not shown since they are smaller than the line width – the standard errors are below 0.003 and 0.006 for the v_i and k_i, respectively. The standard error is defined as $2\hat{\sigma}/(N_{\text{samples}})^{0.5}$, where $\hat{\sigma}$ is the sample standard deviation and N_{samples} is the number of samples.

v_i. The protocol optimization problem also becomes a single-objective optimization problem in other networks with a strong symmetry, such as regular networks [177].

In Figure 4.6 we consider a (2,3)-tree network where the users are the nodes at the lowest level, and we study the influence of the background application requirements (F_{app} and p_{cons}) on the maximum expected virtual neighborhood size of the users. Here, we assume that the entanglement generation rate is much larger than the consumption rate ($p_{\text{gen}} \geq 3p_{\text{cons}}$). Otherwise, links are consumed shortly after they are generated and the behavior of the system is not interesting, as discussed in 4.4.1. From the figure, we observe that the consumption rate has a stronger effect on the virtual neighborhood. For example, for $F_{\text{app}} = 0.8$, decreasing p_{cons} from 0.3 to 0.1 increases the maximum expected virtual neighborhood size by 20.3%. However, when decreasing F_{app} from 0.8 to 0.5, the maximum increase in v_i is 3.5% (for $p_{\text{cons}} = 0$). The consumption rate has a bigger effect on the virtual neighborhood because it directly impacts the configuration of virtual links, while F_{app} only affects links via the cutoff. In this case, the smallest cutoff is 17 time steps for $F_{\text{app}} = 0.8$ and the largest is 411 time steps for $F_{\text{app}} = 0.5$. When the generation rate is large, virtual neighbors are likely to share multiple entangled links. In that case, cutoffs barely impact the virtual neighborhood size since links can be regenerated quickly and they are only removed after some time t_{cut}. However, link consumption can still have a strong impact on the virtual neighborhood size since any link can be consumed at any time step. If the cutoffs are very close to unity (e.g., when applications require a fidelity $F_{\text{app}} > 0.8$), the cutoff value may strongly affect the virtual neighborhood size.

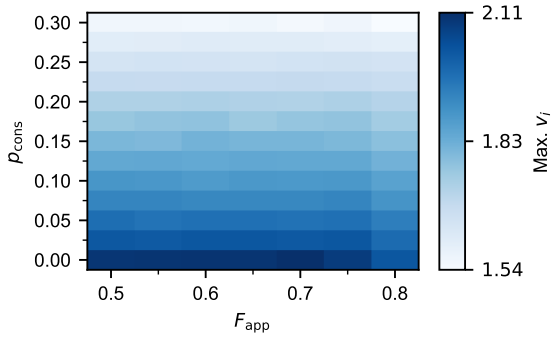

Figure 4.6: **The consumption rate has a stronger impact on the performance than the application fidelity when the entanglement generation rate is high.** Maximum virtual neighborhood size (maximized over q) of a layer-2 node in a (2,3)-tree network vs the application fidelity, F_{app}, and the consumption rate, p_{cons}. Other parameter values used in this experiment: $p_{\text{gen}} = 0.9$, $F_{\text{new}} = 0.95$, $p_{\text{swap}} = 1$, $r = 5$, $T = 2000$ time steps, $M = 4$. The cutoff time t_{cut} is given by (4.3). Results obtained using a network simulation and Monte Carlo sampling with 10^4 samples. The maximum error is 0.015 (the error is defined as $2\hat{\sigma}/N_{\text{samples}}^{0.5}$, where $\hat{\sigma}$ is the sample standard deviation and N_{samples} is the number of samples). Note that $\max_q(v_i)$ should be monotonic in F_{app} and p_{cons} but in this plot we observer small deviations due to the sample size.

4.4.3. HETEROGENEOUS SET OF USERS: MULTI-OBJECTIVE OPTIMIZATION

In a more general topology, the user nodes may have different properties and different physical degrees. In that case, the size of the virtual neighborhood of each user may be maximized for a different value of q. Hence, optimizing the protocol for node i generally means that the protocol will be suboptimal for some other node $j \neq i$. This leads to a multi-objective optimization problem where we must find a tradeoff between the variables that we want to maximize. In such a problem, optimality can be defined in different ways [126]. A practical definition is the Pareto frontier:

Definition 4.4. *Let U be the set of user nodes. Let $\vec{\theta} \in \Theta$ be a combination of parameter values describing the topology, the hardware, and the software of the quantum network, where Θ is the parameter space. Let $v_i(\vec{\theta})$, with $i \in U$, be the set of variables that we want to maximize. The* Pareto frontier *is defined as*

$$P = \left\{ \vec{\theta} \;\middle|\; \forall \vec{\theta}' \in \Theta \; \exists i \text{ s.t. } v_i(\vec{\theta}) \geq v_i(\vec{\theta}') \right\}. \tag{4.6}$$

Lemma 4.1. *If the parameter space is non-empty, i.e., $\Theta \neq \emptyset$, then the Pareto frontier is non-empty, i.e., $P \neq \emptyset$.*

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

Note that the parameter space Θ can be a constrained space, i.e., it does not necessarily include all combinations of parameters values. For example, combinations of parameters that are experimentally unfeasible may be excluded from Θ. The Pareto frontier achieves

a tradeoff in maximizing every v_i, $i \in U$. For all the points $\vec{\theta}$ in the Pareto frontier, we cannot obtain an increase in $v_i(\vec{\theta})$ without decreasing or keeping constant some other $v_j(\vec{\theta})$. Moreover, Lemma 4.1 ensures that there is at least one point $\vec{\theta}$ in the Pareto frontier.

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

$$Q = \left\{ \vec{\theta} \;\middle|\; v_i(\vec{\theta}) \geq c_i \right\}. \tag{4.7}$$

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

Definition 4.5. *The* optimal region P^* *is the set of parameters that are in the Pareto frontier and meet the quality-of-service requirements, i.e.,*

$$P^* = P \cap Q, \tag{4.8}$$

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

As an example, we consider a $(2,3)$-tree network where the nodes in levels 1 and 2 are users. Due to the symmetry of the topology, we only need to explicitly optimize v_i for one node in each level. In this case, it is possible to provide a graphical representation of the Pareto frontier and the optimal region. Figure 4.7 shows the expected virtual neighborhood size in the steady state for a level-1 user and a level-2 user in a quantum network with a (2,3)-tree topology running the SRS protocol with probabilistic entanglement generation, deterministic swaps, and entanglement consumption at a fixed rate. Each data point corresponds to a different value of the protocol parameter q. The data points highlighted with blue crosses form the Pareto frontier P. In this example, we want the users in the first and second level to have an expected virtual neighborhood size larger than 3 and 1.6, respectively. Then,

$$Q = \left\{ \vec{\theta} \;\middle|\; v_1(\vec{\theta}) \geq 3, v_2(\vec{\theta}) \geq 1.6 \right\}. \tag{4.9}$$

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

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

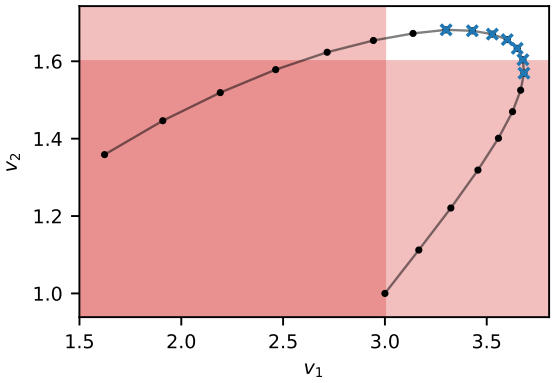

Figure 4.7: **The optimal region determines the combinations of parameters that provide an optimal performance.** Virtual neighborhood size of a level-1 node, v_1, and a level-2 node, v_2, in a (2,3)-tree network running the SRS protocol for different values of the protocol parameter q (for $q = 0$, we have $v_1 = 3$ and $v_2 = 1$; we increase q in intervals of 0.05 following the black line up to $q = 1$). The data points with blue crosses form the Pareto frontier P. The regions shaded in red are forbidden by the quality-of-service requirements ($c_1 = 3$, $c_2 = 1.6$). The optimal region P^* is formed by the blue crosses in the white region. Other parameter values used in this experiment: $p_{\text{gen}} = 0.9$, $F_{\text{new}} = 0.888$, $p_{\text{swap}} = 1$, $r = 5$, $T = 2000$ time steps, $M = 4$, $p_{\text{cons}} = p_{\text{gen}}/4 = 0.225$, $F_{\text{app}} = 0.6$, $t_{\text{cut}} = 56$ time steps (given by (4.3)). Results obtained using a network simulation and Monte Carlo sampling with 10^6 samples. Error bars are not shown since they are smaller than the line width – the standard errors are below 0.003 and 0.002 for v_1 and v_2, respectively. The standard error is defined as $2\hat{\sigma}/(N_{\text{samples}})^{0.5}$, where $\hat{\sigma}$ is the sample standard deviation and N_{samples} is the number of samples.

4.5. DISCUSSION

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

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

Our formulation also allows the study of protocols that continuously distribute entanglement to maintain a supply of high quality pre-shared entanglement. Specifically, the SRS protocol described in Algorithm 1 delivers pre-shared entanglement when the consumption rate is set to zero. This can be useful to determine the feasibility of quantum network protocols that assume pre-shared entanglement among the nodes of the network. In this case, an application that uses the available entanglement would disrupt the distribution of entangled states and would bring the system to a new transient state.

Hence, an additional useful metric would be the time required to converge to a steady state after such a disruption. We leave this analysis for future work.

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

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

4.6. [APPENDIX] FURTHER DETAILS ON THE NETWORK MODEL

ENTANGLEMENT SWAP. Two nodes that are not physical neighbors cannot generate entanglement directly between them. Instead, they rely on entanglement swap operations to produce a shared entangled state between them [55, 164, 209]. As an example, consider two end nodes A and B, which are not physically connected but share a physical link with an intermediate node C. To generate an entangled link between A and B, they need to first generate entangled links between A and C, and also between C and B. Then, node C can perform a Bell state measurement to transform links A-C and C-B into a single entangled link between A and B. When both input links are Werner states with fidelities F_1 and F_2, the output state in a swap operation is also a Werner state with fidelity [132]

$$F_{\text{swap}}(F_1, F_2) = F_1 \cdot F_2 + \frac{(1-F_1)\cdot(1-F_2)}{3}. \tag{4.10}$$

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

Additionally, entanglement swaps can be either probabilistic [32, 55, 62] or deterministic [147], depending on the hardware employed. With probability p_s, the swap operation succeeds and both input states are consumed to produce a single entangled link. With probability $1 - p_s$, the swap operation fails: both input states are consumed but no other entangled state is produced.

PURIFICATION. If the fidelity of an entangled state is not large enough for a specific application, nodes can run a purification protocol to increase its fidelity. In general, these protocols take as input multiple entangled states and output a single state with larger fidelity [57, 81, 188].

For simplicity, we do not consider any kind of purification in our analysis. Nevertheless, it is possible to integrate purification of entangled states at the physical link level into our model by decreasing the value of p_{gen}, to account for all the states that must be prepared in advance to perform the purification protocol. This would also impact the fidelity of newly generated links, F_{new}, which would correspond now to the fidelity of the links after purification at the physical link level. The cutoff time would also need to be

adjusted, since the time step would take a longer time (it would have to include more than one entanglement generation attempt). If physically distant nodes require larger fidelity links, they can run a purification subroutine as part of the application once they have generated enough entangled links.

CUTOFF TIMES. Quantum states decohere, mainly due to environmental couplings [38, 58]. Decoherence decreases the fidelity of states over time. We consider a depolarizing noise model, which is a worst-case scenario (other types of noise can be converted to depolarizing noise via twirling [30, 58, 86]). As shown in Section 2.6 and in Appendix A from ref. [91], if we assume that each qubit of a Werner state is stored in a different memory and experiences depolarizing noise independently, the fidelity of the Werner states evolves as

$$F(t+\Delta t) = \frac{1}{4} + \left(F(t) - \frac{1}{4}\right) e^{-\frac{\Delta t}{T}}, \tag{4.11}$$

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

When the fidelity of the entangled links drops below some threshold, they are no longer useful. Hence, a common practice is to discard states after a cutoff time t_{cut} to prevent wasting resources on states that should not be used anymore [160]. We refer to the time passed since the creation of a quantum state as the age of the state. Whenever the age of an entangled state equals the cutoff time, the state is removed, i.e., the qubits involved are reset. As shown in Section 2.7 and in ref. [91], to ensure that any two nodes that are at most M physical links away will only share entangled states with fidelity larger than F_{app}, the cutoff time must satisfy

$$t_{\text{cut}} \leq -T \ln\left(\frac{3}{4F_{\text{new}}-1}\left(\frac{4F_{\text{app}}-1}{3}\right)^{1/M}\right), \tag{4.12}$$

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

4.7. [APPENDIX] EXISTENCE OF A UNIQUE STEADY STATE

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

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

The state of the network can be represented using the ages of all entangled links present in the network (the age is measured in number of time slots). This can be written

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

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

$$\Pr\big[s(t+1) = \sigma \mid s(0), s(1), \ldots, s(t)\big] = \Pr\big[s(t+1) = \sigma \mid s(t)\big].$$

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

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

2. The chain is aperiodic. A sufficient condition for an irreducible chain to be aperiodic is that $\Pr\big[s(t+1) = \sigma \mid s(t) = \sigma\big] > 0$ for some state $\sigma \in \mathscr{S}$ [184]. When entanglement generation is probabilistic ($p_{\text{gen}} < 1$), the state with no entangled links satisfies the previous condition (if all entanglement generation attempts fail, the network will remain in a state with no links), and therefore the chain is aperiodic.

3. The chain is positive recurrent (i.e., the mean time to return to any state is finite), since it is irreducible and it has a finite state space $\mathscr{S}$ (see Theorem 9.3.5 from ref. [184]).

According to Theorem 9.3.6 from ref. [184], from the three properties above we can conclude that there exists a unique steady-state probability distribution, i.e., the following limit exists: $\lim_{t\to\infty} \Pr\big[s(t) = \sigma\big]$, $\forall \sigma \in \mathscr{S}$.

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

$$\begin{aligned}
v_i \equiv \lim_{t\to\infty} \mathbb{E}\big[v_i(t)\big] &= \lim_{t\to\infty} \sum_{v=0}^{n} v \cdot \Pr\big[v_i(t) = v\big] \\
&= \sum_{v=0}^{n} v \cdot \lim_{t\to\infty} \Pr\big[v_i(t) = v\big] \\
&= \sum_{v=0}^{n} v \cdot \lim_{t\to\infty} \sum_{\sigma\in\mathscr{S}} \Pr\big[v_i(t) = v \mid s(t) = \sigma\big] \cdot \Pr\big[s(t) = \sigma\big] \\
&= \sum_{v=0}^{n} v \cdot \sum_{\sigma\in\mathscr{S}} \lim_{t\to\infty} \Pr\big[v_i(t) = v \mid s(t) = \sigma\big] \cdot \lim_{t\to\infty} \Pr\big[s(t) = \sigma\big] \\
&= \sum_{\sigma\in\mathscr{S}} \lim_{t\to\infty} \Pr\big[s(t) = \sigma\big] \cdot \sum_{v=0}^{n} v \cdot \lim_{t\to\infty} \Pr\big[v_i(t) = v \mid s(t) = \sigma\big].
\end{aligned} \tag{4.13}$$

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

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

Consequently,

$$\Pr\big[v_i(t) = v \mid s(t) = \sigma\big] = \begin{cases} 1, & \text{if } v = \sum_{j\in V\setminus\{i\}} \min\big(1,\kappa(\sigma;i,j)\big) \\ 0, & \text{otherwise} \end{cases}. \tag{4.14}$$

Using (4.14), we can write (4.13) as

$$v_i = \sum_{\sigma\in\mathscr{S}} \lim_{t\to\infty} \Pr\big[s(t)=\sigma\big] \sum_{j\in V\setminus\{i\}} \min\big(1,\kappa(\sigma;i,j)\big), \tag{4.15}$$

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

$$k_i \equiv \lim_{t\to\infty} \mathbb{E}\big[k_i(t)\big] = \sum_{\sigma\in\mathscr{S}} \lim_{t\to\infty} \Pr\big[s(t)=\sigma\big] \sum_{j\in V\setminus\{i\}} \kappa\big(\sigma;i,j\big). \tag{4.16}$$

Since we have shown that the probability distributions that appear in (4.15) and (4.16) exist and are unique, then the quantities v_i and k_i also exist and are unique. That is, there is a unique steady-state value for the expected number of virtual neighbors and the expected virtual degree of any node i.

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

Lastly, note that in practice one may find an initial transient state with periodic behavior. This happens in quasi-deterministic systems, i.e., systems in which all probabilistic events (e.g., successful entanglement generation) happen with probability very close to 1. In quasi-deterministic systems, all realizations of the stochastic processes are identical at the beginning with a very large probability. For some combinations of parameters, these processes may display a periodic behavior with a period on the order of the cutoff time. Over time, each realization starts to behave differently due to some random events yielding different outcomes. Consequently, the periodic oscillations will dephase, and they will cancel out after averaging over all realizations. In the example from Figure 4.8, we find that both $\mathbb{E}\big[v_i(t)\big]$ and $\mathbb{E}\big[k_i(t)\big]$ are periodic with period approximately t_{cut}. The amplitude of the oscillations vanishes after a few periods.

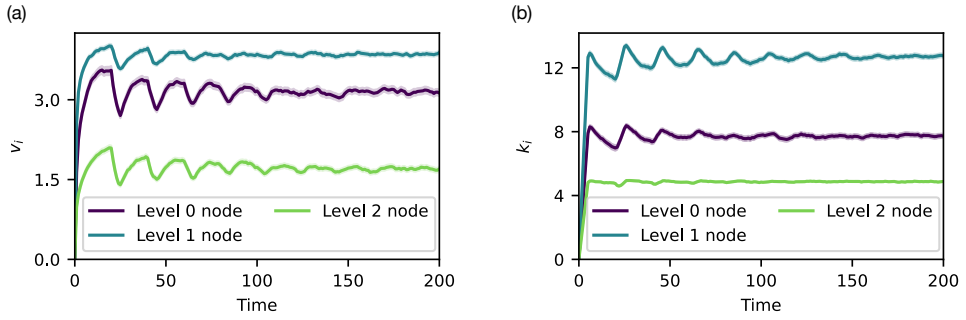

Figure 4.8: **A transient state with periodic oscillations may exist in quasi-deterministic systems.** Evolution of v_i and k_i in a quantum network with a (2,3)-tree topology running the SRS protocol described in the main text. Each line (purple, blue, and green) corresponds to a node in a different level of the tree (level 0, 1, and 2). The error for each solid line is shown as a shaded region, although it is hard to notice since its maximum value is 0.040 in (a) and 0.084 in (b) – the error is defined as $2\hat{\sigma}/(N_{\text{samples}})^{0.5}$, where $\hat{\sigma}$ is the sample standard deviation and N_{samples} is the number of samples. Other parameters used in this experiment: $p_{\text{gen}} = 0.99$, $F_{\text{new}} = 0.88$, $p_{\text{swap}} = 1$, $r = 5$, $T = 2000$ time steps, $M = 4$, $p_{\text{cons}} = 0.01$, $q = 0.2$, $F_{\text{app}} = 0.6$, $t_{\text{cut}} = 20$ time steps. Numerical results obtained using a network simulation and Monte Carlo sampling with 10^3 samples.

4.8. [APPENDIX] ANALYTICAL PERFORMANCE METRICS IN THE ABSENCE OF SWAPS

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

$$v_i \equiv \lim_{t\to\infty} \mathbb{E}\big[v_i(t)\big] = d_i \frac{1 - \frac{1-p_{\text{cons}}}{1-p_{\text{gen}}}\lambda^r}{1 - \frac{p_{\text{cons}}}{p_{\text{gen}}}\lambda^r} \tag{4.17}$$

and

$$k_i \equiv \lim_{t\to\infty} \mathbb{E}\big[k_i(t)\big] = d_i\, p_{\text{gen}} \frac{r + \frac{p_{\text{cons}}(1-p_{\text{cons}})}{p_{\text{gen}}-p_{\text{cons}}}(\lambda^r - 1)}{p_{\text{gen}} - p_{\text{cons}}\lambda^r}, \tag{4.18}$$

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

We define w_{ij} as the number of entangled links shared between nodes i and j (similar to the definition of κ in Appendix 4.7). In the absence of swaps, nodes i and j can only share entangled links if they are physical neighbors, since the only mechanism available is heralded entanglement generation. If i and j are not physical neighbors, then $w_{ij} = 0$. The entangled links shared between nodes i and j can be consumed in some application

or discarded when applying cutoffs. However, we also assume that entangled links are always consumed before they reach the cutoff time, i.e., $t_{\text{cut}} \gg \frac{1}{p_{\text{cons}}}$ (cutoff measured in number of time steps). This assumption allows us to model w_{ij} using the general random walk shown in Figure 4.9:

- The maximum value for w_{ij} is the number of qubits available per physical link, r. This state is reachable even when entanglement generation is done sequentially, since links can be stored for longer than r time steps (we assume $t_{\text{cut}} > r$).
- The probabilities of transition forward are $p_k = p_{\text{gen}}(1 - p_{\text{cons}})$, $\forall k < r$, and $p_r = 0$.
- The probabilities of transition backwards are $q_0 = 0$, $q_k = p_{\text{cons}}(1 - p_{\text{gen}})$, $\forall 0 < k < r$, and $q_r = p_{\text{cons}}$.
- The no-transition probability is $z_k = 1 - p_k - q_k$, $\forall k$.

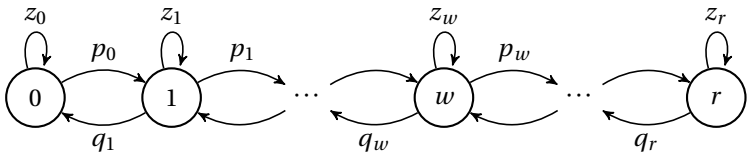

Figure 4.9: **General random walk modeling the number of entangled links w_{ij} between nodes i and j in the absence of swaps.**

The steady-state probability distribution of this Markov chain is given by [184]

$$\lim_{t\to\infty} \Pr\big[w_{ij}(t) = w \mid A_{ij} = 1\big] = \begin{cases} \left(1 + \sum_{k=1}^{r} \prod_{m=0}^{k-1} \frac{p_m}{q_{m+1}}\right)^{-1}, & w = 0 \\ \left(1 + \sum_{k=1}^{r} \prod_{m=0}^{k-1} \frac{p_m}{q_{m+1}}\right)^{-1} \prod_{m=0}^{w-1} \frac{p_m}{q_{m+1}}, & w > 0 \end{cases}, \tag{4.19}$$

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

$$\lim_{t\to\infty} \Pr\big[w_{ij}(t) = w \mid A_{ij} = 1\big] = \begin{cases} \pi_0, & w = 0 \\ \pi_0 \rho^w, & 0 < w < r \\ \pi_0 \rho^r (1 - p_{\text{gen}}), & w = r \end{cases}, \tag{4.20}$$

where

$$\pi_0 \equiv \frac{p_{\text{gen}} - p_{\text{cons}}}{(1 - p_{\text{gen}})(p_{\text{gen}}\rho^r - p_{\text{cons}})} \quad \text{and} \quad \rho \equiv \frac{p_{\text{gen}}(1 - p_{\text{cons}})}{p_{\text{cons}}(1 - p_{\text{gen}})}. \tag{4.21}$$

The expected value of w_{ij} is

$$\begin{aligned}
\lim_{t\to\infty}\mathbb{E}\big[w_{ij}(t)\mid A_{ij}=1\big] &= \lim_{t\to\infty}\sum_{w=0}^{r} w\cdot\Pr\big[w_{ij}(t)=w\mid A_{ij}=1\big]\\
&= \sum_{w=0}^{r} w\cdot\lim_{t\to\infty}\Pr\big[w_{ij}(t)=w\mid A_{ij}=1\big]\\
&\overset{a}{=}\pi_0\sum_{w=0}^{r-1} w\rho^w + r\pi_0\rho^r(1-p_{\text{gen}})\\
&=\pi_0\frac{\rho - r\rho^r + (r-1)\rho^{r+1}}{(1-\rho)^2} + r\pi_0\rho^r(1-p_{\text{gen}})\\
&=\frac{p_{\text{gen}}}{(p_{\text{gen}}-p_{\text{cons}})(p_{\text{gen}}\rho^r - p_{\text{cons}})}\\
&\quad\cdot\Big(r(p_{\text{gen}}-p_{\text{cons}})\rho^r + p_{\text{cons}}(1-p_{\text{cons}})(1-\rho^r)\Big),
\end{aligned} \tag{4.22}$$

where we have used (4.20) in step a.

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

$$\begin{aligned}
v_i \equiv \lim_{t\to\infty}\mathbb{E}\big[v_i(t)\big] &= \lim_{t\to\infty}\mathbb{E}\Big[\sum_{j=1}^{n}\min(w_{ij}(t),1)\Big] = \sum_{j=1}^{n}\lim_{t\to\infty}\mathbb{E}\big[\min(w_{ij}(t),1)\big]\\
&\overset{a}{=}\sum_{j=1}^{n}\lim_{t\to\infty}\sum_{x=0}^{r} x\cdot\Pr\big[\min(w_{ij}(t),1)=x\big]\\
&=\sum_{j=1}^{n}\sum_{x=0}^{r} x\cdot\lim_{t\to\infty}\Pr\big[\min(w_{ij}(t),1)=x\big]\\
&\overset{b}{=}\sum_{j=1}^{n}\lim_{t\to\infty}\Pr\big[\min(w_{ij}(t),1)=1\big] = \sum_{j=1}^{n}\lim_{t\to\infty}\Pr\big[w_{ij}(t)>0\big]\\
&\overset{c}{=}\sum_{j=1}^{n}\lim_{t\to\infty}\Pr(A_{ij}=1)\cdot\Pr\big[w_{ij}(t)>0\mid A_{ij}=1\big]\\
&\overset{d}{=}\sum_{j=1}^{n} A_{ij}\lim_{t\to\infty}\Big(1-\Pr\big[w_{ij}(t)=0\mid A_{ij}=1\big]\Big)\\
&\overset{e}{=}\sum_{j=1}^{n} A_{ij}(1-\pi_0)\\
&\overset{f}{=} d_i(1-\pi_0)\\
&\overset{g}{=} d_i\frac{p_{\text{gen}}^{r+1}(1-p_{\text{cons}})^r - p_{\text{gen}}(1-p_{\text{gen}})^{r-1}p_{\text{cons}}^r(1-p_{\text{cons}})}{p_{\text{gen}}^{r+1}(1-p_{\text{cons}})^r - (1-p_{\text{gen}})^r p_{\text{cons}}^{r+1}}\\
&= d_i\frac{1-\frac{1-p_{\text{cons}}}{1-p_{\text{gen}}}\lambda^r}{1-\frac{p_{\text{cons}}}{p_{\text{gen}}}\lambda^r},
\end{aligned} \tag{4.23}$$

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

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

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

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

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

e. We use (4.20).

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

g. We use (4.21).

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

$$
\begin{aligned}
k_i \equiv \lim_{t\to\infty} \mathbb{E}\big[k_i(t)\big] = \lim_{t\to\infty} \mathbb{E}\Big[\sum_{j=1}^{n} w_{ij}(t)\Big] &= \lim_{t\to\infty} \sum_{j=1}^{n} \mathbb{E}\big[w_{ij}(t)\big] \\
&\overset{a}{=} \lim_{t\to\infty} \sum_{j=1}^{n} A_{ij} \cdot \mathbb{E}\big[w_{ij}(t) \mid A_{ij} = 1\big] \\
&\overset{b}{=} \lim_{t\to\infty} \mathbb{E}\big[w_{ij}(t) \mid A_{ij} = 1\big] \cdot \sum_{j=1}^{n} A_{ij} \\
&\overset{c}{=} d_i \cdot \lim_{t\to\infty} \mathbb{E}\big[w_{ij}(t) \mid A_{ij} = 1\big] \\
&\overset{d}{=} d_i\, p_{\text{gen}} \frac{r + \frac{p_{\text{cons}}(1-p_{\text{cons}})}{p_{\text{gen}}-p_{\text{cons}}}(\lambda^r - 1)}{p_{\text{gen}} - p_{\text{cons}}\lambda^r},
\end{aligned}
\tag{4.24}
$$

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

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

b. In a homogeneous network with no swaps, w_{ij} depends on A_{ij} but is otherwise independent of the nodes i and j. Hence, $\mathbb{E}\big[w_{ij}(t) \mid A_{ij} = 1\big]$ does not depend on j. This can also be seen in (4.22).

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

d. We use (4.22) and rearrange terms.

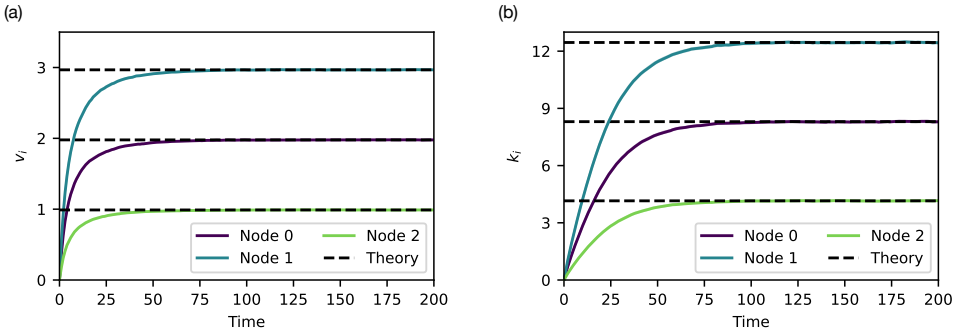

Figure 4.10: **The expected virtual neighborhood size and the expected virtual node degree converge to the steady-state analytical values in the absence of swaps.** In this example, we ran the SRS protocol (Algorithm 1 from the main text) with $q = 0$ (i.e., no swaps) on quantum network with (2,3)-tree topology. Nodes 0, 1, and 2 correspond to nodes in levels 0, 1, and 2 of the tree, respectively (i.e., they have physical node degrees $d_0 = 2$, $d_1 = 3$, and $d_2 = 1$, respectively). Each solid line corresponds to each of the three nodes. The dashed lines correspond to the expected steady-state values predicted by Equations (4.23) and (4.24) for each of the nodes. The standard error for each solid line is shown as a shaded region, although it is hard to notice since its maximum value is 0.017 in (a) and 0.056 in (b). Other parameters used in this experiment are $p_{\text{gen}} = 0.2$, $F_{\text{new}} = 0.9$, $r = 5$, $T = 2000$ time steps, $p_{\text{cons}} = 0.1$, $F_{\text{app}} = 0.6$, $t_{\text{cut}} = 221$ time steps (given by (4.3)). Numerical results obtained using a network simulation and Monte Carlo sampling with 10^4 samples.

(4.23) and (4.24) can be used to study the performance of the protocol in the limit of large number of resources ($r \to \infty$). When $p_{\text{gen}} > p_{\text{cons}}$ we find

$$\lim_{r\to\infty} v_i = d_i, \quad \text{and} \quad \lim_{r\to\infty} k_i = \lim_{r\to\infty} r d_i = \infty. \tag{4.25}$$

This means that, when the generation rate exceeds the consumption rate, the virtual neighborhood size will eventually saturate and every node will share entanglement with every physical neighbor. In particular, the average number of entangled links will increase infinitely (for large but finite r, k_i reaches a maximum value of $\sim r d_i$). When $p_{\text{gen}} < p_{\text{cons}}$,

$$\lim_{r\to\infty} v_i = d_i \frac{p_{\text{gen}}(1-p_{\text{cons}})}{p_{\text{cons}}(1-p_{\text{gen}})}, \quad \text{and} \quad \lim_{r\to\infty} k_i = d_i\, p_{\text{gen}} \frac{1-p_{\text{cons}}}{p_{\text{cons}}-p_{\text{gen}}}. \tag{4.26}$$

In Figure 4.4 from the main text, we plot the expected virtual neighborhood size and expected virtual degree for $r = 5$, focusing on the interplay between p_{gen} and p_{cons}. Both quantities decrease with increasing consumption rate, as one would expect, and quickly drop to zero for $p_{\text{cons}} > p_{\text{gen}}$.

Lastly, Figure 4.10 shows an example of the convergence of $\mathbb{E}\big[v_i(t)\big]$ and $\mathbb{E}\big[k_i(t)\big]$ to v_i and k_i over time, respectively. The time-dependent quantities have been calculated using a simulation on a quantum network with a (2,3)-tree physical topology. The dashed lines correspond to the steady-state values in the absence of swaps predicted by (4.23) and (4.24).

4.9. [APPENDIX] STEADY STATE OF A STOCHASTIC PROCESS

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

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

Algorithm 2 - Steady state estimation.

Inputs:

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

Outputs:

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

Algorithm:

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

Theorem 4.1. *Let $X(t) \in [a,b]$, with $a,b \in \mathbb{R}$, be a stochastic process with constant steady-state mean, i.e., $\lim_{t\to\infty}\mathbb{E}\big[X(t)\big] = X_\infty < \infty$. Let $\overline{X}_N(t_k)$ be a sample mean over N samples at time $t_k \in \{t_0, t_1, ..., t_{M-1}\}$, with $t_0 < t_1 < ... < t_{M-1}$. Consider a minimum size of the steady-state window w. When $N \to \infty$, Algorithm 2 with inputs $\overline{X}_N(t_k)$, a, b, and w, finds α such that*

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

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

4

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

$$\overline{X}_N(t) = \frac{1}{N}\sum_{n=0}^{N-1} x_n(t). \tag{4.27}$$

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

$$\Pr\left[\mathbb{E}\big[X(t)\big] \in \left(\overline{X}_N(t) - \frac{2\sigma(t)}{\sqrt{N}},\ \overline{X}_N(t) + \frac{2\sigma(t)}{\sqrt{N}}\right)\right] > 0.95. \tag{4.28}$$

The values of $X(t)$ are constrained to the interval $[a,b]$, and therefore the standard deviation is upper bounded by [17]

$$\sigma(t) \leq (b-a)/2. \tag{4.29}$$

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

$$\mathrm{IC}_i = \Big(\overline{X}_N(t_i) - \varepsilon,\ \overline{X}_N(t_i) + \varepsilon\Big) \tag{4.30}$$

Using (4.28), (4.29), and (4.30), we can write

$$\Pr\Big[\mathbb{E}\big[X(t_i)\big] \in \mathrm{IC}_i\Big] > 0.95. \tag{4.31}$$

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

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

$$\mathrm{IC}_{ij} = \Big(\ \max\big(\overline{X}_N(t_i), \overline{X}_N(t_j)\big) - \varepsilon,\ \min\big(\overline{X}_N(t_i), \overline{X}_N(t_j)\big) + \varepsilon\ \Big). \tag{4.32}$$

The size of this interval of confidence is

$$\Delta_{ij} = 2\varepsilon - \left|\overline{X}_N(t_i) - \overline{X}_N(t_j)\right|. \tag{4.33}$$

We provide a graphical intuition in Figure 4.11.

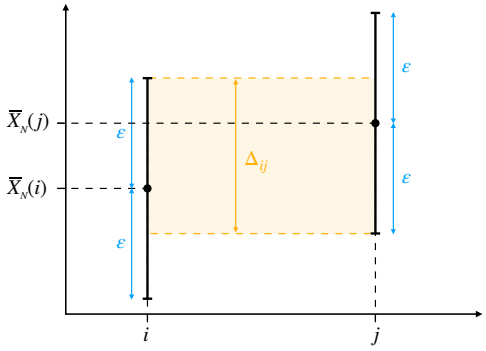

Figure 4.11: **Graphical intuition for the interval of confidence ij used to identify the steady state.** $\overline{X}_N(i)$ corresponds to the sample mean at i, ε is the error, and Δ_{ij} is the size of the interval of confidence ij (highlighted in yellow).

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

$$\Pr\Big[\ \mathbb{E}\big[X(t_i)\big] \in \mathrm{IC}_{ij}\ \Big] \geq 0.815,\ \forall i, j \geq \alpha, \tag{4.34}$$

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

$$\begin{aligned}
\Pr\Big[\ \mathbb{E}\big[X(t_i)\big] \in \mathrm{IC}_{ij}\ \Big] &\overset{a}{=} \Pr\Big[\ \mathbb{E}\big[X(t_i)\big] \in \Big(\overline{X}_N(t_i) - \varepsilon,\ \overline{X}_N(t_i) + \Delta_{ij} - \varepsilon\Big)\ \Big] \\
&\overset{b}{=} \int_{\overline{X}_N(t_i)-\varepsilon}^{\overline{X}_N(t_i)} f_N(x_i)\mathrm{d}x_i + \int_{\overline{X}_N(t_i)}^{\overline{X}_N(t_i)+\Delta_{ij}-\varepsilon} f_N(x_i)\mathrm{d}x_i \\
&\overset{c}{\geq} \int_{\overline{X}_N(t_i)-\frac{2\sigma(t)}{\sqrt{N}}}^{\overline{X}_N(t_i)} f_N(x_i)\mathrm{d}x_i + \int_{\overline{X}_N(t_i)}^{\overline{X}_N(t_i)+\Delta_{ij}-\varepsilon} f_N(x_i)\mathrm{d}x_i \\
&\overset{d}{\geq} \frac{0.95}{2} + \int_{\overline{X}_N(t_i)}^{\overline{X}_N(t_i)+\Delta_{ij}-\varepsilon} f_N(x_i)\mathrm{d}x_i \\
&\overset{e}{\geq} 0.475 + \int_{\overline{X}_N(t_i)}^{\overline{X}_N(t_i)+\frac{\varepsilon}{2}} f_N(x_i)\mathrm{d}x_i \\
&\overset{f}{\geq} 0.475 + \int_{\overline{X}_N(t_i)}^{\overline{X}_N(t_i)+\frac{\sigma}{\sqrt{N}}} f_N(x_i)\mathrm{d}x_i \\
&\overset{g}{\geq} 0.475 + \frac{0.68}{2} \\
&= 0.815
\end{aligned} \tag{4.35}$$

with the following steps:

a. Without loss of generality, assume $\overline{X}_N(t_i) \geq \overline{X}_N(t_j)$.

b. Let $f_N(x_i)$ be the probability distribution function of $\mathbb{E}\big[X(t_i)\big]$. As previously shown, when N goes to infinity, this distribution converges to $\mathcal{N}\big(\overline{X}_N(t_i), \sigma(t)^2/N\big)$. We assume N is sufficiently large.

c. Using (4.29): $\varepsilon = \frac{b-a}{\sqrt{N}} \geq \frac{2\sigma(t)}{\sqrt{N}}$.

d. The probability that a normally distributed random variable takes a value between the mean and two standard deviations away is larger than $\frac{0.95}{2}$, i.e., $\int_{\mu-2\sigma}^{\mu} f(z)dz = \int_{\mu}^{\mu+2\sigma} f(z)dz \geq \frac{0.95}{2}$, where $f(z)$ is the probability distribution of $Z \sim \mathcal{N}(\mu, \sigma^2)$.

e. The algorithm only considers i and j such that $\Delta_{ij} \geq \frac{3}{2}\varepsilon$.

f. Using (4.29) again: $\varepsilon \geq \frac{2\sigma(t)}{\sqrt{N}}$.

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

□

Note that the validity of this method depends on the number of samples N, which must be sufficiently large in order to apply the Central Limit Theorem.

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

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

We considered employing other data analysis techniques, such as bootstrapping and data blocking [178], to improve our estimates. However, we decided to not use them since (i) bootstrapping would require running the simulations over many more time steps to be able to take many samples spaced an autocorrelation time; and (ii) data blocking requires a much larger storage space.

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

Figure 4.12 shows an example of our algorithm finding the steady state of the virtual neighborhood size when running the SRS protocol in a network with a tree topology. The virtual neighborhood size of three nodes is shown in different colors. Dots correspond to the time t_α at which the algorithm declares that the steady state has been reached.

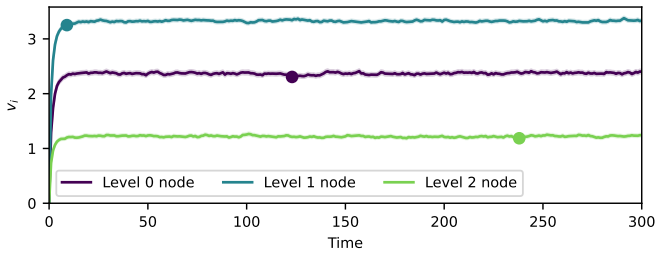

Figure 4.12: **Algorithm 2 can identify the steady state of a stochastic process.** Evolution of the average virtual neighborhood size in a quantum network with a (2,3)-tree topology running the SRS protocol described in the main text. Each line (purple, blue, and green) corresponds to a node in a different level of the tree (level 0, 1, and 2). Dots indicate that the steady state has been reached, according to Algorithm 2. The error for each solid line is shown as a shaded region, although it is hard to notice since its maximum value is 0.029 (the error is defined as $2\hat{\sigma}/N_{\text{samples}}$, where $\hat{\sigma}$ is the sample standard deviation and N_{samples} is the number of samples). Other parameters used in this experiment: $p_{\text{gen}} = 0.9$, $F_{\text{new}} = 0.88$, $p_{\text{swap}} = 1$, $r = 5$, $T = 2000$ time steps, $M = 4$, $p_{\text{cons}} = 0.225$, $q = 0.1$, $F_{\text{app}} = 0.6$, $t_{\text{cut}} = 56$ time steps. Numerical results obtained using a network simulation and Monte Carlo sampling with 10^3 samples. The simulation was run over 560 time steps (only the first 300 are shown here) and the steady-state window was 112 time steps.

4.10. [APPENDIX] EXTRA EXPERIMENTS ON A TREE NETWORK

Here, we provide more examples of the dependence of the virtual neighborhood size, v_i, and the virtual node degree, k_i, on the SRS protocol parameter q (probability that a node performs a swap). In the main text, we discuss the dependence on q using a network with the following baseline set of parameters: (2,3)-tree topology, $p_{\text{gen}} = 0.9$, $F_{\text{new}} = 0.888$, $p_{\text{swap}} = 1$, $r = 5$, $T = 2000$ time steps, $M = 4$, $p_{\text{cons}} = 0.225$, $F_{\text{app}} = 0.6$, $t_{\text{cut}} = 56$ time steps. Figure 4.13 shows similar plots for networks with slightly different combinations of parameters that correspond to larger trees, smaller consumption rate, and probabilistic swapping. In all situations we observe the same qualitative behavior as in the baseline case: the value of q that maximizes the virtual neighborhood size is node-dependent, and k_i is monotonically decreasing with increasing q.

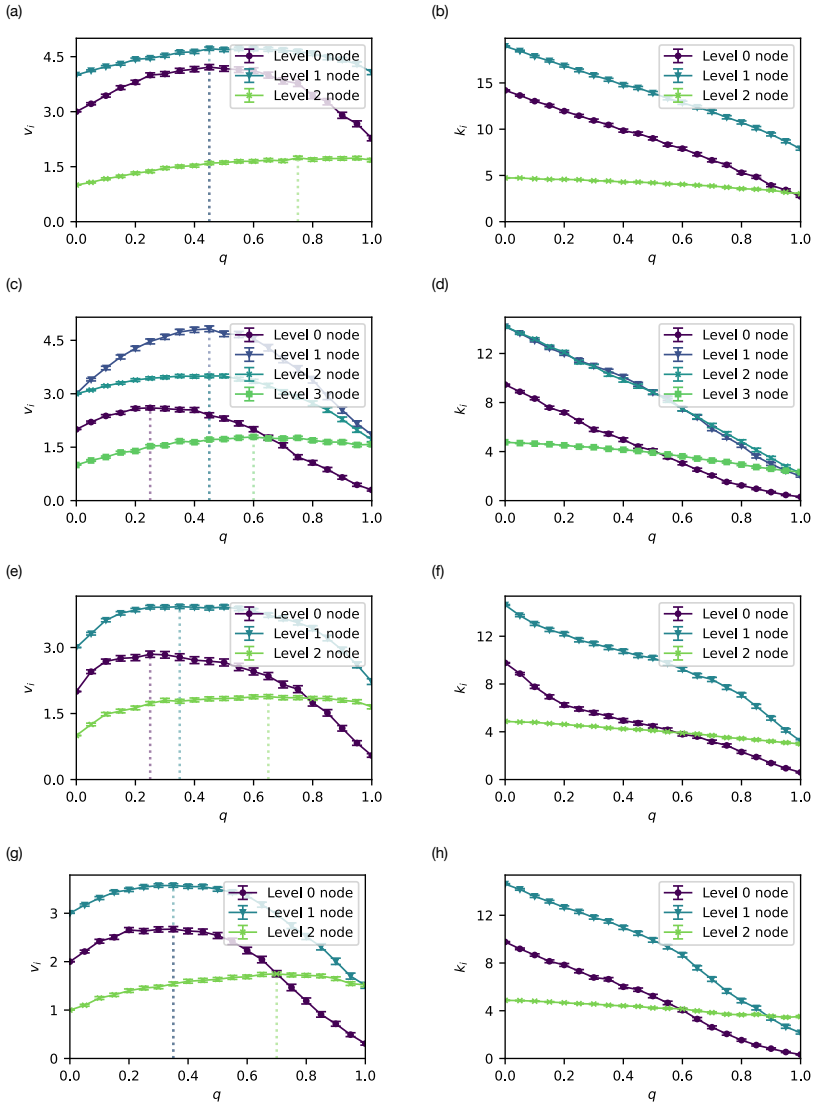

Figure 4.13: **Our performance metrics show the same qualitative behavior for different combinations of parameters.** Expected virtual neighborhood size (a, c, e, g) and virtual node degree (b, d, f, h) in the steady state in a tree network running the SRS protocol vs the protocol parameter q. The value of q that maximizes the virtual neighborhood size is indicated by the dotted lines. Baseline parameters: (2,3)-tree topology, $p_{\text{gen}} = 0.9$, $F_{\text{new}} = 0.888$, $p_{\text{swap}} = 1$, $r = 5$, $T = 2000$ time steps, $M = 4$, $p_{\text{cons}} = 0.225$, $F_{\text{app}} = 0.6$, $t_{\text{cut}} = 56$ time steps. The subfigures in each row correspond to a different experiment: (a,b) (3,3)-tree, (c,d) (2,4)-tree, (e,f) $p_{\text{cons}} = 0.1$, and (g,h) $p_{\text{cons}} = 0.1$ and $p_{\text{swap}} = 0.5$. Results obtained using a network simulation and Monte Carlo sampling with 10^3 samples. The error in the error bars is defined as $2\hat{\sigma}/N_{\text{samples}}$, where $\hat{\sigma}$ is the sample standard deviation and N_{samples} is the number of samples.

DATA AND CODE AVAILABILITY

The data shown in this chapter can be found in ref. [93]. Our code can be found in the following GitHub repository: https://github.com/AlvaroGI/optimizing-cd-protocols.

AUTHOR CONTRIBUTIONS

ÁGI conceived and defined the project. ÁGI conducted the analysis and wrote the paper, which was published as ref. [94]. SW provided active feedback at every stage of the project.

5

The Viability of Preemptive Delivery of Quantum Resources

Álvaro G. Iñesta and Stephanie Wehner

The inspection paradox is
a common source of confusion,
an occasional cause of error,
and an opportunity for clever experimental design.
— Allen Downey

Quantum network applications often rely on the use of specific quantum states as consumable resources, such as entangled states shared among remote parties. Ensuring the reliable and efficient delivery of these resource states is critical for the success of such applications. Here, we analyze the performance of two strategies for the delivery of quantum resource states: on-demand protocols, where resource delivery is triggered upon request, and continuous-delivery protocols, where resources are preemptively delivered without prior knowledge of when consumption requests will arrive. To evaluate these protocols, we use two key performance metrics: the expected completion time, defined as the time between a request arrival and the successful delivery of a resource state, and the wastage, which measures the number of delivered resources that go unused. Continuous-delivery protocols can significantly reduce completion times at the cost of increased resource wastage – when resources are delivered in advance, they may have to be discarded upon delivery due to insufficient storage capacity, or after some time due to decoherence. We derive analytical

expressions and approximations to evaluate the performance metrics for both types of protocol. Surprisingly, we find that, in certain scenarios, preemptive delivery of resource states can paradoxically increase the expected completion time. This phenomenon occurs when the time required to distribute a single resource state follows a broad probability distribution. Our findings offer insights into the efficient delivery of quantum resource states and identify the conditions under which preemptive delivery is beneficial.

5.1. INTRODUCTION

Quantum network applications commonly rely on the use of specific quantum states as consumable resources, with entangled states being a particularly prevalent type. For example, in the verifiable quantum secret sharing from refs. [46, 120], the dealer, who wants to distribute a secret quantum state among a number of parties, must also distribute multiple ancillary entangled states that are later consumed to verify the correct distribution of the secret.

The delivery[1] of quantum resource states among remote parties is typically a slow and stochastic process [67, 87, 110, 118, 147]. As an example, consider the state-of-the-art experiment from ref. [147], where three physically distant qubits, realized with nitrogen vacancy centers, are entangled to produce a tripartite GHZ state. In this work, the lifetime of the generated entangled state (approximately 11.6 ms) is of the same order of magnitude as the average time required to deliver the GHZ state (over 11.8 ms). This can negatively impact the performance of quantum network applications that are waiting to consume the resource states. For instance, consider an application in which entanglement is needed to teleport some quantum data to a remote location with high quality [14]. If the quantum data is stored in memory for a long and unknown period of time while waiting for entanglement to be delivered, its quality will degrade due to decoherence [38, 58], potentially compromising the success of the application. In such cases, ensuring rapid delivery of entanglement is crucial to enable the reliable execution of the application.

Quantum resource states, including entangled states, can be distributed among remote parties using a *delivery subroutine*, which takes time D to deliver the resource state after being triggered. We consider two primary strategies to trigger this subroutine: on-demand and continuous delivery [35]. In on-demand (OD) protocols, the delivery subroutine is initiated only after users explicitly request the resource, which is immediately consumed upon delivery. In the context of bipartite entanglement delivery, such strategies are sometimes referred to as "generate-when-requested". OD protocols have been extensively studied in multiple contexts, including end-to-end entanglement delivery in quantum repeater chains [22, 44, 91, 167], entanglement distribution in star networks utilizing a quantum switch [65], and entanglement routing within larger networks [140, 142, 187, 188]. In continuous-delivery (CD) protocols, the subroutine is triggered repeatedly, without the need for users to make explicit requests. This approach ensures that resource states are continuously delivered, allowing users to consume them whenever necessary.

[1] In this chapter, we use the term *delivery of quantum states* rather than *distribution* (as we do in other chapters) to enhance clarity, as the latter term will be frequently used to refer to probability distributions.

CD protocols have been explored for bipartite entanglement delivery across arbitrary networks [94, 177], and for pre-entangling specific sets of network nodes to facilitate subsequent on-demand delivery [66, 149]. Additionally, adaptive CD protocols, which determine where in the network entanglement should be delivered based on past events, have been proposed as an efficient alternative to naive CD approaches [105]. Note that OD and CD protocols may be ill-defined in certain scenarios. For example, in refs. [179, 186], users can request a steady delivery of resource states at a fixed rate. In this case, an OD protocol may need to trigger the delivery subroutine continuously, temporarily functioning as a CD protocol.

One would expect CD protocols to provide faster service than OD protocols, since they start delivering resources preemptively, before any request arrives. However, if quantum states are delivered, stored and left unused for an extended period, they will undergo decoherence [38, 58], eventually rendering them unusable and leading to resource wastage. In ref. [35], OD and CD protocols are compared in the context of entanglement routing, where the goal is to deliver bipartite entanglement between two users in a network. The study concluded that pre-distributing entanglement with a CD protocol can outperform OD protocols in terms of average latency (defined as the time needed to deliver bipartite entanglement between two specific pairs of users), particularly when there is a small number of requests. Here, we compare the performance of OD and CD protocols in a more abstract setting. We assume the objective is to deliver quantum resource states to an arbitrary but fixed number of parties. The entire delivery process is characterized by the probability distribution of the delivery time D, which encapsulates all network properties and constraints. Our central goal is to answer the following general question: *should we deliver quantum resources preemptively, even if this leads to resource wastage?*

We first define relevant performance metrics to compare both types of protocol. Namely, we use the expected request completion time (time since a request arrives until a resource state is provided, see Definition 5.1) and the wastage (number of wasted resource states per request, see Definition 5.2). We provide analytical expressions to compute these quantities. Then, we study a setting in which users cannot store delivered resources for later use – that is, resources must be consumed or discarded upon delivery. Our main findings are the following:

- When the delivery time D follows a broad distribution, preemptive resource delivery, as employed in CD protocols, paradoxically leads to longer request completion times compared to OD protocols. We show that this occurs when the standard deviation of D exceeds its expected value. We also provide an intuitive example to understand this behavior.

- We show that the number of wasted resource states per completed request in CD protocols is proportional to the expected time between consumption requests and inversely proportional to the expected value of D, while being zero for OD protocols.

In Section 5.2, we introduce the parameters of the problem and the model we use for OD and CD protocols. In Section 5.3, we formally define our performance metrics, and we derive analytical expressions to compute them. Then, in Section 5.4, we study a network in which the users cannot store any resource states, i.e., resources must be immediately

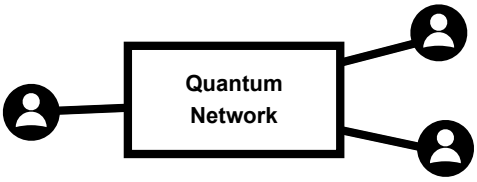

Figure 5.1: **We model the quantum network as a black box.** The network can trigger a delivery subroutine that takes time D to deliver a resource state to the users.

consumed or discarded upon delivery. Lastly, we discuss the main insights and future research directions in Section 5.5.

5.2. MODEL FOR QUANTUM RESOURCE STATE DELIVERY

Here, we consider a quantum network whose goal is to deliver resource states to a set of users, which they need in order to run an application. The resource state is generally an entangled state. For example, bipartite entanglement may be requested to perform entanglement-based quantum key distribution [59], or a cluster state may be needed for a measurement-based quantum computation [23, 155]. However, the resource state may also consist of multiple copies of an entangled state, i.e., it can be a tensor product of multiple entangled states. In our model, the number of users and the specific resource state being generated are irrelevant, as long as they are always the same: we assume that **the same set of users will always ask for the same resource state**. In fact, we model the whole network as a black box that can run a subroutine to deliver the resource state, as illustrated in Figure 5.1.

After being triggered, **the delivery subroutine takes time D to deliver the resource state**. The subroutine is likely to involve multiple stochastic processes and therefore D is a random variable – for example, end-to-end entanglement generation over a two-way repeater chain requires successful entanglement generation attempts among each pair of adjacent repeaters [22]. The probability distribution of D depends on multiple factors, such as the physical properties of the network and the specific delivery subroutine employed. We assume the network operates in some steady state where the probability distribution of D is arbitrary but remains fixed over time, a common assumption in the literature – characterizing this probability distribution for different quantum networking systems has been a frequent topic of research (see, e.g., [44, 96, 103, 167]).

Every time the users need to consume a resource state, they must submit a **consumption request** – when a request is submitted, we say that the request has *arrived* in the system. We assume that the times between request arrivals are independent and identically distributed (i.i.d.), and we represent them with R, a random variable following an arbitrary probability distribution. If a request arrives and a resource state is already available, the state is consumed immediately to meet the request. If a request arrives when no resources are available, the request is put in a first-in first-out queue. Whenever a resource state is delivered, it is assigned to the first request in the queue and the request is marked as completed.

When the rate at which requests arrive exceeds the rate at which resources can be delivered, the system becomes unstable and the number of requests waiting in the queue goes to infinity over time. To prevent this undesirable behavior, we assume the following:

Assumption 5.1. *On average, resource states are delivered at a rate exceeding the arrival rate of consumption requests, i.e.,*

$$\mathbb{E}[D] < \mathbb{E}[R]. \tag{5.1}$$

For further details about the relationship between condition (5.1) and the stability of the system, see, e.g., Chapter 14 from ref. [184].

As introduced earlier, the delivery subroutine can be triggered when a request arrives or in advance. This leads to two general types of protocols [35], which we define as follows:

- In an *on-demand delivery (OD)* protocol, the arrival of a request triggers the subroutine responsible for resource state delivery. While the queue is empty, the network remains idle and the subroutine is not executed. However, when requests are present in the queue, the subroutine is called sequentially, processing each request until the queue is cleared. This is illustrated in Figure 5.2a.

- In contrast, a *continuous-delivery (CD)* protocol proactively triggers the subroutine in advance, aiming to provide faster service. Unlike the on-demand approach, the CD protocol may continue generating resource states even when the requests queue is empty. Specifically, we assume that the delivery subroutine is triggered immediately after the completion of the previous subroutine execution, ensuring a steady flow of resource state delivery. This is illustrated in Figure 5.2b.

Alternative strategies could blend these two philosophies, triggering the delivery subroutine in advance but not continuously, potentially guided by live network traffic information. While these approaches might improve protocol performance, our work focuses on a preliminary analysis of fundamental strategies, leaving the exploration of such advanced methods for future research.

CD protocols preemptively trigger the delivery subroutine, and thus are generally expected to provide resource states for incoming requests more quickly than OD protocols. This is depicted in the example from Figure 5.2. There, requests arrive at the same time in both cases. However, under the CD protocol, the delivery subroutine is already underway when the requests arrive, resulting in requests 1 and 2 being completed faster than in the OD protocol. Later, we will demonstrate that this intuition does not always hold true: CD protocols can, in fact, result in longer completion times, depending on the probability distribution of D.

Despite the potential advantage in completion times, CD protocols can become inefficient if resources are delivered significantly faster than requests arrive: unused resource states may accumulate in memory and eventually must be discarded, as their quality degrades over time due to decoherence. In the next section, we propose two performance metrics to measure the rate at which requests are completed and also the amount of resource wastage. These metrics are then used to evaluate the benefits and drawbacks of preemptively distributing quantum resource states.

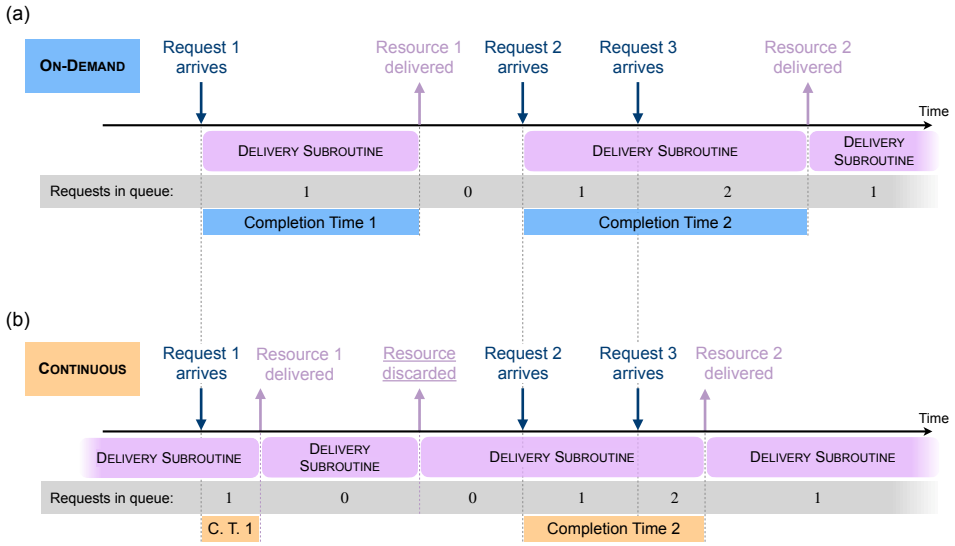

Figure 5.2: **Illustration of OD and CD protocol operations over time.** (a) In an OD protocol, the delivery subroutine is triggered only when requests are present in the queue, ensuring resource-efficient delivery. (b) In a CD protocol, the delivery subroutine is triggered continuously. This can result in resource wastage: in this example, we assume the users lack memory to store quantum states, and therefore the second resource delivered has to be discarded. Here, the CD protocol completes requests 1 and 2 faster than the OD protocol.

5.3. PERFORMANCE METRICS

In our work, the primary goal of resource delivery protocols is to deliver resources and fulfill requests as quickly as possible. To evaluate performance, we first propose the expected completion time as the main metric. Additionally, we introduce a second quantity to assess the efficiency of the protocol by measuring the amount of wasted resources.

5.3.1. EXPECTED COMPLETION TIME

We define the *completion time* of the i-th request, denoted as T_i, as the time passed since the arrival of the request until the delivery of a resource state that fulfills the request. The completion time of the i-th request depends on the number of requests already in the queue at the time of its arrival, as these requests must be completed before the i-th one can be processed, and also on the delivery time of the subroutine. Both the queue size and the delivery time are random variables. Consequently, T_i is a random variable. To evaluate the performance of resource delivery protocols, we focus on the expected completion time as the primary metric. Specifically, we consider the expected completion time in the steady state[2]:

[2]The completion times form a stochastic process $(T_i, i \in \mathbb{N})$, and these random variables are not i.i.d. in general. That is, $\mathbb{E}[T_i]$ is not necessarily equal to $\mathbb{E}[T_j]$, $i \neq j$. However, in the long run, $\mathbb{E}[T_i]$ typically converges to a fixed value. There are exceptions in certain pathological cases, where the expected completion time may

Definition 5.1. *The* expected completion time *is defined as*

$$\mathbb{E}[T] = \lim_{i \to \infty} \mathbb{E}[T_i], \tag{5.2}$$

where T_i is the arrival time of the i-th request.

Equivalent performance metrics have been defined in related works. Examples include the expected rate at which resources are delivered (equivalent to $\mathbb{E}[T]$ when all resources are immediately consumed upon delivery) [35, 140, 142, 187]; the average latency (inverse of $\mathbb{E}[T]$) [35, 105]; and the mean sojourn time (equivalent to $\mathbb{E}[T]$ when the network can be modeled as a queueing system, see Chapter 3) [89]. A related performance metric is the probability of rejecting requests [36, 42, 65, 111], which we do not use since we assume requests are never rejected. Lastly, note that the term "completion time" has been used in some cases to refer to what we call delivery time [191].

In Section 5.4, we derive analytical solutions and approximations for the expected completion time of OD and CD protocols.

5.3.2. WASTED RESOURCES

As previously discussed, quantum states that are delivered and stored for an extended period before use are subject to decoherence, eventually rendering them unusable. Moreover, if no quantum memories are available for storage when a resource state is delivered, the excess resources must be discarded immediately. CD protocols may encounter this issue: if the delivery subroutine generates resource states significantly faster than requests arrive, some resources might need to be discarded. Next, we define a quantity that measures this resource wastage.

We define $N_\mathrm{d}(t)$ as the total number of resource states distributed in the time interval $(0, t]$ when running a CD protocol. This protocol operates such that the delivery subroutine is triggered immediately after the completion of the previous delivery. Consequently, the times between resource deliveries are i.i.d. random variables with the same distribution as D. Similarly, let $N_\mathrm{r}(t)$ denote the total number of requests that arrive within $(0, t]$. The interarrival times between consecutive requests are also i.i.d. and follow the same distribution as R. Both $\{N_\mathrm{d}(t), t \geq 0\}$ and $\{N_\mathrm{r}(t), t \geq 0\}$ are counting processes. More specifically, they are renewal processes, since the time between consecutive events – whether resource deliveries (D) or request arrivals (R) – are i.i.d. random variables. We define the ratio of extra deliveries per request in the interval $(0, t]$ as

$$W(t) := \frac{N_\mathrm{d}(t)}{N_\mathrm{r}(t)} - 1, \tag{5.3}$$

and we use the value of this ratio in the steady state to measure resource wastage.

Definition 5.2. *The* wastage *of a CD protocol, W, is defined as the value of $W(t)$ in the steady state, i.e., $W := \lim_{t \to \infty} W(t)$.*

By definition, the wastage W is the number of extra resource states delivered per request, in the steady state. As we show in Lemma 5.1, W can also be conveniently computed as the ratio between interrarrival times.

diverge to infinity, as we show later in Section 5.4.

Lemma 5.1. *The wastage can be computed as*

$$W = \frac{\mathbb{E}[R]}{\mathbb{E}[D]} - 1. \tag{5.4}$$

Proof. We can show this as follows:

$$W := \lim_{t\to\infty} W(t) = \lim_{t\to\infty} \frac{N_{\mathrm{d}}(t)/t}{N_{\mathrm{r}}(t)/t} - 1 \overset{\text{a.s.}}{=} \frac{\mathbb{E}[R]}{\mathbb{E}[D]} - 1, \tag{5.5}$$

where the definition of $W(t)$ is applied in the second step, and the strong law of large numbers is applied to the renewal processes $\{N_{\mathrm{d}}(t), t \geq 0\}$ and $\{N_{\mathrm{r}}(t), t \geq 0\}$ in the last step (see Section 10.2, Theorem 1, from [76]). □

From Lemma 5.1, we observe that W exists as long as $\mathbb{E}[R] < \infty$ and $\mathbb{E}[D] > 0$, which we assume to be the case. Recall that we also assume $\mathbb{E}[D] < \mathbb{E}[R]$ to ensure system stability (see assumption (5.1)). As a result, the wastage W is a positive quantity. When a protocol delivers resource states much faster than requests arrive, i.e., $\mathbb{E}[D] \ll \mathbb{E}[R]$, the wastage W becomes large, since the protocol generates an excess of resource states. On the other hand, if $\mathbb{E}[D]$ is close to $\mathbb{E}[R]$, then W remains close to zero, indicating that the CD protocol is efficient in terms of wasted resources.

Wastage is an operational metric to quantify the use of resources, motivated by an engineering perspective. Related metrics include the cost of replenishing entanglement (i.e., the time required to regenerate some entangled state) [166] and the percentage of quantum memories in use within storage nodes, which may store entanglement pre-distributed via a CD protocol [149]. More fundamental quantities, which are used to quantify certain quantum effects such as entanglement, are defined within quantum resource theories (see, e.g., [39] for a review).

5.4. PERFORMANCE ANALYSIS WITH MEMORYLESS USERS

In this section, we explore a scenario in which users lack memories to store delivered resource states. In this setting, resource states must be either immediately consumed or discarded upon delivery. We consider this a worst-case scenario for CD protocols: we expect them to provide a greater advantage over OD protocols in terms of expected completion time when memories are available, as these memories allow for resource states to be delivered in advance and stored for later use. Analyzing the no-memory case serves as a useful baseline for comparing the performance of CD and OD protocols. Even in this restrictive scenario, one would expect CD protocols to achieve shorter expected completion times than OD strategies, since CD protocols initiate the delivery subroutine no later than their OD counterparts. However, our analysis reveals a surprising result: preemptively distributing resource states, as in CD protocols, can result in longer request completion times. This happens when the probability distribution of the delivery time D is broad – specifically, when the standard deviation is larger than the mean, as we show next.

FAST-DELIVERY REGIME. Let us start considering the fast-delivery regime, in which $\mathbb{E}[D] \ll \mathbb{E}[R]$. In this regime, the average resource delivery time is much shorter than the

average interarrival time of requests. As a result, the request queue is expected to remain empty most of the time, allowing incoming requests to be processed almost immediately upon arrival[3]. This observation allows us to estimate the expected request completion time of an OD protocol in the fast-delivery regime as approximately equal to the expected delivery time, that is,

$$\mathbb{E}[T]_{\text{OD}} \approx \mathbb{E}[D], \quad \text{if } \mathbb{E}[D] \ll \mathbb{E}[R], \tag{5.6}$$

where the subindex 'OD' indicates that this is the expected request completion time of an on-demand protocol. It is worth noting that the approximation (5.6) remains valid when the users have quantum memories since, in OD protocols, resources are only delivered upon request and can be consumed immediately, eliminating the need for storage.

When running a CD protocol in the fast-delivery regime, incoming requests are also likely to encounter an empty queue. As a result, the next delivered resource after a request arrival will typically be consumed to fulfill that request. The request completion time can be estimated as the interval between the request arrival and the subsequent resource delivery. As previously discussed, calls to the delivery subroutine are independent and executed sequentially, and therefore resource deliveries form a renewal process. Consequently, the expected completion time corresponds to the mean residual waiting time of such a process[4], which is given by [184]:

$$\mathbb{E}[T]_{\text{CD, no memories}} \approx \frac{\mathbb{E}[D]}{2} + \frac{\text{Var}[D]}{2\mathbb{E}[D]}, \quad \text{if } \mathbb{E}[D] \ll \mathbb{E}[R]. \tag{5.7}$$

Comparing (5.6) and (5.7), we find a striking result: in certain cases, the preemptive delivery of quantum resources, as in CD protocols, can result in larger expected completion times compared to on-demand delivery. In fact, OD protocols, which trigger the delivery subroutine at a later time than their CD counterparts, yield a lower completion time when the following condition is satisfied:

$$\mathbb{E}[T]_{\text{OD}} < \mathbb{E}[T]_{\text{CD, no memories}} \quad \Leftrightarrow \quad \text{Std}[D] > \mathbb{E}[D], \quad \text{if } \mathbb{E}[D] \ll \mathbb{E}[R], \tag{5.8}$$

where $\text{Std}[D] := \sqrt{\text{Var}[D]}$ denotes the standard deviation of the delivery time D. When the delivery time follows a broad distribution, in which the standard deviation is larger than the expected value, CD protocols (without the use of quantum memories) cannot complete requests faster than OD protocols, on average. This phenomenon is closely related to the *inspection paradox*, a well-known and counterintuitive result in renewal theory (see, e.g., ref. [184]).

The previous result may come as a surprise, as one would expect that triggering the delivery subroutine in advance would lead to shorter completion times. Let us provide an intuitive example of why the continuous execution of the delivery subroutine negatively impacts $\mathbb{E}[T]$. Consider a delivery subroutine that delivers resource states after one unit of time with probability 0.95, and takes 100 units of time otherwise. This could be a scenario in which the network can quickly deliver a resource state deterministically, except when

[3] In fact, in our model, incoming requests can always be processed upon arrival if the delivery time is smaller than the request interarrival time, i.e., if there exists some t such that $\Pr(D < t) = 1$ and $\Pr(R < t) = 0$.

[4] The *residual time* at a given instant t is defined as the amount of time remaining between the current time t and the next epoch of the renewal process (in our case, until the next delivery).

it malfunctions, with probability 0.05, in which case it requires 100 units of time to fix the failure and deliver the state. In this case, the expected delivery time is $\mathbb{E}[D] = 5.95$ and the standard deviation is $\mathrm{Std}[D] = 21.58$, and therefore on-demand delivery must lead to a shorter expected completion time than continuous delivery, as dictated by (5.8). If the delivery subroutine is triggered when a request arrives, as in OD protocols, 95% of the requests will take one unit of time to complete, while the other 5% will take 100 units of time – recall that, in the fast-delivery regime approximation, we assume that incoming requests find an empty queue and are processed immediately. This leads to $\mathbb{E}[T]_{\mathrm{OD}} = 5.95$. Conversely, if we continuously perform deliveries, the system is likely to encounter a malfunction at some point and get stuck fixing it: the system will spend, on average, 95 units of time performing a quick delivery for every 500 units of time that are spent fixing a malfunction and performing a slow delivery (5% of the executions take 100 units of time each). Hence, requests are substantially more likely to arrive within a slow execution of the subroutine than during a quick one. Specifically, the probability of a request arriving within a slow execution is $500/595 \approx 0.84$. This imbalance results in a significantly larger expected completion time under the CD protocol compared to the OD protocol: $\mathbb{E}[T]_{\mathrm{CD,\,no\,memories}} \approx 42.08 > \mathbb{E}[T]_{\mathrm{OD}}$.

In Figure 5.3 we present another example. In this case, we consider a delivery subroutine with a running time that follows a lognormal distribution: $D \sim \mathrm{LogNormal}(\mu, \sigma^2)$, where $\mu \in \mathbb{R}$ and $\sigma > 0$. The mean value and the standard deviation are given by

$$\mathbb{E}[D] = e^{\mu + \frac{\sigma^2}{2}}, \quad \mathrm{Std}[D] = e^{\mu+\sigma^2/2}\sqrt{e^{\sigma^2} - 1}. \tag{5.9}$$

We choose this specific distribution for two reasons: (*i*) the delivery time D must be positive, which excludes common distributions, like the normal distribution, that allow negative values, and (*ii*) the lognormal distribution provides flexibility to continuously adjust the parameter σ while keeping $\mathbb{E}[D]$ fixed, allowing the distribution to become progressively broader until condition (5.8) is met. Using (5.8), we find that on-demand delivery provides a lower expected completion time $\mathbb{E}[T]$ when $\sigma > \sqrt{\ln 2}$. In Figure 5.3, we show $\mathbb{E}[T]$ in terms of σ, with $\mu = -\sigma^2/2$ (such that $\mathbb{E}[D] = 1$), in the fast-delivery regime. The curves are calculated using (5.6) and (5.7). The insets display the shape of the delivery time distribution for $\sigma = 0.2, \sqrt{\ln 2}, 1.2$. We observe that, as the distribution broadens, the CD protocol provides increasingly larger $\mathbb{E}[T]$, becoming worse than the OD protocol for $\sigma > \sqrt{\ln 2}$.

Deterministic delivery subroutines are of special interest, since $\mathbb{E}[T]_{\mathrm{CD,\,no\,memories}}$ is minimized when $\mathrm{Var}[D] = 0$ (for a fixed $\mathbb{E}[D]$, see (5.7)). In this case, $\mathbb{E}[T]_{\mathrm{CD,\,no\,memories}} \approx \mathbb{E}[D]/2$, whereas $\mathbb{E}[T]_{\mathrm{OD}} \approx \mathbb{E}[D]$ (see (5.6)). That is, when delivery times are deterministic and users lack quantum memories, CD protocols achieve the greatest advantage in terms of expected completion time, and can complete requests on average twice as fast as OD protocols. Deterministic delivery subroutines are a practical reality. For example, the deterministic delivery of bipartite entanglement has been experimentally demonstrated using nitrogen-vacancy centers in diamond [87].

Lastly, we highlight that, in the fast-delivery regime, CD protocols suffer from significant resource wastage, W. In fact, in the limit of ultra-fast delivery, where $\mathbb{E}[R]/\mathbb{E}[D] \to \infty$, the number of resources wasted per request in a CD protocol diverges to ∞, as shown in Lemma 5.1. Consequently, even if the delivery time distribution is such that

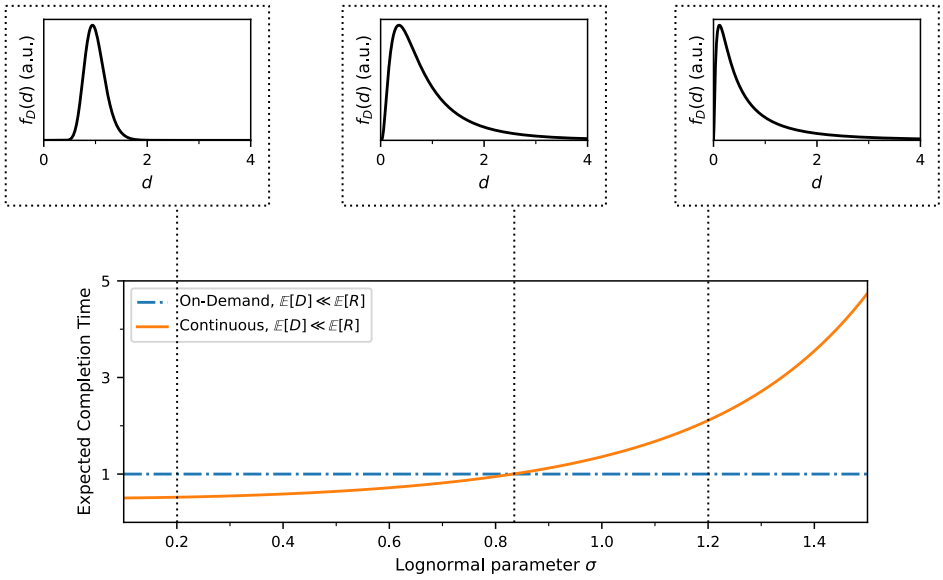

Figure 5.3: **In the fast-delivery regime, preemptive distribution of quantum resources is detrimental when the delivery time distribution is broad.** (Main plot) Expected completion time, $\mathbb{E}[T]$, of an OD protocol (blue dash-dotted line) and a CD protocol (solid orange line) versus the parameter σ. The delivery time follows a lognormal distribution, $D \sim \text{LogNormal}(\mu, \sigma^2)$, where $\mu = -\sigma^2/2$ (such that $\mathbb{E}[D] = 1$ for any value of σ). The users have no memories to store resource states for later use. We consider a fast-delivery regime in which $\mathbb{E}[D] \ll \mathbb{E}[R]$, and use the analytical solutions from (5.6) and (5.7). (Insets) Shape of the probability distribution of D for $\sigma = 0.2, \sqrt{\ln 2}, 1.2$, from left to right.

$\mathbb{E}[T]_{\text{CD, no memories}} < \mathbb{E}[T]_{\text{OD}}$, the excessive resource wastage in CD protocols may justify the use of an OD protocol, despite its higher expected completion time.

SLOW-DELIVERY REGIME. In the previous analysis we concluded that preemptive delivery of resources can be detrimental when $\mathbb{E}[D] \ll \mathbb{E}[R]$. This raises a natural question: *what if resource deliveries are not orders of magnitude faster than request interarrival times? Do CD protocols show a clear advantage in such cases?* Next, we show that the insights drawn from the $\mathbb{E}[D] \ll \mathbb{E}[R]$ regime largely extend to the slow-delivery regime as well.

First, let us explain how to compute $\mathbb{E}[T]$ outside the fast-delivery regime. An OD protocol can be modeled as a GI/G/1 queue since (i) the time in between requests arrivals follows a general distribution (GI), (ii) the time to process and complete a request is given by the running time of the delivery subroutine, D, which follows a general distribution (G), and (iii) requests are placed in a common queue and processed one at a time, in order of arrival. The expected completion time corresponds to the mean sojourn time of this queueing model, which is the addition of the time that a request waits in the queue (waiting time) and the time taken by the delivery subroutine to deliver a resource state and complete the request (service time). Approximations and numerical solutions exist for the GI/G/1 queue (see, e.g., ref. [184] for an overview). Here, we focus on a specific case that

can be solved analytically: when request arrivals follow a Poisson process, meaning that the interarrival times R are exponentially distributed. In this case, the expected request completion time can be computed as the mean sojourn time of an M/G/1 queue, which is given by [184]

$$\mathbb{E}[T]_{\text{OD}} = \mathbb{E}[D] + \frac{\mathbb{E}\left[D^2\right]}{2\left(\mathbb{E}[R] - \mathbb{E}[D]\right)}, \quad \text{if} \quad R \sim \text{Exp}\left(1/\mathbb{E}[R]\right). \tag{5.10}$$

Recall that this solution only applies when the stability condition (5.1) is met – otherwise $\mathbb{E}[T]_{\text{OD}}$ diverges to infinity. Recall also that OD protocols do not benefit from the use of quantum memories, making (5.10) applicable even when users have memories to store resource states. Regarding CD protocols in the slow-delivery regime, we estimate $\mathbb{E}[T]_{\text{CD, no memories}}$ using a discrete-event Monte Carlo simulation[5].

Figure 5.4 shows $\mathbb{E}[T]$ for both types of protocol, versus $\mathbb{E}[R]$. In this figure, we assume $D \sim \text{LogNormal}(\mu, \sigma^2)$, with $\mu = -\sigma^2/2$ (such that $\mathbb{E}[D] = 1$ for any value of σ) and $\sigma = 0.2, \sqrt{\ln 2}, 1.2$ (subplots from left to right). We observe a number of interesting features:

- As $\mathbb{E}[R]$ decreases and approaches $\mathbb{E}[D]$ (in this case, $\mathbb{E}[D] = 1$), $\mathbb{E}[T]$ diverges to infinity. As mentioned earlier, if $\mathbb{E}[R] < \mathbb{E}[D]$, requests would accumulate over time as the system cannot deliver resources fast enough, and completion times would grow to infinity. This behavior was the reason why we imposed Assumption 5.1.

- When $\mathbb{E}[R]$ is large, $\mathbb{E}[T]$ converges to the approximations (5.6) and (5.7) derived for the fast-delivery regime (dotted and dashed lines in Figure 5.4). In fact, the approximation $\mathbb{E}[T]_{\text{OD}} \approx \mathbb{E}[D]$ from (5.6) can be rigorously derived by taking the limit $\mathbb{E}[R] \to \infty$ in the analytical solution (5.10). This serves as a validation check for such approximations.

- Interestingly, the qualitative relative performance of OD and CD protocols in the fast-delivery regime still holds when $\mathbb{E}[R]$ is small. For example, when the distribution is peaked (left subplots in Figure 5.4), $\mathbb{E}[T]_{\text{CD, no memories}} < \mathbb{E}[T]_{\text{OD}}$ for any $\mathbb{E}[R]$, and not only in the limit $\mathbb{E}[R] \gg \mathbb{E}[D]$. Moreover, we observe that, when $\text{Std}[D] = \mathbb{E}[D]$ (central panel in Figure 5.4), both types of protocol provide the same expected completion time for any value of $\mathbb{E}[R]$.

In Appendix 5.6, we study another example in which requests arrive according to a Poisson process but delivery times follow a Gamma distribution, and we extract similar conclusions. We also show that, when requests are Poisson-distributed and delivery times follow an exponential or an Erlang distribution, CD protocols always provide lower $\mathbb{E}[T]$ than OD approaches. Note that it is also possible to derive analytical solutions for $\mathbb{E}[T]_{\text{OD}}$ when requests arrivals are not Poisson-distributed. In Appendix 5.7, we calculate $\mathbb{E}[T]_{\text{OD}}$ for R following an arbitrary distribution and D following an exponential distribution.

To summarize, we have demonstrated that the spread of the delivery time distribution is a key factor in designing an efficient protocol. In particular, CD protocols perform best with sharply peaked distributions, whereas OD protocols are more effective than CD strategies when the distribution is broader. Moreover, as discussed in the context of

[5] Our code can be found in our repository: https://github.com/AlvaroGI/anticipating-quantum-needs.

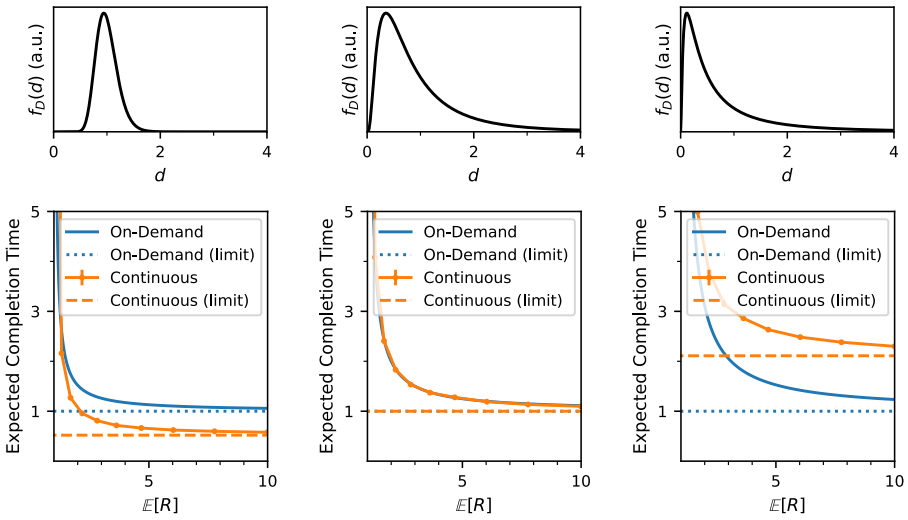

Figure 5.4: **Preemptive distribution of quantum resources can be either advantageous or detrimental depending on the delivery time distribution, regardless of whether we are in the fast-delivery regime.** (Bottom subplots) Expected completion time, $\mathbb{E}[T]$ of an OD protocol (blue line) and a CD protocol (orange line with markers) versus the expected request interarrival time, $\mathbb{E}[R]$. The delivery time follows a lognormal distribution, $D \sim \text{LogNormal}(\mu, \sigma^2)$, with $\sigma = 0.2, \sqrt{\ln 2}, 1.2$ (from left to right) and $\mu = -\sigma^2/2$ (such that $\mathbb{E}[D] = 1$ for any value of σ). The users have no memories to store resource states for later use. We compute $\mathbb{E}[T]$ with (5.10) for the OD protocol and with a discrete-event Monte Carlo simulation for the CD protocol (error bar sizes are smaller than line width). The fast-delivery ($\mathbb{E}[D] \ll \mathbb{E}[R]$) approximations from (5.6) and (5.7) are shown as dotted and dashed lines, respectively. (Top subplots) Shape of the probability distribution of D for $\sigma = 0.2, \sqrt{\ln 2}, 1.2$, from left to right.

the fast-delivery regime, resource wastage is an important consideration when selecting a delivery strategy. As shown in Lemma 5.1, the wastage grows linearly in $\mathbb{E}[R]$. Hence, for small values of $\mathbb{E}[R]$, the wastage may not significantly influence the choice between CD and OD protocols, as opposed to the fast-delivery regime, in which the large wastage associated with CD protocols makes it a critical factor to consider.

5.5. OUTLOOK

In this chapter, we analyzed and compared protocols for on-demand (OD) and continuous delivery (CD) of quantum resource states, focusing on two key performance metrics: the expected request completion time, $\mathbb{E}[T]$, and the amount of wasted resources, W. Specifically, we studied a situation in which users have no memories to store delivered resource states for later use. We found a particularly surprising result: under certain delivery-time probability distributions, preemptive delivery of resource states can paradoxically increase the expected request completion time. This counterintuitive finding highlights the nuanced interplay between delivery strategies and system performance, offering a wide range of follow-up research directions.

First, introducing quantum memories for storing resource states could significantly improve the expected request completion time of CD protocols. If memory lifetimes are short, $\mathbb{E}[T]$ is likely to converge to the results derived in this thesis for the memoryless case. However, even short-lived memories could enable continuous-delivery (CD) protocols to consistently outperform on-demand (OD) strategies in terms of $\mathbb{E}[T]$. An interesting extension of this idea would be to analyze how the use of entanglement buffers, such as those proposed in ref. [50], influences the performance of the system. These buffers could potentially mitigate the limitations of short memory lifetimes, enhancing the efficiency of CD protocols.

A basic assumption we made here is that incoming requests are placed in a common queue, which they only leave after all previous requests have been completed. Relaxing this assumption introduces opportunities to explore more complex dynamics. For instance, allowing requests to leave the queue after a timeout or imposing a finite queue capacity could introduce new challenges, such as dropped requests and increased wastage. Introducing different types of requests, such as different sets of users requesting distinct types of resource states, and understanding their effect on the performance of the system would also be a necessary step towards designing protocols for realistic networks.

Lastly, hybrid protocols that dynamically combine OD and CD strategies based on network conditions and user requirements could provide major performance improvements. By adaptively switching between strategies, these protocols could optimize completion times while keeping the wastage low. Exploring such adaptive approaches might reveal new design principles for quantum resource state delivery.

5.6. [APPENDIX] POISSON REQUESTS AND GAMMA-DISTRIBUTED DELIVERY TIME

Here, we consider a system in which requests arrive according to a Poisson process and delivery times follow a Gamma distribution. We obtain the same conclusions found in Section 5.4, where we assumed delivery times followed a lognormal distribution. The main conclusion is again that, when the delivery time distribution is broad, OD protocols outperform CD protocols in terms of expected completion time.

We consider a delivery subroutine whose running time follows a Gamma distribution: $D \sim \text{Gamma}(k,\theta)$, where $k > 0$ is the shape parameter and $\theta > 0$ is the scale parameter. The probability density function is given by

$$f_{\text{D}}(d;k,\theta) = \frac{d^{k-1}e^{-d/\theta}}{\theta^k \Gamma(k)}, \quad d > 0, \tag{5.11}$$

where $\Gamma(k) = \int_0^\infty t^{k-1}e^{-t}\,dt$ is the Gamma function. The mean value and the standard deviation are given by

$$\mathbb{E}[D] = k\theta, \quad \text{Std}[D] = \sqrt{k}\theta. \tag{5.12}$$

When k is an integer, the distribution becomes an Erlang distribution: a sum of k independent exponentially distributed random variables, each with mean θ. In particular, when $k = 1$, the Gamma distribution simplifies to an exponential distribution with mean θ.

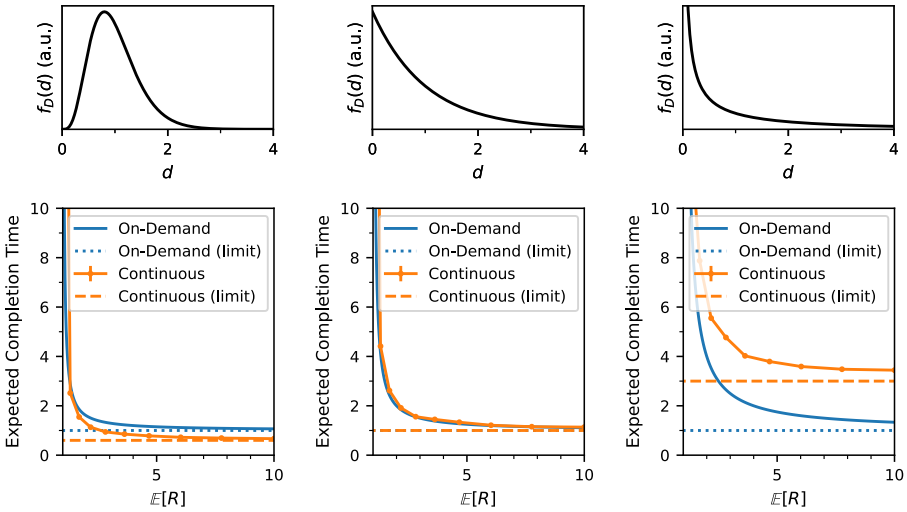

Figure 5.5: **Additional example with Gamma-distributed delivery times: preemptive distribution of quantum resources can be either advantageous or detrimental depending on the delivery time distribution, regardless of whether we are in the fast-delivery regime.** (Bottom subplots) Expected completion time, $\mathbb{E}[T]$ of an OD protocol (blue line) and a CD protocol (orange line with markers) versus the expected request interarrival time, $\mathbb{E}[R]$. The delivery time follows a Gamma distribution, $D \sim \text{Gamma}(k,\theta)$, with $\theta = 0.2, 1, 5$ (from left to right) and $k = 1/\theta$ (such that $\mathbb{E}[D] = 1$). The users have no memories to store resource states for later use. We compute $\mathbb{E}[T]$ with (5.10) for the OD protocol and with a discrete-event Monte Carlo simulation for the CD protocol (error bar sizes are smaller than line width). The fast-delivery ($\mathbb{E}[D] \ll \mathbb{E}[R]$) approximations from (5.6) and (5.7) are shown as dotted and dashed lines, respectively. (Top subplots) Shape of the probability distribution of D for $\theta = 0.2, 1, 5$, from left to right.

FAST-DELIVERY REGIME. When $\mathbb{E}[D] \ll \mathbb{E}[R]$, we know from condition (5.8) that on-demand delivery provides a lower expected completion time $\mathbb{E}[T]$ if and only if $\text{Std}[D] > \mathbb{E}[D]$. In the case of Gamma-distributed delivery times, this condition becomes the following:

$$\mathbb{E}[T]_{\text{OD}} < \mathbb{E}[T]_{\text{CD, no memories}} \quad \Leftrightarrow \quad 0 < k < 1. \tag{5.13}$$

This means that, when requests arrive according to a Poisson process and delivery times follow an exponential or an Erlang distribution, CD protocols provide (on average) faster service than OD approaches.

SLOW-DELIVERY REGIME. Outside the fast-delivery regime, we compute $\mathbb{E}[T]_{\text{OD}}$ using (5.10) and estimate $\mathbb{E}[T]_{\text{CD, no memories}}$ using a discrete-event Monte Carlo simulation. We show three examples in Figure 5.5, with $\theta = 0.2, 1, 5$ (from left to right) and $k = 1/\theta$ (such that $\mathbb{E}[D] = 1$ in all three cases). The top panels show the shape of the delivery time distributions. When the distribution is peaked ($\theta = 0.2$), the CD protocol provides lower $\mathbb{E}[T]$. Both CD and OD protocols yield the same $\mathbb{E}[T]$ when the delivery time distribution is exponential ($\theta = 1$). When the distribution is broad ($\theta = 5$), OD protocols are advantageous.

5.7. [APPENDIX] EXPONENTIALLY-DISTRIBUTED DELIVERY TIME

In this appendix, we derive analytical solutions for the expected completion time of an OD protocol, assuming exponentially-distributed delivery times.

If we use an OD protocol and the delivery time D follows an exponential distribution, the system can be modeled as a G/M/1 queue. The completion time corresponds to the sojourn time of the G/M/1 queue, which is also exponentially distributed [184]:

$$T \sim \text{Exp}\left(\frac{1-\xi}{\mathbb{E}[D]}\right), \tag{5.14}$$

with ξ being the unique solution in the interval $(0,1)$ to

$$\xi = \int_0^\infty e^{-(1-\xi)t/\mathbb{E}[D]} f_R(t)\,dt, \tag{5.15}$$

where f_R is the probability density function (pdf) of the request interarrival time. The integral from (5.15) is known as the Laplace-Stieltjes transform of f_R, evaluated at $(1-\xi)/\mathbb{E}[D]$. If $\mathbb{E}[D] < \mathbb{E}[R]$ (which is our only assumption about the probability distributions of D and R, see (5.1)), it can be shown that there is indeed a unique solution ξ in the interval $(0,1)$ [3]. The expected request completion time is then given by the expected value of the exponential distribution from (5.14):

$$\mathbb{E}[T]_{\text{OD}} = \frac{\mathbb{E}[D]}{1-\xi}, \quad \text{if } D \sim \text{Exp}\left(1/\mathbb{E}[D]\right). \tag{5.16}$$

Let us consider an example in which the integral from (5.15) has a simple form. In this example, we assume that R follows an Gamma(k,θ) distribution. In this case, $\mathbb{E}[R] = k\theta$, and the integral from (5.15) can be solved as follows:

$$\begin{aligned}
\xi &= \int_0^\infty e^{-(1-\xi)t/\mathbb{E}[D]} f_R(t)\,dt \\
&\overset{a}{=} \int_0^\infty e^{-(1-\xi)t/\mathbb{E}[D]} \frac{t^{k-1}}{\theta^k \Gamma(k)} e^{-t/\theta}\,dt \\
&= \frac{1}{\theta^k \Gamma(k)} \int_0^\infty t^{k-1} e^{-((1-\xi)/\mathbb{E}[D]+1/\theta)t}\,dt \\
&\overset{b}{=} \frac{1}{\theta^k \Gamma(k)} \left(\frac{1}{(1-\xi)/\mathbb{E}[D]+1/\theta}\right)^k \int_0^\infty u^{k-1} e^{-u}\,du \\
&\overset{c}{=} \left(\frac{\mathbb{E}[D]}{(1-\xi)\theta+\mathbb{E}[D]}\right)^k,
\end{aligned} \tag{5.17}$$

with the following steps:

(a) We use the pdf of the Gamma function, given in (5.11).

(b) We apply a change of variables: $u = ((1-\xi)/\mathbb{E}[D] + 1/\theta)\,t$.

(c) We use the definition of the Gamma function: $\Gamma(k) = \int_0^\infty t^{k-1} e^{-t}\,dt$.

Equation (5.17) can be efficiently solved for ξ using a numerical solver. When $k = 1$, R is exponentially distributed, which leads to $\xi = \mathbb{E}[D]/\mathbb{E}[R]$, and we recover the solution from (5.10), which we derived for R following an exponential distribution and D following an arbitrary distribution.

The analysis from this appendix, where we have considered $D \sim \text{Exp}(\mathbb{E}[D]^{-1})$ and $R \sim \text{Gamma}(k,\theta)$, can also be used to validate the fast-delivery approximation (5.6) derived in the main text. In the fast-delivery regime, $\mathbb{E}[D] \ll \mathbb{E}[R] = k\theta$. This is achieved when, e.g., $k \to \infty$ (with fixed $\mathbb{E}[D]$ and θ) or $\theta \to \infty$ (with fixed $\mathbb{E}[D]$ and k). Taking one of these limits, we find that $\xi \to 0$ in the fast-delivery regime. Using (5.16), this yields $\mathbb{E}[T]_{\text{OD}} \to \mathbb{E}[D]$, as we predicted in the estimate (5.6).

CODE AVAILABILITY

Our code can be found in the following GitHub repository:
https://github.com/AlvaroGI/anticipating-quantum-needs.

AUTHOR CONTRIBUTIONS

ÁGI conceived and defined the project, conducted the analysis and was the writer of this chapter. SW provided active feedback during the project.

6

Entanglement Buffering with Two Quantum Memories

Álvaro G. Iñesta*, Bethany Davies*, and Stephanie Wehner

Pura simpleza, riqueza en la mesa.
[*Pure simplicity, abundance at the table.*]
— Ana Tijoux

Quantum networks crucially rely on the availability of high-quality entangled pairs of qubits, known as entangled links, distributed across distant nodes. Maintaining the quality of these links is a challenging task due to the presence of time-dependent noise, also known as decoherence. Entanglement purification protocols offer a solution by converting multiple low-quality entangled states into a smaller number of higher-quality ones. In this work, we introduce a framework to analyze the performance of entanglement buffering setups that combine entanglement consumption, decoherence, and entanglement purification. We propose two key metrics: the availability, which is the steady-state probability that an entangled link is present, and the average consumed fidelity, which quantifies the steady-state quality of consumed links. We then investigate a two-node system, where each node possesses two quantum memories: one for long-term entanglement storage, and another for entanglement generation. We model this setup as a continuous-time stochastic process and derive analytical expressions for the performance metrics. Our findings unveil a trade-off between the availability and the average consumed fidelity. We also bound these performance metrics for a buffering system that employs the well-known bilocal Clifford

* These authors contributed equally.
This chapter has been published separately in ref. [50].

purification protocols. Importantly, our analysis demonstrates that, in the presence of noise, consistently purifying the buffered entanglement increases the average consumed fidelity, even when some buffered entanglement is discarded due to purification failures.

6.1. INTRODUCTION

The functionality of quantum network applications typically relies on the consumption of entangled pairs of qubits, also known as *entangled links*, that are shared among distant nodes [194]. The performance of quantum network applications does not only depend on the rate of production of entangled links, but also on their quality. In a quantum network, it is therefore a priority for high-quality entangled states to be readily available to network users. This is a challenging task, since entangled links are typically stored in memories that are subjected to time-dependent noise, meaning that the quality of stored entangled links decreases over time. This effect is known as decoherence.

A common way of overcoming the loss in quality of entangled links is to use *entanglement purification* protocols [15, 53, 56, 205]. An m-to-n entanglement purification protocol consumes m entangled quantum states of low quality and outputs n states with a higher quality, where typically $m > n$. The simplest form of purification schemes are 2-to-1, also known as *entanglement pumping* protocols. One downside of using purification is that there is typically a probability of failure, in which case the input entangled links must be discarded and nothing is produced.

In this work, we take a crucial step towards the design of high-quality entanglement buffering systems. The goal of the buffer is to make an entangled link available with a high quality, such that it can be consumed at any time for an application. We develop methods to analyze the performance of an entanglement buffering setup in a system with entanglement consumption, decoherence, and entanglement pumping. We introduce two metrics to study the performance: (*i*) the *availability*, which is the steady-state probability that a link is available, and (*ii*) the *average consumed fidelity*, which is the steady-state average quality of entangled links upon consumption. We measure the quality of quantum states with the fidelity, which is a well-known metric for this [138].

We use these metrics to study a two-node system where each of the nodes has two quantum memories, each of which can store a single qubit (see Figure 6.1). This system is of practical relevance since early quantum networks are expected to have a number of memories per node of this order (e.g., in [100] and [204], entanglement purification was demonstrated experimentally between two distant nodes, each with the capability of storing two qubits). We study a system where each node has one good (long-term) quantum memory, G, and one bad (short-term) memory, B, per node. We therefore refer to this entanglement buffering setup as the *1G1B system*. The good memories are used to store an entangled link between the nodes that can be consumed at any time. The bad memories are used to generate a new entangled link between the nodes. The new link may be used to pump the stored link with fresh entanglement.

Calculating the temporal evolution of the fidelity of an entangled link is generally a difficult task, since the fidelity depends on the history of operations that have been applied to the link in the past. By modelling the state of the 1G1B system as a continuous-

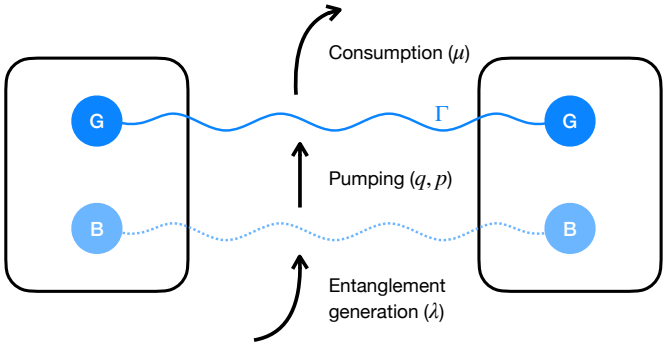

Figure 6.1: **Illustration of the entanglement buffering system with two quantum memories (1G1B system).** Each of the nodes has two memories (G and B). Memory G is used to store the buffered link. An entangled link is generated at a rate λ in memory B. If memory G is empty when the new link is generated in B, the link is immediately transferred to G. If memory G is occupied, the new link generated in B is immediately used to purify the buffered link with probability q (otherwise, the new link is discarded). The pumping protocol consumes the link in B to increase the quality of the buffered link in G, and it succeeds with probability p (otherwise, it destroys the link in G). The buffered link is consumed at a fixed rate μ. The quality of the entanglement stored in G decays exponentially with rate Γ. Formal definitions of the problem parameters can be found in Section 6.3.

time stochastic process, we are able to find analytical solutions for the availability and the average consumed fidelity of the system. We illustrate the application of these results in a simplified scenario where purification has a linear action on the quality of the buffered link.

Our main contributions are the following:

- We propose two metrics to measure the performance of an entanglement buffering system: the availability and the average consumed fidelity.
- We provide a simple closed-form expression for the availability in the 1G1B system.
- We develop an analytical framework to calculate the average fidelity of the links consumed in a 1G1B system. We provide a closed-form expression for pumping schemes that increase the fidelity of the entangled link linearly with the initial fidelity.

Our main findings are the following:

- We confirm the intuition that, except in some edge cases, there is a trade-off between availability and average consumed fidelity: one must either consume low-quality entanglement at a higher rate, or high-quality entanglement at a lower rate.
- Consider a situation where bilocal Clifford protocols are employed (this is one of the most popular and well-studied classes of purification protocols [52]). Then, if the noise experienced by the quantum memories is above certain threshold, pumping the stored link with fresh entanglement always increases the average consumed fidelity, even if the stored link is often discarded due to a small probability of successful pumping. We provide an explicit expression for this noise threshold,

which depends on the purification protocol employed and the fidelity of newly generated links.

The structure of the chapter is the following. In Section 6.2, we provide a short overview of related work. In Section 6.3, we explain the physical setup and provide a formal definition of the 1G1B system as a stochastic process. In Section 6.4, we define the performance metrics of interest and provide analytical expressions that enable their computation. In Section 6.5, we analyze the system in the case where the pumping protocol produces an output state whose fidelity is a linear function of the fidelity of one of the input states. In Section 6.5.1, we use these results to bound the performance of the 1G1B system, in the case where bilocal Clifford protocols are employed for entanglement pumping. Lastly, in Section 6.6, we discuss the implications of this work and future research directions.

6.2. RELATED WORK

The performance analysis of quantum networks is unique because of the trade-off between the rate of distribution of entangled links and the quality of distributed links, both of which are important for the functionality of networking applications. This leads to interesting stochastic problems, which are important to understand the parameter regimes of a possible architecture. For example, [91, 116, 167] deal with the problem of generating an end-to-end entangled link across a chain of quantum repeaters, where both the rate of production and the quality of the end-to-end links are quantities of interest. Another example is the problem of generating multiple entangled links between two users with a high quality, which is treated in [49, 150]. In these works, the time between successfully generated entangled links is modelled by a geometric distribution. However, the time taken up by an entanglement generation attempt is generally small compared to other relevant time scales [122, 147]. Hence, a simplifying assumption that we make in this work is that the time between entanglement generation attempts is exponentially distributed. This is a common assumption in the quantum networking literature (see, e.g., [36, 137, 187]), because it can enable the finding of closed-form relations between physical variables and protocol parameters. Here, we introduce and find expressions for the values of two key performance metrics in the steady state.

Previous work that incorporates entanglement purification schemes into the analysis of quantum network architectures typically involves numerical optimization methods (see, e.g., [189]), or only considers specific purification protocols [21]. By contrast, in this work we focus on presenting the purification protocol in a general way, and finding closed-form solutions for the performance metrics of interest (albeit for a simpler architecture). This is an important step towards an in-depth understanding of how one can expect purification to impact the performance of a near-term quantum network.

Other works have introduced the concept of entanglement buffering (preparing quantum links to be consumed at a later time) over a large-scale quantum network [94, 149]. To the best of our knowledge, the only work with a similar set-up to ours is [60], which was developed in parallel and independently of our work. There, the authors study the steady-state fidelities of a system involving two memories used for storage (good memo-

ries), and one memory used for generation (bad memory). This work differs from ours in multiple ways. For example, the analysis is done in discrete time and it is assumed that the fidelity takes a discrete set of values, whereas we do not make this assumption since we work in continuous time. Additionally, consumption of entanglement is not included in the system studied in [60], which may impact the steady-state behavior.

Lastly, we note that previous work generally assumes a specific protocol for entanglement buffering between each pair of nodes, and does not address the following fundamental question: *what is the best way to buffer entanglement between two users in a quantum network?* To the best of our knowledge, we address this question for the first time.

6.3. THE 1G1B (ONE GOOD, ONE BAD) SYSTEM

We now define the 1G1B system. In Section 6.3.1, we describe and motivate the model of the system. In Section 6.3.2, we define the variables of interest precisely. This facilitates the definition of the performance metrics in Section 6.4.

6.3.1. SYSTEM DESCRIPTION

Below we provide a list of assumptions that model the 1G1B system, and provide motivation for each assumptions. An illustration of the system is given in Figure 6.1.

1. **Each of the nodes has two memories: one long-term memory (good, G) and one short-term memory (bad, B). The B memories are used to generate new entangled links. The G memories are used as long-term storage (entanglement buffer).**

   This is motivated by the fact that *storage* (G) and *communication* (B) qubits are often present in experimental scenarios, where the former is used to store entanglement and the latter is used to generate new links [11, 100, 114].

2. **New entangled links are generated in memory B according to a Poisson process with rate λ. New entangled links always have the form ρ_{new}.**

   Physical entanglement generation attempts are typically probabilistic and heralded [9, 16]. In other words, the attempt can fail with some probability and, when this occurs, a failure flag is raised. Therefore, the generation of a single link may take multiple attempts. The time taken by an attempt is typically fixed (this is both the case in present-day quantum networks [147] and an assumption that is commonly made in the theoretical analysis of quantum networks [49, 91, 94]). Then, the time between attempts follows a geometric distribution. Since the probability of successful generation and the length of the time step is often small compared to other relevant time scales [121, 147], we use a continuous approximation, i.e., that the time between arrivals are exponentially distributed. This is a Poisson process (see, e.g., Chapter 6.8 from [76]).

3. **When a link is newly generated in memory B, if memory G is empty (no link present), the new link is immediately placed there. If memory G is not empty, the**

nodes immediately either (*i*) attempt pumping with probability q, or (*ii*) discard the new link from memory B (probability $1-q$).

This step is included because it may not always be a good idea to carry out pumping, due to there being a possibility of this failing.

4. **Links stored in memory G are Werner states.**

   Werner states take the simple form

$$\rho_{\mathrm{W}} = F\left|\phi^+\right\rangle\!\left\langle\phi^+\right| + \frac{1-F}{3}\left|\psi^+\right\rangle\!\left\langle\psi^+\right| + \frac{1-F}{3}\left|\psi^-\right\rangle\!\left\langle\psi^-\right| + \frac{1-F}{3}\left|\phi^-\right\rangle\!\left\langle\phi^-\right|,$$

   where $\{|\phi^+\rangle, |\psi^+\rangle, |\phi^-\rangle, |\psi^-\rangle\}$ denote the Bell basis. A Werner state corresponds to maximally entangled state that has been subjected to isotropic noise. The state in the good memory is therefore fully described by one parameter: the fidelity F to the target state $|\phi^+\rangle$. Any state can be transformed into a Werner state with the same fidelity by applying extra noise, a process known as *twirling* [13, 86]. Hence, this assumption constitutes a worst-case model.

5. **While in memory G, states are subject to depolarizing noise with memory lifetime $1/\Gamma$.**

   Depolarizing noise can also be seen as a worst-case noise model [58]. After a time t in memory, this maps the state fidelity F to

$$F \to e^{-\Gamma t}\left(F - \frac{1}{4}\right) + \frac{1}{4}.$$

6. **Consumption requests arrive according to a Poisson process with rate μ. When a consumption request arrives, if there is a stored link in memory G, it is immediately used for an application (and therefore removed from the memory). If there is no link available, the request is ignored.**

   This means that the time until the next consumption request arrives is independent of the arrival time of previous requests, and it is exponentially distributed. This assumption is commonly made in the performance analysis of queuing systems (see, e.g., Chapter 14 from [184]).

7. **Assumptions about pumping:**

   (a) **Pumping is carried out instantaneously.**

   This is because the execution time is generally much lower than the other timescales involved in the problem. For example, in state-of-the-art setups, an entangled link is generated approximately every 0.5 s [147], while entanglement pumping may take around $0.5 \cdot 10^{-3}$ s [100]. If the nodes are far apart, classical communication between them would only add a negligible contribution to the purification protocol (e.g., classical information takes less than 10^{-4} s to travel over 10 km of optical fiber).

(b) **Suppose that the link in memory G has fidelity F and the link in memory B is in state ρ_{new}. If pumping succeeds, the output link has fidelity $J(F,\rho_{\text{new}})$, and remains in the good memory. If pumping fails, all links are discarded from the system.** Here, the *jump function* $J(F,\rho_{\text{new}}) \in [0,1]$ is dependent on the choice of purification protocol. Given the assumption that one of the links is a Werner state, the form of this function is

$$J(F,\rho_{\text{new}}) = \frac{\tilde{a}(\rho_{\text{new}})F + \tilde{b}(\rho_{\text{new}})}{p(F,\rho_{\text{new}})}, \tag{6.1}$$

with

$$p(F,\rho_{\text{new}}) = c(\rho_{\text{new}})F + d(\rho_{\text{new}}) \tag{6.2}$$

where $\tilde{a},\tilde{b},c,d$ are functions of ρ_{new}. Here, $p(F,\rho_{\text{new}})$ is the success probability of purification. See Appendix 6.7 for an explanation of why the jump function and success probability take this form.

(c) **Pumping succeeds with probability p, which is constant in the fidelity of memory G.** We see from the above that this is a special case, and that in general the probability of purification success varies linearly with the fidelity of the good memory. However, performing the analysis with a constant probability of success does allow us to find bounds on the operating regimes of the system by considering the best-case and worst-case values of p (see Section 6.5.1). Combining this with Assumption 7b, we see that this is effectively equivalent to setting $c(\rho_{\text{new}}) = 0$. The jump function is then linear in the fidelity of memory G, and can be written as

$$J(F,\rho_{\text{new}}) = a(\rho_{\text{new}})F + b(\rho_{\text{new}}),$$

where $a := \tilde{a}/p$ and $b := \tilde{b}/p$.

Implicit in the above is that the process of entanglement generation, pumping and consumption ((2),(3),(6) and (7b)) are independent. We provide a summary of the parameters involved in the 1G1B system in Table 6.1.

6.3.2. SYSTEM DEFINITION

In this subsection, we define the state of the system mathematically, which will be the main object of study in the rest of this work. We view the state of the system as the number of rounds of pumping that the link in memory has undergone. From now on, when we refer to 1G1B, we refer to the stochastic process that evolves according to the following definition.

Definition 6.1 (1G1B system)**.** *Let $s(t)$ be the state of the 1G1B system at time t. This takes values*

$$s(t) = \begin{cases} \emptyset & \text{if there is no link in memory,} \\ i \geq 0 & \text{if there is a link in memory which is the result of } i \text{ successful pumping rounds,} \end{cases} \tag{6.3}$$

Table 6.1: **Parameters of the 1G1B system.** See main text for detailed explanations.

Hardware	
λ	Rate of heralded entanglement generation (time between successful attempts is exponentially distributed with rate λ)
ρ_{new}	Entangled state produced after a successful entanglement generation
Γ	Rate of decoherence (fidelity of the entangled link decays exponentially over time with rate Γ)
Application	
μ	Rate of consumption (specified by application)
Pumping protocol	
q	Probability of attempting pumping immediately after a successful entanglement generation attempt (otherwise the new link is discarded)
p	Probability of successful pumping
$J(F, \rho_{\text{new}})$	Jump function: fidelity of the output state following successful pumping (F is the fidelity of the Werner state stored in the good memory)

where $i = 0$ corresponds to a link in memory that has not undergone any pumping. Assume that the system starts with no link, i.e., $s(0) = \emptyset$. The system transitions from state $\emptyset$ to state 0 when a new link is generated and placed in the good memory, which was previously empty. The rate of transition from $\emptyset$ to 0 is then given by the entanglement generation rate λ. Pumping success occurs when a new link is produced (rate λ), pumping is attempted (probability q), and pumping succeeds (probability p). Therefore, the transition from state i to $i+1$ occurs with rate $\lambda q p$. The final allowed transition is from i to $\emptyset$ which occurs due to consumption or purification failure, which occurs with rate $\mu + \lambda q(1-p)$.

We also refer to the state $i \geq 0$ as the ith *purification level*. Since the transitions between each state in 1G1B occur according to an exponential distribution with rate that is only dependent on the current state of the system, this is a continuous-time Markov chain (CTMC) on the state space $\{\emptyset, 0, 1, ...\}$. The resulting CTMC and the rate of transitions is depicted in Figure 6.2. This is the main object of study in our work.

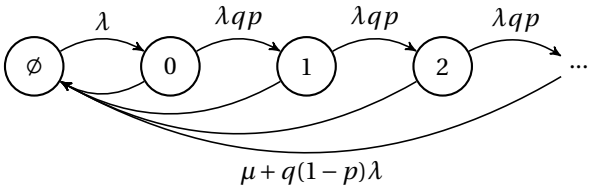

Figure 6.2: **The transitions of the 1G1B system.**

Recall that we are also interested in the fidelity of the link in memory. This is dependent not only on the state $s(t) \in \{\emptyset, 0, 1, ...\}$, but also on the time spent in the states leading up to the current purification level. This motivates the following definition.

Definition 6.2. *Suppose that $s(t) = i$. Then, we define random variable $\vec{T}(t)$ to be the length-$(i+1)$ vector storing the times spent in the recent purification levels $0, 1, \ldots, i$ leading up to the current one, where time $T_j(t)$ was spent in the most recent visit to state j ($j \leq i-1$), and time $T_i(t)$ is the time spent so far in state i. See Figure 6.3 for a depiction of this.*

We also need a framework with which to compute the fidelity at time t. Recalling assumption (5) of Section 6.3.1, we denote decoherence by the following.

Definition 6.3. *Let $D_t : [0,1] \to [0,1]$ denote the action of depolarizing noise on the state fidelity F. This has action*

$$D_t[F] = e^{-\Gamma t}\left(F - \frac{1}{4}\right) + \frac{1}{4}.$$

We now formally define the jump function.

Definition 6.4. *After successfully applying purification to a Werner state with fidelity F and a general two-qubit state ρ_{new}, the output state has fidelity $J(F, \rho_{\text{new}})$. We refer to J as the jump function of the protocol. The general form of this is given in (6.1).*

We note that every purification protocol has a corresponding jump function. The exact form of J is dependent on the choice of pumping protocol, but in general is a continuous rational function of F, taking values in $[0,1]$.

We also need to compute the fidelity after many rounds of decoherence and pumping. This essentially means composing D_t and J.

Definition 6.5. *Let $F^{(i)}(t_0, ..., t_i)$ denote the fidelity after spending time $t_0, ..., t_i$ in each purification level $0, 1, ..., i$. This may be defined recursively as*

$$F^{(i)}(t_0, ..., t_i) = D_{t_i}\left[J(F^{(i-1)}(t_0, \ldots, t_{i-1}), \rho_{\text{new}})\right], \tag{6.4}$$

with $F^{(0)}(t_0) = D_{t_0}[F_{\text{new}}]$, where F_{new} is the fidelity of ρ_{new}.

Note that $F^{(i)}$ is a continuous and bounded function of its inputs, since the same is true for D_t and J. We are now equipped to define the fidelity of the system.

Definition 6.6. *The fidelity of the 1G1B system at t is given by*

$$F(t) = \begin{cases} F^{(i)}(\vec{T}(t)) \text{ if } s(t) = i \geq 0, \\ 0, \text{ if } s(t) = \emptyset. \end{cases} \tag{6.5}$$

Note that this formulation can also be adapted to incorporate a system where we apply a different pumping protocol in each state of the CTMC. In that case, we would employ a more general recurrence relation:

$$F^{(i)}(t_0, \ldots, t_i) = D_{t_i}\left[J^{(i)}(F^{(i-1)}(t_0, \ldots, t_{i-1}), \rho_{\text{new}})\right], \tag{6.6}$$

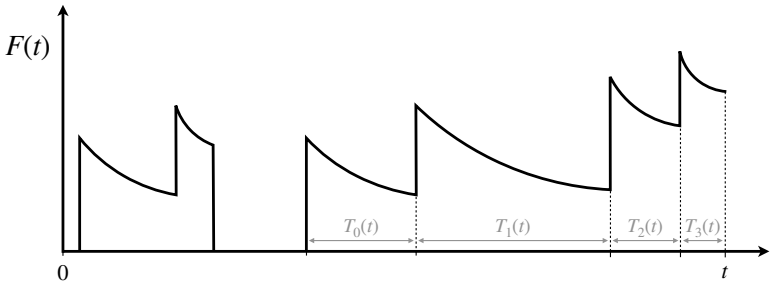

Figure 6.3: **Example of the evolution of the fidelity of the buffered entanglement over time.** The fidelity experiences a sudden boost every time a pumping protocol is successful. Then, it decays exponentially due to decoherence. Each state in the CTMC is identified by the number of times the current buffered link has been purified. If $s(t) = i$, the random variables $\{T_j(t) : j = 0, 1, ..., i-1\}$ are the times spent in each state of the CTMC immediately leading up to the current state, and $X_i(t)$ is the time so far spent in state i.

where the $J^{(i)}$ is the jump function corresponding to the pumping protocol applied in state i of the CTMC. For simplicity, however, we study recurrence relations of the form (6.4). This may be used to model the situation where the same pumping protocol is applied every time, or provide bounds for using multiple protocols, as we do in Section 6.5.1.

6.4. PERFORMANCE METRICS

In this section, we define two metrics to evaluate the performance of an entanglement buffering system: the *availability* and the *average consumed fidelity*. We also provide analytical expressions for both metrics in the 1G1B system.

6.4.1. AVAILABILITY

A natural measure for the quality of service provided to users is the probability that a consumption request may be served at any given time. If there is a link stored in the good memories, the consumption request is immediately served. However, if there is no entanglement available, the request is ignored. Letting $P(s(t) = i)$ be the probability that the system is in state i at time t, we define the steady-state distribution as

$$\pi_i := \lim_{t\to\infty} P(s(t) = i). \tag{6.7}$$

Then, we define our first performance metric as follows.

Definition 6.7 (Availability)**.** *The availability A is defined as*

$$A := 1 - \pi_\emptyset, \tag{6.8}$$

which is the probability that there is a link in memory in the limit $t \to \infty$.

This definition can be applied to any entanglement buffering setup. In the 1G1B system, the availability is well-defined, as shown in Appendix 6.8. Moreover, it is possible to derive a closed-form expression for the availability, as stated in the proposition below.

Proposition 6.1. *Consider the 1G1B system (Definition 6.1). The availability is given by*

$$A = 1 - \pi_\emptyset = \frac{\lambda}{\lambda + \mu + \lambda q(1-p)}, \tag{6.9}$$

and the rest of the steady-state distribution is given by

$$\pi_i = \frac{\lambda^{i+1} q^i p^i}{(\mu + \lambda q)^{i+1}} \pi_\emptyset. \tag{6.10}$$

See Appendix 6.8 for a proof of this proposition. We note that this can be derived in a straightforward manner using the balance equations for a CTMC. Instead, we use renewal theory, for two reasons. Firstly, this approach ties in neatly with the proof of the formula for the average fidelity (see the next subsection). Secondly, this approach provides a formula for the availability that is more general, as it also applies to the case where entanglement generation is described by a general random variable instead of being exponentially distributed. See Appendix 6.8 for the general formula for the availability.

6.4.2. AVERAGE CONSUMED FIDELITY

The quality of service of an entanglement buffering system can also be measured in terms of the quality of the entanglement provided to the users. Therefore, the average fidelity of the entangled links upon consumption can be used as an additional metric to assess the performance of the system.

Definition 6.8 (Average consumed fidelity)**.** *The average consumed fidelity is the average fidelity of the entangled link upon consumption, in the steady state. More specifically,*

$$\overline{F} \coloneqq \lim_{t\to\infty} \mathbb{E}\big[\, F(t) \,|\, s(t) \neq \emptyset \,\big]. \tag{6.11}$$

In the definition of $\overline{F}$, we condition on not being in $\emptyset$ since consumption events do not happen when there is no link present. As before, this performance metric can be applied to any entanglement buffering setup. In the case of the 1G1B system, it is possible to derive an analytical expression for $\overline{F}$ which explicitly depends on the steady-state distribution. The formula is given in the following theorem.

Theorem 6.1. *In the 1G1B system, the average consumed fidelity can be written as*

$$\overline{F} = \frac{1}{A} \sum_{i=0}^{\infty} c_i \pi_i, \tag{6.12}$$

where $\pi_i = \lim_{t\to\infty} P(s(t) = i)$*, and*

$$c_i = \mathbb{E}\left[F^{(i)}(Q_0, Q_1, ..., Q_i) \right] \tag{6.13}$$

where A is the availability, $Q_0, Q_1, \ldots, Q_i$ *are i.i.d. random variables with* $Q_0 \sim Exp(\mu + \lambda q)$*, and* $F^{(i)}$ *is given in Definition 6.5.*

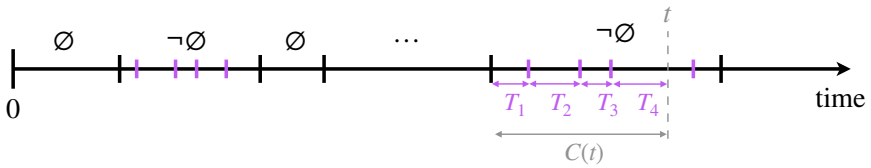

Figure 6.4: **An example timeline of the 1G1B process.** Black dashes are link generation and removal. Shorter purple dashes are pumping rounds. If there is a link present at time t, the random variable $C(t)$ is the total time spent so far in $\neg\emptyset$ (link present). Pumping rounds occur within the time $C(t)$ as a Poisson process with rate λqp. This may be used to characterize the distribution of $\vec{T}(t)$ in the limit $t \to \infty$, which is needed to prove Theorem 6.1.

Sketch proof of Theorem 6.1. A first step is to expand by conditioning on the value of $s(t)$,

$$\begin{aligned}\mathbb{E}[F(t)|s(t)\neq\emptyset] &= \sum_{i=0}^{\infty}\mathbb{E}[F(t)|s(t)=i]P(s(t)=i|s(t)\neq\emptyset)\\ &= \frac{1}{P(s(t)\neq\emptyset)}\sum_{i=0}^{\infty}\mathbb{E}[F(t)|s(t)=i]P(s(t)=i)\,.\end{aligned}$$

In Proposition 6.5 (Appendix 6.8.2), we show that, when $t\to\infty$, the limit can be brought inside of the sum, and so

$$\begin{aligned}\overline{F} &= \lim_{t\to\infty}\mathbb{E}[F(t)|s(t)\neq\emptyset]\\ &= \frac{1}{A}\sum_{i=0}^{\infty}\pi_i\cdot\lim_{t\to\infty}\mathbb{E}\big[F(t)|s(t)=i\big],\end{aligned}$$

where we have used the definition of the steady-state distribution and the availability (see (6.7) and (6.8)). The values π_i may be computed using Proposition 6.1. The remaining work is then to show that

$$\lim_{t\to\infty}\mathbb{E}\big[F(t)|s(t)=i\big] = \mathbb{E}\left[F^{(i)}(Q_0,\ldots,Q_i)\right], \tag{6.14}$$

which essentially requires the characterization of the limiting distribution of $\vec{T}(t)$, since from Definition 6.6 we recall that $\mathbb{E}\big[F(t)|s(t)=i\big] = \mathbb{E}\big[F^{(i)}\left(\vec{T}(t)\right)\big|s(t)=i\big]$. This is achieved with the following result: conditional on $s(t)=i$, $\vec{T}(t)\to(Q_0,\ldots,Q_i)$ in distribution as $t\to\infty$, where the Q_j are i.i.d. random variables with $Q_0\sim\text{Exp}(\mu+\lambda q)$. There are two main steps to show this (see Figure 6.4 for graphical intuition):

1. Let $C(t)$ be the total time spent so far in $\neg\emptyset$ (link in memory G) at the time t. The first step is to show that $C(t)\to C$ in distribution as $t\to\infty$, where $C\sim\text{Exp}(\mu+\lambda q(1-p))$. This is shown with renewal theory. For more details, see the results of Appendix 6.8.1.

2. Characterize the limiting distribution of the time spent in each purification level *within* the time $C(t)$. These are the $T_j(t)$. We use the fact that pumping rounds occur as a Poisson process within the time $C(t)$. For more details, see the results of Appendix 6.8.2.

Finally, since $F^{(i)}$ is a continuous function of its inputs, (6.14) follows. □

For the full proof, see Appendix 6.8. The particularly simple form of (6.13) can be attributed to the fact that in a CTMC, the time spent in a state is not influenced by the state to which the system transitions. As an example, in the CTMC from Figure 6.5, the time spent in state B before a transition does not depend on the transition itself, and this time is exponentially distributed with rate $r_{BA} + r_{BC}$. In the 1G1B system, the times spent in the states $j = 0, 1, \ldots, i-1$ leading up to state i are all exponentially distributed with rate $\lambda qp + \mu + \lambda q(1-p) = \mu + \lambda q$. As a consequence, the average fidelity after i successful purifications, c_i, does not depend on the probability of successful purification p.

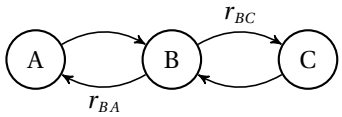

Figure 6.5: **In a CTMC, the time spent in a state is independent of the transition that happens next.** In this example, the time spent in state B before leaving is exponentially distributed with rate $r_{BA} + r_{BC}$.

Having systematic closed-form expressions for the functions $F^{(i)}$ enables the efficient computation of c_i and, therefore, $\overline{F}$. The calculation of $F^{(i)}$ in closed-form for a general J is quite involved, since the recurrence relation (6.4) becomes a rational difference equation with arbitrary coefficients. However, in the following sections we consider a jump function which is linear, for which it is possible to find a closed-form solution for $\overline{F}$.

6.5. ENTANGLEMENT BUFFERING WITH A LINEAR JUMP FUNCTION

In a purification protocol with a linear jump function, the output fidelity is a linear function of the fidelity of one of the input entangled links. When the probability of successful purification is constant with the fidelity of the good memory, as we assume in 1G1B, this implies that the jump function is linear. This is shown in Appendix 6.7. In this section, we compute a closed-form solution for the average consumed fidelity in a 1G1B system assuming a linear jump function. Then, we analyze the performance of the system using the performance metrics defined in Section 6.4 (availability and average consumed fidelity). In Section 6.5.1, we focus on bilocal Clifford protocols, an important type of purification scheme. For a given value of target availability, we provide upper and lower bounds on the average consumed fidelity that can be achieved by any bilocal Clifford protocol in the 1G1B system.

Purification protocols with linear jump functions are relevant for two main reasons:

(i) Purification protocols are generally more effective within some range of input fidelities (the increase in fidelity is larger when the input fidelities are within some interval). If the system operates within a small range of fidelities, one may approximate the true jump function with a linear jump function.

(ii) One can find linear jump functions that upper and lower bound a set of jump functions of interest. These may then be used to upper and lower bound a fidelity-based performance metric (such as the average consumed fidelity) of a system that has the freedom to employ any of these jump functions.

In Appendix 6.10, we demonstrate (ii) in the case where bilocal Clifford protocols are employed in the 1G1B system. The output fidelity of a bilocal Clifford protocol can be upper and lower bounded by nontrivial linear functions when one of the input states is a Werner state (using some additional minor assumptions).

Consider a pumping scheme that takes as input a Werner state with fidelity F and an arbitrary state ρ_{new}. In the 1G1B system, these are the states in the good and the bad memories, respectively. A linear jump function can be written as

$$J(F,\rho_{\text{new}}) = a(\rho_{\text{new}})F + b(\rho_{\text{new}}), \tag{6.15}$$

with $0 \leq a(\rho_{\text{new}}) \leq 1$ and $(1-a(\rho_{\text{new}}))/4 \leq b(\rho_{\text{new}}) \leq 1-a(\rho_{\text{new}})$, as shown in Proposition 6.6. In what follows, we implicitly assume that a and b depend on ρ_{new}.

We now derive a closed-form solution for the average consumed fidelity of 1G1B when the jump function is linear, using Theorem 6.1. The formula requires knowledge of the steady state distribution $\{\pi_i : i = \emptyset, 0, 1, ...\}$, and the expected fidelities c_i, as defined in (6.13). Recall that we assume a constant p, and therefore the steady-state distribution is independent of the jump function. Hence, we can use the formula for π_i from Proposition 6.1. The work then lies in computing the c_i, which are dependent on the choice of jump function, recalling their definition in (6.13). From the same equation, we see that the first step to compute c_i is to find an explicit solution for the function $F^{(i)}$. The linear jump function (6.15) allows us to do this by solving the recurrence relation (6.4). The explicit form of $F^{(i)}$ is provided in the following proposition (see Appendix 6.9.2 for a proof).

Proposition 6.2. *Consider a 1G1B system with $J(F,\rho_{\text{new}}) = aF+b$ and $F^{(0)}(t_0) = D_{t_0}(F_{\text{new}})$, where F_{new} is the fidelity of the state ρ_{new}. Then,*

$$F^{(i)}(t_0,...,t_{i-1},t_i) = \frac{1}{4} + \sum_{j=0}^{i} m_j^{(i)} e^{-\Gamma(t_j+t_{j+1}...+t_i)} \tag{6.16}$$

where the constants $m_j^{(i)}$ are given by $m_0^{(0)} = F_{\text{new}} - \frac{1}{4}$, and

$$m_j^{(i)} = \begin{cases} a^{i-j}\left(\frac{a}{4}+b-\frac{1}{4}\right), \text{ if } j>0, \\ a^i\left(F_{\text{new}}-\frac{1}{4}\right) \text{ if } j=0. \end{cases} \tag{6.17}$$

for $i > 0$.

In the following Lemma, we use the formula for $F^{(i)}$ (found in Proposition 2) and combine this with Theorem 6.1 to derive a closed-form expression for c_i, and therefore for the average consumed fidelity.

Lemma 6.1. *Consider a 1G1B system with $J(F,\rho_{\text{new}}) = aF+b$ and $F^{(0)}(t_0) = D_{t_0}(F_{\text{new}})$, where F_{new} is the fidelity of the state ρ_{new}. Then, the average fidelity after $i \geq 0$ purification*

rounds is given by

$$c_i = \frac{1}{4} + \left(F_{\text{new}} - \frac{1}{4}\right) \cdot a^i \gamma^{i+1} + \left(\frac{a}{4} + b - \frac{1}{4}\right) \gamma \frac{1 - a^i \gamma^i}{1 - a\gamma}, \tag{6.18}$$

where $\alpha = \mu + \lambda q$ *and* $\gamma = \alpha/(\alpha + \Gamma)$. *Moreover, the average consumed fidelity is given by*

$$\overline{F}_{\text{linear}} = \frac{\frac{1}{4}\Gamma + b\lambda q p + F_{\text{new}}\Big(\mu + \lambda q(1-p)\Big)}{\Gamma + \mu + \lambda q(1 - pa)}. \tag{6.19}$$

The closed-form solution (6.19) is obtainable since $\overline{F} = \frac{1}{A}\sum_{i=0}^{\infty} \pi_i c_i$ is a geometric series with the linear jump function, as can be seen from the form of π_i and c_i as found in Proposition 6.1 and Equation 6.18. In the following proposition, we see how $\overline{F}$ varies with p and q.

Proposition 6.3. *The quantity* $\overline{F}_{\text{linear}}$ *has the following properties:*

(a) $\overline{F}_{\text{linear}}$ *is a monotonic function of* q*;*

(b) $\overline{F}_{\text{linear}}$ *is a monotonic function of* p*;*

We provide a proof of Lemma 6.1 and Proposition 6.3 in Appendix 6.9.2. We now have closed-form expression for A and $\overline{F}_{\text{linear}}$, which allows for a thorough analysis of the performance of the 1G1B system with the linear jump function. In particular, the following conclusions may already be drawn.

- Result (*a*) from Proposition 6.3 implies that the average consumed fidelity is maximized for $q = 0$ or $q = 1$. Consider a 1G1B system with a fixed set of parameters and a pumping scheme with a linear jump function. If the pumping protocol is good enough (e.g., when $b \geq F_{\text{new}}(1 - a)$, as explained in Appendix 6.9.2), then pumping every time a link is generated ($q = 1$) maximizes the average consumed fidelity. Sometimes, the pumping protocol chosen may impact the average consumed fidelity negatively and in that case one should never pump entanglement ($q = 0$) to increase the average consumed fidelity.
- Result (*b*) from Proposition 6.3 provides similar insights: a pumping protocol with a good jump function always benefits from a larger probability of success, i.e., $\overline{F}_{\text{linear}}$ is maximized for $p = 1$. When the protocol is detrimental, failure ($p = 0$) benefits the overall procedure, since it frees the good memory and allows for a fresh entangled link to be allocated there.

When the jump function is good (i.e., when $\overline{F}_{\text{linear}}$ is monotonically increasing in q), we observe a trade-off between $\overline{F}_{\text{linear}}$ and the availability A, which is a decreasing function of q, as can be seen from (6.9). This behavior is shown in Figure 6.6. If we rarely purify (small q), a low-quality entangled state (small $\overline{F}_{\text{linear}}$) will be available most of the time (large A). In that case, the average consumed fidelity can be lower than the fidelity of newly generated links, since the entanglement is not being purified often enough to

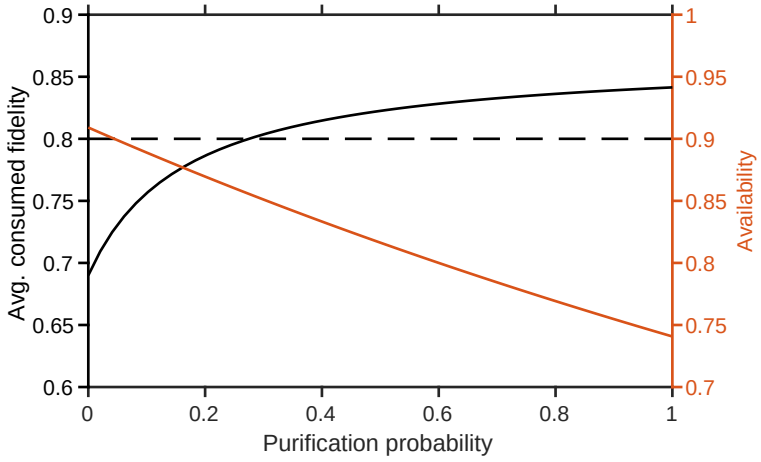

Figure 6.6: **Trade-off between average consumed fidelity and availability.** When the pumping is good enough (see discussion in main text), the average consumed fidelity $\overline{F}$ (black line) increases with increasing purification probability q, while the availability A (orange line) decreases. The dashed line corresponds to the fidelity of newly generated links ($F_{\text{new}} = 0.8$). Other parameters used in this example (times and rates in the same arbitrary units): $\lambda = 1$, $\mu = 0.1$, $p = 0.75$, $\Gamma = 1/40$, $J(F, \rho_{\text{new}}) = (1/3)F + (1 + F_{\text{new}})/3$, $\rho_{\text{new}} = F_{\text{new}} |\phi^+\rangle\langle\phi^+| + (1 - F_{\text{new}}) \left(|\psi^+\rangle\langle\psi^+| + |\psi^-\rangle\langle\psi^-| + |\phi^-\rangle\langle\phi^-| \right)/3$. This jump function corresponds to a linear approximation of a specific bilocal Clifford protocol (the DEJMPS protocol) in the high-fidelity regime [53].

compensate the noise introduced by the memory over time (in Figure 6.6, the average consumed fidelity is below the dashed line for small q). When purification is performed more often (larger q), the quality of the stored entanglement will be higher (larger $\overline{F}_{\text{linear}}$), at the expense of a more limited availability (smaller A), since purification can fail and destroy the entanglement stored in the long-term memory. This trade-off disappears when the pumping scheme is deterministic ($p = 1$): the availability remains constant when varying q since purification will always succeed and the stored entanglement will not be destroyed. Note that, if the system is dominated by decoherence ($\Gamma \gg \lambda, \mu$), the average consumed fidelity will always be smaller than F_0.

As a validation check, we also implemented a Monte Carlo simulation of the 1G1B system, which provided the same availability and average consumed fidelity that we obtained analytically (our code is available at https://github.com/AlvaroGI/buffering-1G1B; see also Appendix 6.11, where we compare simulation results to the bounds derived in the next section).

6.5.1. OPERATING REGIMES OF BILOCAL CLIFFORD PROTOCOLS

In this subsection, we study the operating regimes of the 1G1B system, under the assumption that the pumping protocol employed is a bilocal Clifford protocol [52, 95]. Firstly, we find upper and lower bounds for the availability. Then, for a desired value of the availability within these bounds, we find lower and upper bounds for the average consumed fidelity that can be provided by bilocal Clifford protocols. This analysis finds limits to the performance of the 1G1B buffering system.

Bilocal Clifford protocols are one of the most well-studied types of protocol [52, 68, 95]. One of their main advantages is that they are relatively simple to execute, since they involve a basic set of gates. To the best of our knowledge, bilocal Clifford circuits have been the only purification protocols implemented experimentally so far (see, e.g., [100, 204]). In Appendix 6.10 we provide further details on bilocal Clifford protocols.

Let us start our performance analysis by discussing the availability. The maximum value that can be achieved by any protocol (bilocal Clifford or not) is $\lambda/(\lambda+\mu)$, as can be seen from (6.9). This maximum value is obtained when there is no pumping or the pumping protocol succeeds deterministically, i.e., when $q=0$ or $p=1$. The availability is lower bounded by $\lambda/(2\lambda+\mu)$, and the minimum value is attained when a pumping protocol is always applied and it never succeeds, i.e., when $q=1$ and $p=0$.

To find bounds for the average consumed fidelity, we first need to bound the jump functions of all bilocal Clifford protocols, which we do in the following Lemma. We only consider nontrivial protocols, i.e., we do not consider bilocal Clifford protocols with $J(F,\rho_{\text{new}})=F$ or $J(F,\rho_{\text{new}})=F_{\text{new}}$, where F_{new} is the fidelity of ρ_{new}. The former trivial jump function corresponds to a protocol that leaves the buffered link untouched, while the second trivial jump function corresponds to a protocol that replaces the buffered link by the newly generated link.

Lemma 6.2. *Let $J(F,\rho_{\text{new}})$ be the jump function of a nontrivial bilocal Clifford protocol ($J(F,\rho_{\text{new}})\neq F$ and $J(F,\rho_{\text{new}})\neq F_{\text{new}}$, where F_{new} is the fidelity of ρ_{new}). Assume ρ_{new} is a Bell-diagonal state:*

$$\rho_{\text{new}} = F_{\text{new}}\left|\Phi^+\right\rangle\!\left\langle\Phi^+\right| + \lambda_1\left|\Psi^+\right\rangle\!\left\langle\Psi^+\right| + \lambda_2\left|\Psi^-\right\rangle\!\left\langle\Psi^-\right| + \lambda_3\left|\Phi^-\right\rangle\!\left\langle\Phi^-\right|, \tag{6.20}$$

with $F_{\text{new}}+\lambda_1+\lambda_2+\lambda_3=1$. Let us define F^* as

$$F^* = \frac{2F_{\text{new}}-1+\sqrt{(2F_{\text{new}}-1)^2-2\lambda_{\min}(1-2F_{\text{new}}-2\lambda_{\min})}}{2(2F_{\text{new}}-1+2\lambda_{\min})}, \tag{6.21}$$

where $\lambda_{\min}=\min(\lambda_1,\lambda_2,\lambda_3)$. Then, for all $F\in[\frac{1}{4},F^*]$, the jump function is lower bounded as follows:

$$a_{\text{l}}F+b_{\text{l}} \leq J(F,\rho_{\text{new}}) \tag{6.22}$$

where

$$a_{\text{l}} = \frac{2(4F^*-1)\left[2F_{\text{new}}-(F_{\text{new}}+\lambda_{\min})(F_{\text{new}}+\lambda_{\max})\right]+4(\lambda_{\max}-\lambda_{\min})(1-F^*)}{(4F^*-1)\left[(4F_{\text{new}}+4\lambda_{\max}-2)F^*+2-F_{\text{new}}-\lambda_{\max}\right]}, \text{ and}$$
$$b_{\text{l}} = \frac{F_{\text{new}}+\lambda_{\max}}{2}-\frac{a_{\text{l}}}{4}, \tag{6.23}$$

with $\lambda_{\max}=\max\{\lambda_1,\lambda_2,\lambda_3\}$. For $F\in[1/4,1]$, the jump function is upper bounded as

$$J(F,\rho_{\text{new}}) \leq a_{\text{u}}F+b_{\text{u}}, \tag{6.24}$$

with

$$a_{\text{u}} = \frac{4(1-F_{\text{new}})}{3}, \text{ and } b_{\text{u}} = \frac{4F_{\text{new}}-1}{3}. \tag{6.25}$$

Moreover, the success probability of the protocol is bounded by $p_{\mathrm{l}} \leq p \leq p_{\mathrm{u}}$*, where*

$$p_{\mathrm{l}} = \frac{1}{2}, \text{ and } \quad p_{\mathrm{u}} = F_{\mathrm{new}} + \max(\lambda_1, \lambda_2, \lambda_3). \tag{6.26}$$

A proof of Lemma 6.2 can be found in Appendix 6.10. We show this by considering propoerties of the jump functions of bilocal Clifford protocols, which may be found explicitly. Note that, despite the fact that we assume that newly generated entangled links are Bell-diagonal, other forms of the density matrix are also valid in practice, since they can be brought to Bell-diagonal form by adding extra noise [13, 86]. Note also that the lower bound for the jump function (6.22) only applies when the fidelity of the buffered link is below F^*, but this is always the case in the 1G1B system, as shown in Appendix 6.10.

If we regard F_{new} as a fixed parameter, the upper and lower bounds to the jump function (6.22) and (6.24) are linear in F, and the bounds to the success probability (6.26) are constant. It is now possible to find an upper and lower bound for the average consumed fidelity by combining Lemmas 6.1 and 6.2, as we do in the following corollary.

Corollary 6.1. *The average consumed fidelity of the 1G1B system when using any (nontrivial) bilocal Clifford protocol is lower bounded by*

$$\overline{F}_{\mathrm{l}} = \frac{\frac{1}{4}\Gamma + b_{\mathrm{l}}\lambda q p + F_{\mathrm{new}}\Big(\mu + \lambda q(1-p_{\mathrm{l}})\Big)}{\Gamma + \mu + \lambda q(1 - p_{\mathrm{l}} a_{\mathrm{l}})}, \tag{6.27}$$

and upper bounded by

$$\overline{F}_{\mathrm{u}} = \frac{\frac{1}{4}\Gamma + b_{\mathrm{u}}\lambda q p + F_{\mathrm{new}}\Big(\mu + \lambda q(1-p_{\mathrm{u}})\Big)}{\Gamma + \mu + \lambda q(1 - p_{\mathrm{u}} a_{\mathrm{u}})} \tag{6.28}$$

where a_{l}*,* b_{l}*,* p_{l}*,* a_{u}*,* b_{u}*, and* p_{u}*, are given by (6.23), (6.25), and (6.26).*

Now, we analyze the limits of the performance of the 1G1B system using the bounds on $\overline{F}$ from Corollary 6.1. Let us start with a 1G1B system with perfect memories, i.e., with $\Gamma = 0$. This corresponds to an ideal situation that we can use as a benchmark: once we introduce noise, the average consumed fidelity will be lower than in this ideal case. Figure 6.7(a) shows the achievable combinations of average consumed fidelity and availability for $F_{\mathrm{new}} = 0.8$, generation rate $\lambda = 1$, and consumption rate $\mu = 0.1$. Below, we list some important observations that may be drawn from this figure:

- The regions shaded in grey correspond to **unattainable values** of average fidelity and availability, and they apply to any pumping scheme (bilocal Clifford or not). The average consumed fidelity cannot be larger than the one provided by a hypothetical protocol with jump function $J(F, \rho_{\mathrm{new}}) = 1$ and probability of success $p = 1$, which is applied with probability $q = 1$ (however, such a protocol does not exist).
- The performance of a 1G1B system that uses any **bilocal Clifford protocol** is contained within the **region shaded in blue and yellow**. The yellow/blue line corresponds to a hypothetical protocol with jump function and success probability

saturating the lower/upper bounds from (6.22) and (6.26). For a fixed target availability, the blue line provides an upper bound on the maximum average consumed fidelity that can be achieved by using bilocal Clifford protocols. Here, we observe again the tradeoff between both performance metrics: if our target availability is very close to the maximum value, we cannot increase the average consumed fidelity beyond F_{new} (dotted line); but as we decrease the desired availability, we can achieve a higher consumed fidelity until we reach a maximum value.

- As a reference, we show the performance of the ***replacement protocol*** (red star): in such a protocol, every time a new link is generated in the bad memory, the link in the good memory is replaced by the new one, without any form of purification. The replacement protocol is not bilocal Clifford because success is always declared (in bilocal Clifford circuits, success depends on some measurement outcomes [52]). This simple protocol achieves maximum availability, given by $A = \lambda/(\lambda+\mu)$. However, since no purification is performed, this protocol cannot increase the fidelity above the initial value F_{new}. In the absence of decoherence, the replacement protocol is equivalent to applying no purification at all ($q = 0$).

In Figure 6.7(b), we perform a similar analysis for a 1G1B system in which the good memory has a finite lifetime, i.e., $\Gamma > 0$. This is a more realistic scenario. The following observations may be drawn from this figure:

- **Imperfect memories decrease the average consumed fidelity** but do not affect the availability. The availability is unaffected by the decoherence experienced by the entangled links, and therefore can take the same range of values as in Figure 6.7(a).

- The replacement protocol no longer provides an average fidelity F_{new}. Instead, the average fidelity is lower than F_{new} since the quality of the state stored in the good memory decreases over time and is never increased beyond F_{new} due to the absence of purification. However, the **replacement protocol performs better than no pumping** at all ($q = 0$). This is because the system can improve its fidelity every time a new link is produced, instead of waiting for a consumption event.

- In the presence of noise, the lower and upper bounds for bilocal Clifford protocols also shift towards lower values of average fidelity. Both the upper and lower bounds take their minimum value at $q = 0$. This means that, in the presence of noise, any pumping protocol will increase the average consumed fidelity, i.e., **any pumping ($q > 0$) is better than no pumping ($q = 0$)**, even if it succeeds with the lowest-possible probability. This is in contrast to when there is no noise (Figure 6.7(a)), where the lower bound takes its minimum at $q = 1$ and no such conclusion can be drawn. In fact, this conclusion (any pumping is better than no pumping) always applies when the amount of noise, Γ, is above the following threshold:

$$\Gamma > 4\mu p \frac{F_{\text{new}}(1-a)-b}{4F_{\text{new}}(1-p)+(4b+a)p-1}, \tag{6.29}$$

where a, b, and p are given by the choice of purification protocol (see (6.15)). In Appendix 6.9.3 we compute this threshold analytically.

As a final remark, in Appendix 6.11 we compare the bounds from Corollary 6.1 to simulations which do not assume the probability of successful purification to be constant. The simulation values lie within the bounds. This finding provides empirical evidence that the bounds are still useful when lifting the assumption about a constant probability of success.

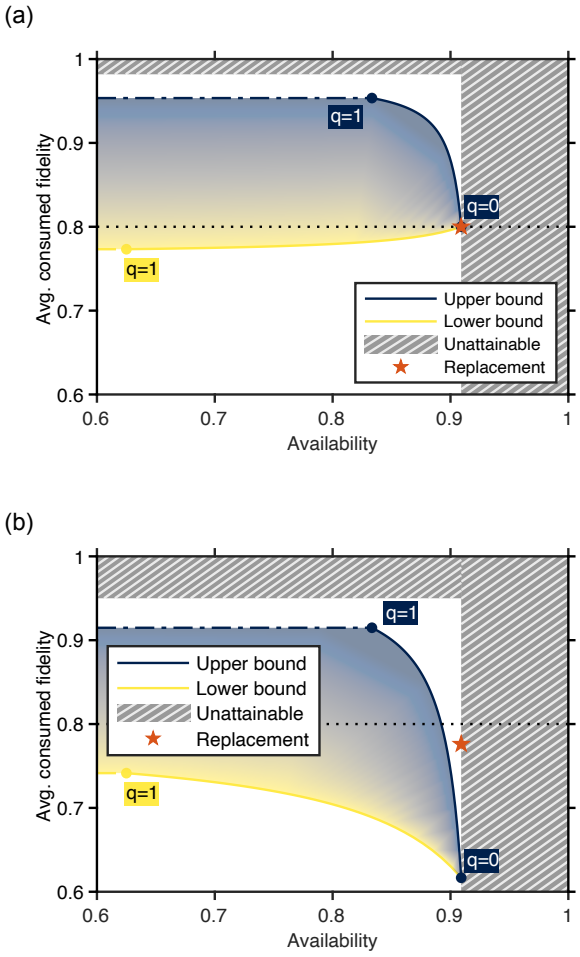

Figure 6.7: **Noise in the memories decreases the average consumed fidelity but does not affect the availability.** Bounds on the performance of a 1G1B system with a bilocal Clifford protocol, and with **(a)** noiseless memories ($\Gamma = 0$) or **(b)** noisy memories ($\Gamma = 5 \cdot 10^{-2}$ a.u.). For a given target availability, the average consumed fidelity is within the blue/yellow region (see Corollary 6.1). Availability is maximized for $q = 0$ (q is the probability of purification after successful entanglement generation), and it decreases for increasing q. White regions cannot be achieved by bilocal Clifford protocols. Striped regions cannot be achieved by any pumping protocol. Red star: performance of the replacement protocol (buffered link is replaced by new links). Dotted line: fidelity of newly generated entangled links. Parameters used in this example (times and rates in the same arbitrary units): $\lambda = 1$, $\mu = 0.1$, $F_{\text{new}} = 0.8$, $\rho_{\text{new}} = F_{\text{new}} \left|\phi^+\right\rangle\!\left\langle\phi^+\right| + (1 - F_{\text{new}}) \left(\left|\psi^+\right\rangle\!\left\langle\psi^+\right| + \left|\psi^-\right\rangle\!\left\langle\psi^-\right|\right)/2$.

6.6. CONCLUSIONS AND OUTLOOK

Our work sheds light on how to buffer high-quality entanglement shared among remote nodes in a quantum network. We have proposed two metrics to measure the performance of an entanglement buffering system: the availability and the average consumed fidelity. The availability corresponds to the fraction of time in which entanglement is available for consumption. The average consumed fidelity measures the quality of the entanglement upon consumption. We have used these performance metrics to analyze the 1G1B system, an entanglement buffering setup that uses two quantum memories per node. One of these memories has a finite lifetime and is used to buffer the entanglement, while the other memory is only used for entanglement generation. Entanglement generated in the bad memory can be used to pump the entanglement stored in the good memory. We have modelled the system as a continuous-time stochastic process and derived analytical expressions for both performance metrics. Our results confirm the intuition that, except in some edge cases, there is a trade-off between consuming entanglement at a high rate (high availability) and consuming high-quality entanglement (high average consumed fidelity). Remarkably, we found that, in a practical scenario (i.e., when the pumping protocol is bilocal Clifford and there is noise in the good memory), pumping the buffered entanglement is better than no pumping in terms of average consumed fidelity, even if the pumping has some probability of failure.

An assumption that allows us to find analytical solutions for our performance metrics is that the success probability of purification is constant over time. The model would be more realistic if the probability of successful purification was dependent on the state fidelity at that time, since this is the case for most protocols (in particular, the probability of successful purification is typically lower for input states with lower fidelity). This may mean that, realistically, the computation of the average fidelity when *conditioning* on successful purification may bias the system towards higher fidelity. However, we believe the comparison of our model (constant success probability) with a more realistic one incorporating this effect (success probability dependent on $F(t)$) to be beyond the scope of this work, since we expect this to greatly complicate the analysis of the problem.

Our proposed metrics can be used to evaluate the performance of other entanglement buffering systems. An interesting extension of this work would be to compare the performance of the 1G1B system to a bipartite entanglement buffering setup with n quantum memories per node. In such a system, one could employ more advanced purification protocols that consume more than two entangled states. We also expect that the mathematical framework developed in this work can be used to initiate the performance analysis of more complex systems. We leave this as future work.

6.7. [APPENDIX] GENERAL FORM OF JUMP FUNCTION

In this appendix, we explain the form (6.1) and (6.2) of the jump function and success probability for a general purification protocol, for two input states ρ_{W} and ρ_{new}, where

$$\rho_{\mathrm{W}} = F\left|\phi^+\right\rangle\!\left\langle\phi^+\right| + \frac{1-F}{3}\left|\psi^+\right\rangle\!\left\langle\psi^+\right| + \frac{1-F}{3}\left|\psi^-\right\rangle\!\left\langle\psi^-\right| + \frac{1-F}{3}\left|\phi^-\right\rangle\!\left\langle\phi^-\right|$$

is a Werner state and ρ_{new} is a general two-qubit state. Suppose that the purification protocol is described by a sequence of (possibly noisy) quantum operations that are described by a CPTP map Λ, and the final measurement outcome that signals success has measurement operator M_{succ}. From, e.g., Chapter 2.4 of [138], the output state is then given by

$$\rho' = \frac{M_{\text{succ}}\Lambda\left(\rho_{\text{W}} \otimes \rho_{\text{new}}\right) M_{\text{succ}}^\dagger}{p(F, \rho_{\text{new}})}, \tag{6.30}$$

where

$$p(F, \rho_{\text{new}}) = \text{Tr}\left[M_{\text{succ}}\Lambda\left(\rho_{\text{W}} \otimes \rho_{\text{new}}\right) M_{\text{succ}}^\dagger\right]. \tag{6.31}$$

We next rewrite the Werner state as

$$\begin{aligned}\rho_{\text{W}} &= F\left|\phi^+\right\rangle\!\left\langle\phi^+\right| + (1-F)\rho^\perp \\ &= \rho^\perp + F\left(\left|\phi^+\right\rangle\!\left\langle\phi^+\right| - \rho^\perp\right)\end{aligned}$$

where

$$\rho^\perp = \frac{1}{3}\left(\left|\psi^+\right\rangle\!\left\langle\psi^+\right| + \left|\psi^-\right\rangle\!\left\langle\psi^-\right| + \left|\phi^-\right\rangle\!\left\langle\phi^-\right|\right),$$

and p is the probability of success. We therefore have

$$\begin{aligned}M_{\text{succ}}\Lambda\left(\rho_{\text{W}} \otimes \rho_{\text{new}}\right) M_{\text{succ}}^\dagger = {} & M_{\text{succ}}\Lambda\left(\rho^\perp \otimes \rho_{\text{new}}\right) M_{\text{succ}}^\dagger \\ & + F \cdot M_{\text{succ}}\Lambda\left(\left(\left|\phi^+\right\rangle\!\left\langle\phi^+\right| - \rho^\perp\right) \otimes \rho_{\text{new}}\right) M_{\text{succ}}^\dagger,\end{aligned}$$

and taking the trace of the above yields

$$p(F, \rho_{\text{new}}) = d(\rho_{\text{new}}) + F \cdot c(\rho_{\text{new}}),$$

where c and d are obtained from the choice of purification protocol, i.e., from Λ and M_{succ}. Similarly, the output fidelity of upon success is given by

$$\left\langle\phi^+\right|\rho'\left|\phi^+\right\rangle = \frac{F \cdot \tilde{a}(\rho_{\text{new}}) + \tilde{b}(\rho_{\text{new}})}{p(F, \rho_{\text{new}})},$$

where

$$\begin{aligned}\tilde{a}(\rho_{\text{new}}) &= \left\langle\phi^+\right| M_{\text{succ}}\Lambda\left(\left(\left|\phi^+\right\rangle\!\left\langle\phi^+\right| - \rho^\perp\right) \otimes \rho_{\text{new}}\right) M_{\text{succ}}^\dagger \left|\phi^+\right\rangle, \\ \tilde{b}(\rho_{\text{new}}) &= \left\langle\phi^+\right| M_{\text{succ}}\Lambda\left(\rho^\perp \otimes \rho_{\text{new}}\right) M_{\text{succ}}^\dagger \left|\phi^+\right\rangle.\end{aligned}$$

This confirms the form (6.1) and (6.2) for the jump function and success probability.

6.8. [APPENDIX] FORMULAE FOR PERFORMANCE METRICS

In this appendix, we prove Proposition 6.1 and Theorem 6.1, which provide the formulae for our two performance metrics (availability and average consumed fidelity). First, in 6.8.1, we describe the stochastic process in the 1G1B setup in a simplified form and we provide some intermediate results that are necessary for the main proofs. Then, in 6.8.2, we employ the results from 6.8.1 to prove Proposition 6.1 and Theorem 6.1.

6.8.1. SIMPLIFIED 1G1B

We now only view the 1G1B system as taking one of two states: $\emptyset$ (no entangled link in memory G), or $\neg\emptyset$ (link in memory G). The system then alternates between these two states. For an illustration, see Figure 6.8.

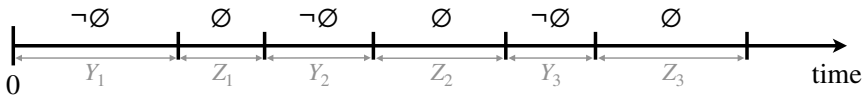

Figure 6.8: **The simplified 1G1B process.** The system alternates between the states $\neg\emptyset$ (link in memory G) and $\emptyset$ (no link in memory G). The system starts in $\neg\emptyset$. The times spent in $\neg\emptyset$ and $\emptyset$ are denoted by Y_i and Z_i, respectively.

More formally, the simplified 1G1B process is the following.

Definition 6.9 (Simplified 1G1B)**.** *Let $r(t) \in \{\emptyset, \neg\emptyset\}$ denote the state of simplified 1G1B at time t. Suppose that $r(0) = \neg\emptyset$, i.e., the system starts when there is a link in memory F. Let Y_1 be the time until this first link is removed, and let Z_1 be the time for which the system is empty until a fresh link is produced again. Let $\{Y_i\}_{i\geq 1}$ be the times spent in $\neg\emptyset$ until the link was removed from memory G (due to consumption or failed purification), and $\{Z_i\}_{i\geq 1}$ be the times which the system spent in $\emptyset$ until a link was produced. Then, according to our model of 1G1B, the Y_i are i.i.d. and exponentially distributed with rate $\beta := \mu + \lambda q(1-p)$, and the Z_i are i.i.d and exponentially distributed with rate λ.*

Recall that λ is the rate of generation of new entangled links, μ is the rate of consumption of links in memory G, q is the probability of immediately using new links for pumping, and p is the probability of successful pumping.

We will write the distribution functions as $F_Y(t) = P(Y_1 \leq t) = 1 - e^{-\beta t}$, and $F_Z(t) = P(Z_1 \leq t) = 1 - e^{-\lambda t}$. The process $X_i := Y_i + Z_i$ defines a *renewal process*, which we introduce with the following definition.

Definition 6.10. *A renewal process $\{N = N(t) : t \geq 0\}$ is a process such that*

$$N(t) = \max\{n : A_n \leq t\} \tag{6.32}$$

where $A_0 = 0$, $A_n = X_1 + \ldots + X_n$ for $n \geq 1$, and X_i is a sequence of i.i.d. and strictly positive random variables.

The value A_n is referred to as the nth *arrival time* of the process, and the values X_i are known as the *interarrival times*. From now on, we also use $A_0 = 0$, $A_n = X_1 + \ldots + X_n$ to denote the nth time at which a fresh link is produced, causing the system to move from $\emptyset$ into $\neg\emptyset$.

The *renewal function* is central to renewal theory, which we define below. Throughout, we use the convention $\mathrm{d}g(x) \equiv g'(x)\mathrm{d}x$ for differentiable functions g.

Definition 6.11. *Let $N(t)$ be a renewal process. Then, the renewal function is $m(t) := \mathbb{E}[N(t)]$.*

We will derive formulae for the availability and average consumed fidelity using this mathematical framework. An important result that we will use in order to do this is the renewal theorem, which we state below. This result assumes that the X_i are not *arithmetic*. If X_1 is arithmetic, this essentially means that X_1 only takes values in a set $\{mk : m = 0, \pm 1, ...\}$, with $k > 0$. For more details of arithmetic random variables, see Chapter 10 of [76].

Theorem 6.2 (Renewal Theorem/ Theorem 10.1.11 from [76])**.** *Consider a renewal process as given in Definition 6.10. Let F_X be the distribution function of the random variable X_1, where X_1 is not arithmetic. Let $H(t)$ be a bounded function. Consider solutions f to the renewal-type equation*

$$f(t) = H(t) + \int_0^t f(t-x)\mathrm{d}F_X(x). \tag{6.33}$$

Then, a solution is

$$f(t) = H(t) + \int_0^t H(t-x)\mathrm{d}m(x). \tag{6.34}$$

If H is bounded on finite intervals then f is bounded on finite intervals, and (6.34) is the unique solution of (6.33) with this property.

The renewal-type equation often arises when studying renewal processes, as we will see further on. The following result may be derived using Theorem 6.2, and is useful when taking the infinite limit.

Theorem 6.3 (Key renewal theorem/Theorem 11.2.7 from [76])**.** *If $g : [0,\infty) \to [0,\infty)$ is such that*

(a) $g(t) \geq 0$ for all t,

(b) $\int_0^\infty g(t)\mathrm{d}t < \infty$,

(c) g is a non-increasing function,

then

$$\lim_{t\to\infty} \int_0^t g(t-x)\mathrm{d}m(x) = \frac{1}{\mathbb{E}[X_1]} \int_0^\infty g(x)\mathrm{d}x,$$

whenever X_1 is not arithmetic.

We are now partially equipped to show the formulae for the availability and average fidelity. Next, we show a set of intermediate results that we will need for the main proofs.

Proposition 6.4. *Let $p(t) = P(r(t) = \neg\emptyset)$ be the probability that a link is available at time t in the simplified 1G1B process. Then,*

$$\lim_{t\to\infty} p(t) = \frac{\mathbb{E}(Y_1)}{\mathbb{E}(Y_1) + \mathbb{E}(Z_1)}. \tag{6.35}$$

Proof. We proceed by conditioning on the value of X_1. Now,

$$p(t) = P(r(t) = \neg\emptyset \cap X_1 > t) + P(r(t) = \neg\emptyset \cap X_1 < t). \tag{6.36}$$

Notice that the event $\{r(t) = \neg\emptyset \cap X_1 > t\}$ occurs if and only if $Y_1 > t$. Further, if $x < t$, then

$$P(r(t) = \neg\emptyset | X_1 = x) = p(t-x), \tag{6.37}$$

since the process starts afresh at time x. Then, (6.36) becomes

$$p(t) = 1 - F_Y(t) + \int_0^t p(t-x)\mathrm{d}F_X(x), \tag{6.38}$$

where $\mathrm{d}F_X(x) \equiv F'_X(x)\mathrm{d}x$. We now see that this is of the form (6.33) with $H(t) = 1 - F_Y(t)$, and so by Theorem 6.2,

$$p(t) = 1 - F_Y(t) + \int_0^t (1 - F_Y(t-x))\,\mathrm{d}m(x). \tag{6.39}$$

Taking the infinite limit,

$$\lim_{t\to\infty} p(t) = 1 - 1 + \lim_{t\to\infty} \int_0^t (1 - F_Y(t-x))\mathrm{d}m(x). \tag{6.40}$$

It can be seen that $H(t) = 1 - F_Y(t)$ satisfies the conditions (a)-(c) required by Theorem 6.3, so we may apply this Theorem to take the limit:

$$\lim_{t\to\infty} p(t) = \frac{1}{\mathbb{E}[X_1]} \int_0^\infty (1 - F_Y(x))\mathrm{d}x \tag{6.41}$$

$$= \frac{1}{\mathbb{E}[X_1]} \int_0^\infty P(Y_1 > x)\mathrm{d}x = \frac{\mathbb{E}[Y_1]}{\mathbb{E}[X_1]}. \tag{6.42}$$

Finally, using $\mathbb{E}[X_1] = \mathbb{E}[Y_1 + Z_1] = \mathbb{E}[Y_1] + \mathbb{E}[Z_1]$ suffices to show (6.35). □

Recall that the average fidelity of the system at a given time t is dependent on the time spent in each purification level leading up to this point. Therefore, in order to understand the average fidelity we first of all look at the *current lifetime* in this simplified setting.

Definition 6.12 (Current lifetime)**.** *Consider the simplified* 1G1B *system. Let $C(t)$ be the time spent so far in a state at time t. More formally,*

$$C(t) = \begin{cases} t - A_{N(t)}, & \text{if } r(t) = \neg\emptyset, \\ t - A_{N(t)} - Y_{N(t)+1}, & \text{if } r(t) = \emptyset. \end{cases} \tag{6.43}$$

The first case ($r(t) = \neg\emptyset$) is of most interest here, because it corresponds to when a link is in memory and is subject to decoherence. See Figure 6.9 for an illustration of this concept. In the following Lemma, we characterize the distribution of $C(t)$, conditional on being in the state $\neg\emptyset$.

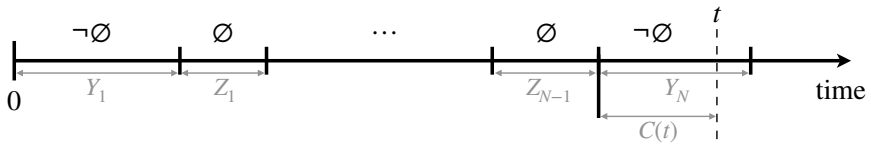

Figure 6.9: **Current lifetime of the simplified 1G1B process.** The random variable $C(t)$ denotes the time spent so far in the current state at time t. This is most interesting when $r(t) = \neg\emptyset$, because it tells us the age of a link in memory.

Lemma 6.3. *Consider the simplified 1G1B system. The limiting distribution of $C(t)$ conditional on there being a link is given by*

$$\lim_{t\to\infty} P(C(t) > x | r(t) = \neg\emptyset) = \frac{1}{\mathbb{E}[Y_1]} \int_x^\infty (1 - F_Y(s))\mathrm{d}s, \tag{6.44}$$

which is an exponential distribution with parameter β when $Y_1 \sim Exp(\beta)$.

Proof. Writing

$$P(C(t) > x | r(t) = \neg\emptyset) = \frac{P(C(t) > x \cap r(t) = \neg\emptyset)}{P(r(t) = \neg\emptyset)}, \tag{6.45}$$

we see that the bottom of the fraction has already been dealt with in Proposition 6.4. We therefore focus on

$$G(t,x) := P(C(t) > x \cap r(t) = \neg\emptyset). \tag{6.46}$$

Conditioning on X_1, we see that

$$G(t,x) = P(C(t) > x \cap r(t) = \neg\emptyset \cap X_1 > t) + P(C(t) > x \cap r(t) = \neg\emptyset \cap X_1 \le t). \tag{6.47}$$

Now, the event $\{C(t) > x \cap r(t) = \neg\emptyset \cap X_1 > t\}$ occurs if and only if $Y_1 > t > x$. Moreover, if $y < t$ then the process starts afresh from time y, and

$$P(C(t) > x \cap r(t) = \neg\emptyset | X_1 = y) = G(t-y). \tag{6.48}$$

Then, noting that $G(t,x) = 0$ for $t < x$, (6.47) becomes

$$G(t,x) = \mathbb{1}_{\{t\ge x\}}(1 - F_Y(t)) + \int_0^t G(t-y)\mathrm{d}F_X(y), \tag{6.49}$$

which is in the form of (6.33) with $H(t) = \mathbb{1}_{\{t\ge x\}}(1 - F_Y(t))$. Then, by Theorem 6.2, $G(t,x)$ is given by

$$G(t,x) = \mathbb{1}_{\{t\ge x\}}(1 - F_Y(t)) + \int_0^t \mathbb{1}_{\{t-y\ge x\}}(1 - F_Y(t-y))\mathrm{d}m(y) \tag{6.50}$$

which has limit

$$\lim_{t\to\infty} G(t,x) = 0 + \lim_{t\to\infty} \int_0^{t-x} (1 - F_Y(t-y))\mathrm{d}m(y) \tag{6.51}$$

$$= \lim_{s\to\infty} \int_0^s (1 - F_Y(s+x-y))\mathrm{d}m(y), \tag{6.52}$$

letting $s = t - x$. Then, noting that $g(s) = 1 - F_Y(s+x)$ satisfies conditions (a)-(c) of Theorem 6.3, we may apply this to find

$$\lim_{t\to\infty} G(t,x) = \frac{1}{\mathbb{E}[X_1]}\int_0^\infty g(s)\mathrm{d}s = \frac{1}{\mathbb{E}[X_1]}\int_0^\infty (1-F_Y(s+x))\mathrm{d}s \tag{6.53}$$

$$= \frac{1}{\mathbb{E}[X_1]}\int_x^\infty (1-F_Y(s))\mathrm{d}s. \tag{6.54}$$

From Proposition 6.4, we observe that

$$\mathbb{E}[X_1] = \frac{\mathbb{E}[Y_1]}{\lim_{t\to\infty} P\left(r(t) = \neg\emptyset\right)}.$$

We can use this to rewrite (6.54) as follows:

$$\lim_{t\to\infty} P\left(C(t) > x | r(t) = \neg\emptyset\right) = \frac{1}{\mathbb{E}[Y_1]}\int_x^\infty (1-F_Y(s))\mathrm{d}s, \tag{6.55}$$

which we notice is only dependent on the distribution of Y_1. In the case $Y_1 \sim \mathrm{Exp}(\beta)$, as considered in the 1G1B system,

$$\lim_{t\to\infty} P\left(C(t) > x | r(t) = \neg\emptyset\right) = \beta\int_x^\infty e^{-\beta s}\mathrm{d}s = e^{-\beta x}, \tag{6.56}$$

and so conditional on there being a link, the current lifetime approaches an exponential distribution. □

We have now characterized the availability (Proposition 6.4) and current lifetime (Lemma 6.3) for the simplified 1G1B system. However, note that both Proposition 6.4 and Lemma 6.3 assumed that the system starts in the state $r(0) = \neg\emptyset$. This was necessary in order to satisfy all of the conditions (a)-(c) of Theorem 6.3. The result below states that Theorem 6.3 still holds, even if the renewal process is *delayed*, which means that the first arrival has a different distribution to the others. For more details of delayed renewal processes, see [76] or [129].

Definition 6.13. *Let $\{X_i\}_{i\geq 1}$ be independent positive random variables such that $\{X_i\}_{i\geq 2}$ have the same distribution. Let $A_0 = 0$, $A_n = \sum_{i=1}^n X_i$, and $N^{\mathrm{d}} = \max\{n : A_n \leq t\}$. Then, $N^{\mathrm{d}}(t)$ is a delayed renewal process.*

Definition 6.14. *Let N^{d} be a delayed renewal process. Then, $m^{\mathrm{d}}(t) := \mathbb{E}[N^{\mathrm{d}}(t)]$ is the delayed renewal function.*

Theorem 6.4 (Key renewal theorem for delayed renewal processes/Theorem 1.20 of [129])**.** *Consider a delayed renewal process $N^{\mathrm{d}}(t)$. If $g : [0,\infty) \to [0,\infty)$ satisfies the same conditions (a)-(c) of Theorem 6.3, then*

$$\lim_{t\to\infty}\int_0^t g(t-x)\mathrm{d}m^{\mathrm{d}}(x) = \frac{1}{\mathbb{E}[X_2]}\int_0^\infty g(x)\mathrm{d}x. \tag{6.57}$$

A consequence of Theorem 6.4 is that even for delayed renewal processes, the limiting distribution is the same as for the non-delayed case. Therefore, the results of Proposition 6.4 and Lemma 6.3 hold even when the distribution of X_1 is not the same as $\{X_i\}_{i\geq 2}$. In particular, they still hold when the process starts in $\emptyset$. This is summarized with the following corollary.

Corollary 6.2. *Consider the simplified 1G1B process, now altered to start in $r(0) = \emptyset$. Let Z_0 be the time for which the system is empty until the first fresh link is produced. Let Y_1 be the time in which this link is present in memory until it is removed again, and so on. Let the probability of finding a link at time t be $p(t) = P(r(t) = \neg\emptyset)$. Then,*

$$\lim_{t\to\infty} p(t) = \frac{\mathbb{E}[Y_1]}{\mathbb{E}[Y_1]+\mathbb{E}[Z_1]} = \frac{\lambda}{\lambda+\beta}, \tag{6.58}$$

and the distribution of the current lifetime of a link satisfies

$$\lim_{t\to\infty} P\left(C(t) > x | r(t) = \neg\emptyset\right) = \frac{1}{\mathbb{E}[Y_1]}\int_x^\infty (1 - F_Y(s))\mathrm{d}s = e^{-\beta x}. \tag{6.59}$$

6

Recalling that $\beta = \mu + \lambda q(1-p)$, we see that the formula for the availability in Proposition 6.1 is already shown by (6.58).

6.8.2. AVAILABILITY AND AVERAGE CONSUMED FIDELITY IN 1G1B

Here, we compute the availability and the rest of the steady-state distribution of the 1G1B system (Proposition 6.1), as well as the average consumed fidelity (Theorem 6.1).

In order to calculate the average fidelity, we not only need the time spent in $\neg\emptyset$, but also the times spent in each pumping level leading up to the current one.

From 1G1B (Definition 6.1), one may define a simplified 1G1B system as

$$r(t) = \begin{cases} \neg\emptyset \text{ if } s(t) \geq 0 \\ \emptyset \text{ if } s(t) = \emptyset. \end{cases}$$

For the characterization of the fidelity of the link in memory at time t, $F(t)$, we are interested in the successful pumping attempts that occur in the the time interval $[A_{N(t)}, A_{N(t)} + C(t))$, where $C(t)$ is the current lifetime (Definition 6.12). In 1G1B, the successful pumping attempts are a Poisson process with rate $\delta := \lambda pq$. Since the rate is constant for all t, the number of successful pumping attempts within the interval $[A_{N(t)}, A_{N(t)} + C(t))$ has the identical distribution as the number of successful pumping attempts in the time interval $[0, C(t))$. From Corollary 6.2, we see that $C(t)$ converges in distribution to $C \sim \mathrm{Exp}(\beta)$. In the following Lemma, we characterize the number of successful pumping attempts that occur within the time C, and the time spent between each pair of consecutive pumping rounds. See Figure 6.10 for an illustration. An observation that we use below is that within the time interval $[0, C)$, the times at which pumping occurs form a separate renewal process, which is convenient for notation.

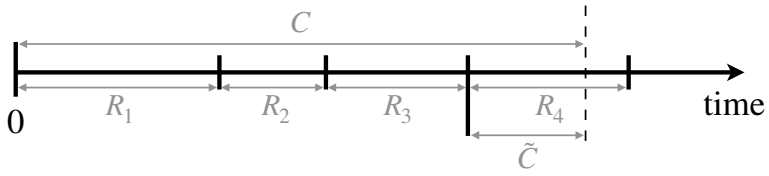

Figure 6.10: **Number of pumping rounds.** We are interested in the number of pumping rounds that have been carried out while a link is in memory. Here, C is the (limiting) distribution of the current lifetime in memory (see Figure 6.9), and R_i is the time between the $(i-1)$th and ith pumping round.

Lemma 6.4. *Consider a renewal process $\tilde{N}(t)$ with arrival times $S_0 = 0$, $S_n = \sum_{i=1}^{n} R_i$, with $R_1 \sim Exp(\delta)$. Let $C \sim Exp(\beta)$ be independent of the R_i. Let $M = N(C)$ be the number of arrivals that have occurred by time C. Let $\tilde{C} := C - S_M$ be the current lifetime at time C. Then,*

1. *The distribution of M is given by*

$$P(M \geq m) = \left(\frac{\delta}{\beta+\delta}\right)^m, \tag{6.60}$$

or equivalently

$$P(M = m) = \left(\frac{\delta}{\beta+\delta}\right)^m \left(\frac{\beta}{\beta+\delta}\right). \tag{6.61}$$

2. *Conditional on $M = m$, the random variables $(R_1, ..., R_m, \tilde{C})$ are mutually independent and identically distributed as $Exp(\beta+\delta)$.*

Proof. 1. We proceed by induction. Letting $F_R := P(R \leq x)$ We have

$$\begin{aligned} P(M \geq 1) = P(C > R_1) &= \int_0^\infty P(C > R_1 | R_1 = x) \mathrm{d}F_R(x) \\ &= \int_0^\infty e^{-\beta x} \cdot \delta e^{-\delta x} \mathrm{d}x = \frac{\delta}{\delta+\beta}, \end{aligned}$$

where we have used $P(C > R_1 | R_1 = x) = P(C > x) = e^{-\beta x}$ and $R_1 \sim \mathrm{Exp}(\delta)$. Then, assuming (6.60),

$$\begin{aligned} P(M \geq m+1) &= P(C > S_{m+1}) \\ &= P(C > R_{m+1} + S_m) \\ &\overset{a}{=} P(C > R_{m+1}) P(C > S_m), \\ &= P(C > R_1) P(M \geq m) \\ &\overset{b}{=} \left(\frac{\delta}{\beta+\delta}\right)\left(\frac{\delta}{\beta+\delta}\right)^m = \left(\frac{\delta}{\beta+\delta}\right)^{m+1}. \end{aligned}$$

In step (b), we have used the inductive assumption. In step (a) we have made use of the memoryless property of the exponential distribution: Since R_{m+1} and S_m are positive and independent random variables, this has as a consequence

$$\begin{aligned} P(C > R_{m+1} + S_m) &= \int_0^\infty \int_0^\infty \mathrm{d}F_R(r)\mathrm{d}F_{S_m}(s) P(C > r+s) \\ &= \int_0^\infty \int_0^\infty \mathrm{d}F_R(r)\mathrm{d}F_{S_m}(s) P(C > r) P(C > s) \\ &= P(C > R_{m+1}) P(C > S_m). \end{aligned} \tag{6.62}$$

Finally, (6.61) follows from

$$\begin{aligned} P(M = m) &= P(M \geq m) - P(M \geq m+1) \\ &= \left(\frac{\delta}{\beta+\delta}\right)^m - \left(\frac{\delta}{\beta+\delta}\right)^{m+1} \\ &= \left(\frac{\delta}{\beta+\delta}\right)^m \left(\frac{\beta}{\beta+\delta}\right). \end{aligned}$$

6

2. We firstly note that for any events E_1, E_2, E_3, it holds that

$$P(E_1 \cap E_2 \cap E_3) = P(E_1 \cap E_2) - P(E_1 \cap E_2 \cap \neg E_3), \tag{6.63}$$

where $\neg E$ denotes the complement of the event E. Now, consider the events

$$E_1 = \left\{R_i > x_i \ \forall i = 1, \ldots, m+1\right\}, \ E_2 = \left\{C \geq x_{m+1} + \sum_{i=1}^m R_i\right\}, \ E_3 = \left\{C < \sum_{i=1}^{m+1} R_i\right\}.$$

Now,

$$\begin{aligned} E_1 \cap E_2 \cap E_3 &= \left\{R_1 > x_1, \ldots, R_m > x_m, R_{m+1} > x_{m+1} \cap \sum_{i=1}^{m+1} R_i > C \geq x_{m+1} + \sum_{i=1}^m R_i\right\} \\ &\overset{a}{=} \left\{R_1 > x_1, \ldots, R_m > x_m, \tilde{C} > x_{m+1} \cap \sum_{i=1}^{m+1} R_i > C \geq \sum_{i=1}^m R_i\right\} \\ &\overset{b}{=} \left\{R_1 > x_1, \ldots, R_m > x_m, \tilde{C} > x_{m+1} \cap M = m\right\}, \end{aligned}$$

where in (a) we have used the definition of $\tilde{C}$, and in (b) we have used the definition of M. Then, by (6.63), we see that

$$\begin{aligned} &P\left(R_1 > x_1, \ldots, R_m > x_m, \tilde{C} > x_{m+1} \cap M = m\right) \\ &\qquad = P\left(R_1 > x_1, \ldots, R_{m+1} > x_{m+1} \cap C \geq x_{m+1} + \sum_{i=1}^m R_i\right) \\ &\qquad\qquad - P\left(R_1 > x_1, \ldots, R_{m+1} > x_{m+1} \cap C \geq \sum_{i=1}^{m+1} R_i\right). \end{aligned} \tag{6.64}$$

By the independence of the R_i, this is equivalent to

$$(6.64) = \left[P\left(C \geq x_{m+1} + \sum_{i=1}^{m} R_i \middle| R_1 > x_1, ..., R_{m+1} > x_{m+1} \right) - P\left(C \geq \sum_{i=1}^{m+1} R_i \middle| R_1 > x_1, ..., R_{m+1} > x_{m+1} \right) \right] \prod_{i=1}^{m+1} P(R_i > x_i), \quad (6.65)$$

and we now use the memoryless property of the exponential distribution (see the argument leading to (6.62)) to rewrite as

$$(6.64) = \left[P\left(C \geq x_{m+1} \middle| R_{m+1} > x_{m+1}\right) \prod_{i=1}^{m} P\left(C \geq R_i \middle| R_i > x_i\right) - \prod_{i=1}^{m+1} P\left(C \geq R_i \middle| R_i > x_i\right) \right] \prod_{i=1}^{m+1} P(R_i > x_i), \quad (6.66)$$

which, using that $P(C_i \geq R_i | R_i > x_i) P(R_i > x_i) = P(C_i \geq R_i > x_i)$, becomes

$$\begin{aligned} (6.64) &= \left[P(C \geq x_{m+1} \cap R_{m+1} > x_{m+1}) - P(C \geq R_{m+1} > x_{m+1})) \right] \prod_{i=1}^{m} P(C \geq R_i > x_i), \\ &= P(R_{m+1} > C \geq x_{m+1}) \prod_{i=1}^{m} P(C \geq R_i > x_i), \end{aligned}$$

where we have again made use of (6.63) to rewrite the factor on the left. Now,

$$\begin{aligned} P(C \geq R_1 > x_1) &= \int_{x_1}^{\infty} P(C \geq y) \mathrm{d}F_R(y) \\ &= \int_{x_1}^{\infty} e^{-\beta y} \cdot \delta e^{-\delta y} = \frac{\delta}{\beta + \delta} e^{-(\beta+\delta)x_1}, \end{aligned}$$

and by symmetry

$$P(R_1 \geq C > x_{m+1}) = \frac{\beta}{\beta + \delta} e^{-(\beta+\delta)x_{m+1}}.$$

We therefore see that

$$\begin{aligned} (6.64) &= \frac{\beta}{\beta + \delta} e^{-(\beta+\delta)x_{m+1}} \cdot \prod_{i=1}^{m} \left[\frac{\delta}{\beta + \delta} e^{-(\beta+\delta)x_i} \right] \\ &= \frac{\beta}{\beta + \delta} \cdot \left(\frac{\delta}{\beta + \delta} \right)^m \prod_{i=1}^{m+1} e^{-(\beta+\delta)x_i} = P(M = m) \prod_{i=1}^{m+1} e^{-(\beta+\delta)x_i}. \end{aligned}$$

It therefore follows that

$$P\left(R_1 > x_1, \ldots, R_m > x_m, \tilde{C} > x_{m+1} \middle| M = m\right) = \prod_{i=1}^{m+1} e^{-(\beta+\delta)x_i},$$

which suffices to show the second result.

□

Recalling that $C(t)$ converges in distribution to C, we now adapt Lemma 6.4 to apply to $C(t)$. In order to do this, we use the following result (for a proof, see Chapter 7 of [76]).

Theorem 6.5 (Continuous mapping theorem)**.** *Let $\{X_n\}$ be a sequence of random variables taking values in $\mathbb{R}^k$. If $X_n \to X$ in distribution as $n \to \infty$ and $g : \mathbb{R}^k \to \mathbb{R}^l$ is continuous, then $g(X_n) \to g(X)$ in distribution as $n \to \infty$.*

Corollary 6.3. *Suppose that $C(t)$ and X are independent random variables, and $C(t)$ converges in distribution to C as $t \to \infty$. Then,*

$$\lim_{t\to\infty} P(C(t) > X) = P(C > X). \tag{6.67}$$

Proof. Consider a sequence of times $\{t_n\}_{n\geq 1}$ such that $0 < t_1 < t_2 < ...$ and $\lim_{n\to\infty} t_n = \infty$. Let $C_n := C(t_n)$. Then, $C_n \to C$ in distribution. Moreover, since C_n and X are independent for all n, the pair $(C_n, -X) \to (C, -X)$ in distribution. Now, the function $g : \mathbb{R}^2 \to \mathbb{R}$, with $g(x, y) = x + y$ is continuous. Then, by Theorem 6.5, $C_n - X \to C - X$ in distribution, and so

$$\lim_{n\to\infty} P(C_n - X > 0) = \lim_{n\to\infty} P(C(t_n) - X > 0) = P(C - X > 0).$$

Since this is true for all such sequences $\{t_n\}$, the result follows. □

In the following corollary, we let the current lifetime be dependent on the parameter u to avoid confusion with the time of the renewal process (which is denoted as t).

Corollary 6.4. *Consider a renewal process $N(t)$ with arrival times $S_0 = 0$, $S_n = \sum_{i=1}^n R_i$, with $R_1 \sim Exp(\delta)$. Suppose that $C(u)$ converges in distribution to $C \sim Exp(\beta)$ as $u \to \infty$. Let $M(u) = N(C(u))$ be the number of arrivals that have occurred by time $C(u)$. Let $\tilde{C}(u) := C(u) - S_{M(u)}$ be the current lifetime at time $C(u)$. Then, the results of Lemma 6.4 still hold in the limit $u \to \infty$. In particular,*

1. *The limiting distribution of $M(u)$ is that of M,*

$$\lim_{u\to\infty} P(M(u) \geq m) = P(M \geq m) = \left(\frac{\delta}{\beta+\delta}\right)^m. \tag{6.68}$$

2. *Conditional on $M(u) = m$, the random variables $(R_1, ..., R_m, \tilde{C})$ converge in distribution to mutually independent and identically distributed $Exp(\beta + \delta)$ as $u \to \infty$, i.e.,*

$$\lim_{u\to\infty} P(X_1 > x_1, ..., X_m > x_m, \tilde{C}(u) > x_{m+1} | M(u) = m) = \prod_{i=1}^{m+1} e^{-(\beta+\delta)x_i}, \tag{6.69}$$

Proof. 1. Making use of Corollary 6.3, we have

$$\lim_{u\to\infty} P(M(u) \geq m) = \lim_{u\to\infty} P(C(u) > S_m) = P(C > S_m) = \left(\frac{\delta}{\beta+\delta}\right)^m.$$

2. One may use exactly the same arguments as were used to obtain (6.65), only replacing C with $C(u)$ and M with $M(u)$, to show that

$$P\left(R_1 > x_1, \ldots, R_m > x_m, \tilde{C}(u) > x_{m+1} \cap M(u) = m\right)$$
$$= \left[P\left(C(u) \geq x_{m+1} + \sum_{i=1}^{m} R_i \middle| R_1 > x_1, ..., R_{m+1} > x_{m+1} \right) \right.$$
$$\left. - P\left(C(u) \geq \sum_{i=1}^{m+1} R_i \middle| R_1 > x_1, ..., R_{m+1} > x_{m+1} \right) \right] \prod_{i=1}^{m+1} P\left(R_1 > x_i\right).$$

By Corollary 6.3, in the limit $u \to \infty$ this satisfies

$$\lim_{u\to\infty} P(R_1 > x_1, ..., R_m > x_m, \tilde{C}(u) > x_{m+1} \cap M(u) = m)$$
$$= \left[P\left(C \geq x_{m+1} + \sum_{i=1}^{m} R_i \middle| R_1 > x_1, ..., R_{m+1} > x_{m+1} \right) \right.$$
$$\left. - P\left(C \geq \sum_{i=1}^{m+1} R_i \middle| R_1 > x_1, ..., R_{m+1} > x_{m+1} \right) \right] \prod_{i=1}^{m+1} P\left(R_1 > x_i\right) = (6.65).$$

It then follows that

$$\lim_{u\to\infty} P(R_1 > x_1, ..., R_m > x_m, \tilde{C}(u) > x_{m+1} \middle| M(u) = m)$$
$$= \lim_{u\to\infty} \frac{P(R_1 > x_1, ..., R_m > x_m, \tilde{C}(u) > x_{m+1} \cap M(u) = m)}{P(M(u) = m)}$$
$$= \frac{P(R_1 > x_1, ..., R_m > x_m, \tilde{C} > x_{m+1} \cap M = m)}{P(M = m)} = \prod_{i=1}^{m+1} e^{-(\beta+\delta)x_i},$$

by Lemma 6.4.

□

For the case when $C(u)$ is the current lifetime of simplified 1G1B, the random variable $(R_1, \ldots, R_m, \tilde{C}(u))$ by definition has the same distribution as $\vec{T}(u)$. Recall that $\vec{T}(u)$ contains the times spent in each purification level leading up to the current one at time u in 1G1B (Definition 6.2). This leads to the following results.

Corollary 6.5. *Conditional on $s(t) = i$, $\vec{T}(t)$ converges in distribution to $(Q_0, \ldots, Q_i)$ as $t \to \infty$, where the Q_j are i.i.d. random variables with $Q_0 \sim \text{Exp}(\beta + \delta)$.*

We now continue with the formulae for the performance metrics. The availability in the 1G1B system was given in Proposition 6.1 and the average consumed fidelity was given in Theorem 6.1. Next, we prove both of them.

Proof of Proposition 6.1. From Corollary 6.2, we see that

$$A = \lim_{t\to\infty} P(s(t) = \neg\emptyset) = \frac{\lambda}{\lambda + \mu + \lambda q(1-p)}.$$

Further, for $i \geq 0$

$$P(s(t) = i) = P(s(t) = i | s(t) \neq \emptyset) \cdot P(s(t) \neq \emptyset).$$

Letting $C(t)$ denote the current lifetime of simplified 1G1B at time t, and $M(t)$ denote the number of purifications that have occurred within this time, by Corollary 6.4 it follows that

$$P(s(t) = i) = P(M(t) = i) \cdot P(s(t) \neq \emptyset) \to P(M = i) \cdot A$$

as $t \to \infty$. Recalling the distribution of M as found in Lemma 6.4, we obtain

$$\begin{aligned} \lim_{t\to\infty} P(s(t) = i) &= \left(\frac{\lambda q p}{\mu + \lambda q}\right)^i \cdot \frac{\mu + \lambda q(1-p)}{\mu + \lambda q} \cdot A \\ &= \frac{\lambda^{i+1} q^i p^i}{(\mu + \lambda q)^{i+1}} \cdot \frac{\mu + \lambda q(1-p)}{\lambda + \mu + \lambda q(1-p)}. \end{aligned}$$

We note that this result can also be derived with the global balance equations of a CTMC. Here, we chose to use the derivation with renewal theory since it offers a more general formula for the availability (see (6.58)) and ties in more neatly with the derivation of the formula for the average consumed fidelity, as we will see below. □

6

The following proposition will be helpful in the proof of Theorem 6.1 (formula for average consumed fidelity).

Proposition 6.5. *Let $\{p_i(t)\}_{i\geq 0}$ and $\{e_i(t)\}_{i\geq 0}$ be such that for all i, $\lim_{t\to\infty} p_i(t) = \pi_i$ and $\lim_{t\to\infty} e_i(t) = c_i$. Suppose also that for all t, $0 \leq e_i(t) \leq 1$, $0 \leq p_i(t) \leq 1$ and $\sum_{i=0} p_i(t) = 1$. Then*

$$\lim_{t\to\infty} \sum_{i=0}^{\infty} e_i(t) p_i(t) = \sum_{i=0}^{\infty} c_i \pi_i. \tag{6.70}$$

Proof of proposition 6.5. To show (6.70), it suffices to show that for any $\epsilon > 0$, there exists a T such that for all $t > T$,

$$\left| \sum_{i=0}^{\infty} e_i(t) p_i(t) - \sum_{i=0}^{\infty} c_i \pi_i \right| < \epsilon. \tag{6.71}$$

We firstly bound the sum using the triangle inequality,

$$\begin{aligned} \left| \sum_{i=0}^{\infty} e_i(t) p_i(t) - \sum_{i=0}^{\infty} c_i \pi_i \right| &= \left| \sum_{i=0}^{\infty} e_i(t)(p_i(t) - \pi_i) + (e_i(t) - c_i)\pi_i \right| \\ &\leq \underbrace{\sum_{i=0}^{\infty} e_i(t) |p_i(t) - \pi_i|}_{(A)} + \underbrace{\sum_{i=0}^{\infty} |e_i(t) - c_i| \pi_i}_{(B)}. \end{aligned} \tag{6.72}$$

We then show that $(A) \to 0$ and $(B) \to 0$ as $t \to \infty$. We firstly show that

$$\lim_{t\to\infty} \sum_{i=0}^{\infty} \left| p_i(t) - \pi_i \right| = 0. \tag{6.73}$$

Note that, since $\sum_{i=0}^{\infty} p_i(t) = 1$, it follows that $\sum_{i=0}^{\infty} \pi_i = 1$. Then, choose N such that

$$\sum_{i=0}^{N} \pi_i > 1 - \frac{\epsilon}{2}$$

and choose T_1 such that

$$\sum_{i=0}^{N} |p_i(t) - \pi_i| < \frac{\epsilon}{2}, \forall\, t > T_1$$

which is possible since the sum is finite. Then $\forall\, t > T_1$,

$$\begin{aligned} \left|1 - \sum_{i=0}^{N} p_i(t)\right| &= \left|1 - \sum_{i=0}^{N} \left(\pi_i - (\pi_i - p_i(t))\right)\right| \\ &< \left|1 - \sum_{i=0}^{N} \pi_i\right| + \sum_{i=0}^{N} |\pi_i - p_i(t)| < \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon. \end{aligned} \tag{6.74}$$

Now, choose T_2 such that $\forall\, t > T_2$,

$$|p_i(t) - \pi_i| < \frac{\epsilon}{N}, \forall\, i = 0, ..., N$$

and let $T = \max\{T_1, T_2\}$. Then, $\forall\, t > T$,

$$\begin{aligned} \sum_{i=0}^{\infty} |p_i(t) - \pi_i| &= \sum_{i=0}^{N} |p_i(t) - \pi_i| + \sum_{i>N} |p_i(t) - \pi_i| \\ &< N \cdot \frac{\epsilon}{N} + \sum_{i>N} p_i(t) + \sum_{i>N} \pi_i \\ &< \epsilon + \epsilon + \frac{\epsilon}{2}, \end{aligned}$$

from (6.74). This suffices to show (6.73). Combined with the fact that the e_i are bounded, it follows that $(A) \to 0$. We now show that $(B) \to 0$, i.e.,

$$\lim_{t \to \infty} \sum_{i=0}^{\infty} |e_i(t) - c_i| \pi_i = 0. \tag{6.75}$$

To show this, let $\epsilon > 0$. Choose N such that $\sum_{i=0}^{N} \pi_i > 1 - \epsilon$. Choose T such that

$$\sum_{i=0}^{N} |e_i(t) - c_i| < \epsilon, \ \forall\ t > T.$$

This is possible since the LHS is a finite sum. Then,

$$\begin{aligned} \sum_{i=0}^{\infty} |e_i(t) - c_i| \pi_i &= \sum_{i=0}^{N} |e_i(t) - c_i| \pi_i + \sum_{i>N} |e_i(t) - c_i| \pi_i \\ &< \left(\sum_{i=0}^{N} |e_i(t) - c_i|\right) \cdot \left(\sum_{i=0}^{N} \pi_i\right) + \sum_{i>N} \pi_i \\ &< \epsilon(1 - \epsilon) + \epsilon \ \forall\ t > T, \end{aligned}$$

which shows (6.75). □

Combining these results ($(A) \to 0$ and $(B) \to 0$) in (6.72)) suffices to show Proposition 6.5. We are now ready to prove Theorem 6.1 (formula for average consumed fidelity).

Proof of Theorem 6.1. We firstly expand out the average consumed fidelity (Definition 6.8) as a sum by conditioning on the value of $s(t)$,

$$\begin{aligned}\overline{F} := \mathbb{E}[F(t)|s(t) \neq \emptyset] &= \sum_{i=0}^{\infty} \mathbb{E}[F(t)|s(t)=i]P(s(t)=i|s(t)\neq\emptyset) \\ &= \frac{1}{P(s(t)\neq\emptyset)} \sum_{i=0}^{\infty} \mathbb{E}[F(t)|s(t)=i]P(s(t)=i)\,. \end{aligned} \tag{6.76}$$

Recall that we are interested in the limit $t \to \infty$ of the above. Note that from Proposition 6.1, we know the limiting values of $P(s(t) \neq \emptyset)$ and $P(s(t) = i)$. We now claim that

$$\lim_{t\to\infty} \mathbb{E}[F(t)|s(t)=i] = \mathbb{E}\left[F^{(i)}(Q_0, Q_1, ..., Q_i)\right] \tag{6.77}$$

where $Q_0, Q_1, \ldots, Q_i$ are i.i.d. random variables with $Q_0 \sim \text{Exp}(\mu + \lambda q)$, and $F^{(i)}$ is given in Definition 6.5. We use the following result:

Theorem 6.6 (Theorem 7.2.19 of [76])**.** *Let X_n be a sequence of random variables. Then, $X_n \to X$ in distribution if and only if $\mathbb{E}[g(X_n)] \to \mathbb{E}[g(X)]$ for all bounded continuous functions g.*

Recall that conditional on $s(t) = i$, we have $F(t) = F^{(i)}\left(\vec{T}(t)\right)$ (from Definition 6.6). As mentioned in Section 6.3, $F^{(i)}$ is a continuous and bounded function. Therefore, (6.77) follows by combining Theorem 6.6 and Corollary 6.5.

From Proposition 6.5, we therefore see that

$$\begin{aligned}\lim_{t\to\infty} \mathbb{E}[F(t)|s(t)\neq\emptyset] &= \lim_{t\to\infty} \frac{1}{P(s(t)\neq\emptyset)} \sum_{i=0}^{\infty} \mathbb{E}[F(t)|s(t)=i]P(s(t)=i) \\ &= \frac{1}{A} \sum_{i=0}^{\infty} \lim_{t\to\infty} \mathbb{E}[F(t)|s(t)=i] \lim_{t\to\infty} P(s(t)=i) \\ &= \frac{1}{A} \sum_{i=0}^{\infty} c_i \pi_i, \end{aligned}$$

where $c_i = \mathbb{E}\left[F^{(i)}(Q_0, Q_1, ..., Q_i)\right]$ and $\pi_i = \lim_{t\to\infty} P(s(t) = i)$. □

6.9. [APPENDIX] AVERAGE CONSUMED FIDELITY WITH A LINEAR JUMP FUNCTION

In this appendix we focus on linear jump functions. In 6.9.1, we provide bounds for the coefficients of a linear jump function. In 6.9.2, we first prove Proposition 6.2. Then, we use that Proposition to derive the average consumed fidelity in a 1G1B system that uses a pumping protocol with a linear jump function, which we denote by $\overline{F}_{\text{linear}}$ (i.e., we show Lemma 6.1). We also show that $\overline{F}_{\text{linear}}$ is monotonic in the probability of pumping q and

the probability of successful pumping p (Proposition 6.3). Lastly, in 6.9.3, we discuss in which situations $\overline{F}_{\text{linear}}$ is monotonically increasing in q, and we compute the noise threshold (6.29) discussed in Section 6.5.1 (above this threshold, any purification is better than no purification).

6.9.1. BOUNDS ON THE PARAMETERS OF A LINEAR JUMP FUNCTION

Proposition 6.6. *Consider a jump function that is linear with the fidelity of one of the input states, i.e.,*

$$J(F,\rho) = a(\rho)F + b(\rho), \tag{6.78}$$

where F is the fidelity of one of the input states and ρ is the second input state. Then, the coefficients $a(\rho)$ and $b(\rho)$ must satisfy

$$0 \leq a(\rho) \leq 1 \quad \text{and} \quad \frac{1}{4}\big(1-a(\rho)\big) \leq b(\rho) \leq 1-a(\rho).$$

Proof. First, we require $J(F,\rho) \leq 1$, which is equivalent to $b \leq 1-a$. We also require $J(F,\rho) \geq 1/4$, which leads to $b \geq (1-a)/4$. By imposing that the upper bound on b has to be larger than the lower bound, we find that $a \leq 1$. Finally, since we want jump functions that increase with increasing F, we want $a \geq 0$. □

6.9.2. DERIVATION OF AVERAGE CONSUMED FIDELITY WITH A LINEAR JUMP

Proof of Proposition 6.2. Here, we consider a 1G1B system with $J(F,\rho_{\text{new}}) = aF + b$ and $F^{(0)}(t_0) = D_{t_0}(F_{\text{new}})$, where F_{new} is the fidelity of the state ρ_{new}. Our goal is to find an analytical solution for the fidelity of the entangled link after i consecutive successful purifications, $F^{(i)}(t_0,...,t_{i-1},t_i)$. The time passed between purification j and $j+1$ is given by t_j. After the i-th purification the system spent time t_i without any transitions (i.e., no purification or consumption events). We show in this proof that $F^{(i)}$ is given by

$$F^{(i)}(t_0,...,t_{i-1},t_i) = \frac{1}{4} + \sum_{j=0}^{i} m_j^{(i)} e^{-\Gamma(t_j+t_{j+1}...+t_i)} \tag{6.79}$$

where the constants $m_j^{(i)}$ are given by $m_0^{(0)} = F_{\text{new}} - \frac{1}{4}$, and

$$m_j^{(i)} = \begin{cases} a^{i-j}\left(\frac{a}{4}+b-\frac{1}{4}\right), \text{ if } j>0, \\ a^i\left(F_{\text{new}}-\frac{1}{4}\right) \text{ if } j=0. \end{cases}$$

for $i > 0$.

We proceed by induction. For $i = 0$, we have

$$F^{(0)}(t_0) = D_{t_0}(F_{\text{new}}) = e^{-\Gamma t_0}\left(F_{\text{new}} - \frac{1}{4}\right) + \frac{1}{4}, \tag{6.80}$$

from which we see that $m_0^{(0)} = F_{\text{new}} - \frac{1}{4}$. If we assume that (6.79) is true for some i, using the recursive relation from (6.4) we can show that (6.79) is also true for $i+1$:

$$\begin{aligned} F^{(i+1)}(t_0,...,t_i) &= D_{t_{i+1}}\left(J(F^{(i)},\rho)\right) \\ &= D_{t_{i+1}}\left(aF^{(i)}+b\right) \\ &= e^{-\Gamma t_{i+1}}\left(aF^{(i)}+b-\frac{1}{4}\right)+\frac{1}{4} \\ &= \frac{1}{4}+\left(\frac{a}{4}+b-\frac{1}{4}\right)e^{-\Gamma t_{i+1}}+\sum_{j=0}^{i} am_j^{(i)}e^{-\Gamma(t_j+...+t_i+t_{i+1})}, \end{aligned}$$

from which it follows that

$$\begin{aligned} m_j^{(i+1)} &= am_j^{(i)} \ (0 \le j \le i) \\ m_{i+1}^{(i+1)} &= \frac{a}{4}+b-\frac{1}{4} \end{aligned}$$

Then, by the inductive assumption, $m_0^{(i+1)} = a^{i+1}\left(F_{\text{new}} - \frac{1}{4}\right)$, and $m_j^{(i+1)} = a^{i+1-j}\left(\frac{a}{4}+b-\frac{1}{4}\right)$ for $j > 0$. □

Proof of Lemma 6.1. Here we consider a 1G1B system with $J(F,\rho_{\text{new}}) = aF+b$ and $F^{(0)}(t_0) = D_{t_0}(F_{\text{new}})$, where F_{new} is the fidelity of the state ρ_{new}. Our goal is to find a closed-form solution for the average fidelity after $i \ge 0$ purification rounds, c_i, and for the average consumed fidelity, $\overline{F}_{\text{linear}}$.

We defined c_i as the average value of $F^{(i)}$ (see (6.13)). Using the expression for $F^{(i)}$ from Proposition 6.2 (also given in (6.79)), we can evaluate c_i as follows

$$\begin{aligned} c_i &:= \int_0^\infty \mathrm{d}t_i f_\alpha(t_i) ... \int_0^\infty \mathrm{d}t_0 f_\alpha(t_0) F^{(i)}(t_0,...,t_{i-1},t_i) \\ &= \int_0^\infty \mathrm{d}t_i f_\alpha(t_i) ... \int_0^\infty \mathrm{d}t_0 f_\alpha(t_0) \left[\frac{1}{4}+\sum_{j=0}^{i} m_j^{(i)} e^{-\Gamma(t_j+...+t_{i-1}+t_i)}\right] \\ &= \frac{1}{4}+\sum_{j=0}^{i} m_j^{(i)}\left(\frac{\alpha}{\alpha+\Gamma}\right)^{i-j+1} \\ &= \frac{1}{4}+\left(F_{\text{new}}-\frac{1}{4}\right)\cdot a^i\gamma^{i+1}+\gamma\left(\frac{a}{4}+b-\frac{1}{4}\right)\sum_{j=1}^{i} a^{i-j}\gamma^{i-j}, \end{aligned}$$

where $\alpha = \mu + \lambda q$, $f_\alpha(t_i) = \alpha e^{-\alpha t_i}$ (since the times t_i are exponentially distributed with rate α), $\gamma = \alpha/(\alpha+\Gamma)$. Using the fact that this is a geometric series, we may now obtain a closed-form solution for c_i:

$$c_i = \frac{1}{4}+\left(F_{\text{new}}-\frac{1}{4}\right)\cdot a^i\gamma^{i+1}+\gamma\left(\frac{a}{4}+b-\frac{1}{4}\right)\frac{1-a^i\gamma^i}{1-a\gamma}. \tag{6.81}$$

The final formula for the average fidelity may then be computed with the results of Proposition 6.1 and Theorem 6.1 as

$$\begin{aligned}\overline{F}_{\text{linear}} &= \lim_{t\to\infty} \mathbb{E}\left(F(t)|s(t)\neq\emptyset\right) = \frac{1}{1-\pi_\emptyset}\sum_{i=0}^{\infty} c_i\pi_i \\ &= \frac{1}{4} + \frac{\gamma}{1-a\gamma}\cdot\left(\frac{a}{4}+b-\frac{1}{4}\right) + \frac{\gamma}{(1-\pi_\emptyset)}\cdot\left(F_{\text{new}} - \frac{1}{4} - \frac{\frac{a}{4}+b-\frac{1}{4}}{1-a\gamma}\right)\sum_{i=0}^{\infty}\pi_i(a\gamma)^i,\end{aligned} \tag{6.82}$$

where the constant terms are no longer in the sum since

$$\frac{1}{1-\pi_\emptyset}\sum_{i=0}^{\infty}\pi_i = 1,$$

by the normalization of the steady state distribution. Recalling the distribution of π from Proposition 6.1, we may evaluate the sum as a geometric series,

$$\begin{aligned}\sum_{i=0}^{\infty}\pi_i(a\gamma)^i &= \frac{\lambda}{\mu+\lambda q}\sum_{i=0}^{\infty}\left(\frac{\lambda q p a\gamma}{\mu+\lambda q}\right)^i\pi_\emptyset \\ &= \frac{\lambda}{\mu+\lambda q}\cdot\frac{1}{1-\frac{\lambda q p a\gamma}{\mu+\lambda q}}\pi_\emptyset \\ &= \frac{\lambda}{\mu+\lambda q-\lambda q p a\gamma}\pi_\emptyset.\end{aligned}$$

We may now substitute this into (6.82) to obtain a closed-form solution for the average fidelity,

$$\begin{aligned}\overline{F}_{\text{linear}} &= \frac{1}{4} + \frac{\gamma}{1-a\gamma}\cdot\left(\frac{a}{4}+b-\frac{1}{4}\right) + \gamma\cdot\left(F_{\text{new}} - \frac{1}{4} - \frac{\frac{a}{4}+b-\frac{1}{4}}{1-a\gamma}\right)\frac{\lambda}{\mu+\lambda q-\lambda q p a\gamma}\frac{\pi_\emptyset}{1-\pi_\emptyset} \\ &= \frac{1}{4} + \frac{\gamma}{1-a\gamma}\cdot\left(\frac{a}{4}+b-\frac{1}{4}\right) + \gamma\cdot\left(F_{\text{new}} - \frac{1}{4} - \frac{\frac{a}{4}+b-\frac{1}{4}}{1-a\gamma}\right)\frac{\mu+\lambda q(1-p)}{\mu+\lambda q-\lambda q p a\gamma} \\ &= \frac{\frac{1}{4}\Gamma + b\lambda q p + F_{\text{new}}\left(\mu+\lambda q(1-p)\right)}{\Gamma+\mu+\lambda q(1-pa)},\end{aligned} \tag{6.83}$$

which completes the closed-form solutions for our two performance metrics in this set-up (in the last step we used Mathematica to simplify the expression). □

Proof of Proposition 6.3. To show (a), we compute the partial derivative of the average consumed fidelity with respect to q:

$$\frac{\partial\overline{F}_{\text{linear}}}{\partial q} = \lambda\frac{\Gamma\left(4F_{\text{new}}(1-p)+(4b+a)p-1\right)+4\mu p\left(b-F_{\text{new}}(1-a)\right)}{4\left(\Gamma+\mu+\lambda q(1-ap)\right)^2}. \tag{6.84}$$

Since the sign of the derivative does not depend on q, we conclude that $\overline{F}_{\text{linear}}$ is monotonic in q.

To show (b), we proceed similarly:

$$\frac{\partial \overline{F}_{\text{linear}}}{\partial p} = \lambda q \frac{4(b - F_{\text{new}})(\Gamma + \mu + \lambda q) + a\big(\Gamma + 4F_{\text{new}}(\mu + \lambda q)\big)}{4\big(\Gamma + \mu + \lambda q(1 - ap)\big)^2}. \tag{6.85}$$

Since the sign of this derivative does not depend on p, we conclude that $\overline{F}_{\text{linear}}$ is monotonic in p.

□

6.9.3. NOISE THRESHOLD

In the previous section, we showed that $\overline{F}_{\text{linear}}$ is monotonic in q and p (Proposition 6.3). Nevertheless, note that $\overline{F}_{\text{linear}}$ can be monotonically increasing or decreasing in q and in p depending on the values of the other parameters. For a pumping protocol with a good enough jump function, $\overline{F}_{\text{linear}}$ becomes increasing in q. A sufficient condition is for the jump function to satisfy $b \geq F_{\text{new}}(1 - a)$, as we show next. The partial derivative with respect to q from (6.84) can be written as follows:

$$\frac{\partial \overline{F}_{\text{linear}}}{\partial q} = \frac{\lambda}{x^2}\big(\Gamma y + 4\mu p z\big), \tag{6.86}$$

where $x = 2(\Gamma + \mu + \lambda q(1 - ap))$, $y = 4F_{\text{new}}(1 - p) + (4b + a)p - 1$, and $z = b - F_{\text{new}}(1 - a)$. Using the fact that $b \geq (1 - a)/4$, we find that $y \geq 0$. A sufficient condition for the partial derivative to be positive is that $z \geq 0$, i.e., if $b \geq F_{\text{new}}(1 - a)$, then the average consumed fidelity is monotonically increasing in q. Moreover, we can conclude that, if the noise is above certain threshold ($\Gamma > -4\mu p z/y$), the derivative is positive and the pumping is always beneficial, even if it succeeds with a very small probability.

6.10. [APPENDIX] BOUNDS FOR BILOCAL CLIFFORD PROTOCOLS

In this appendix, we find bounds to the output fidelity and the probability of success of 2-to-1 purification protocols. In particular, we show Lemma 6.2, where upper and lower bounds on the jump function and the success probability of any bilocal Clifford protocol, taking as input a Werner state ρ_{W} and a Bell-diagonal state ρ_{BD}. We define the fidelity of a state ρ as $F(\rho, |\phi^+\rangle) = \langle\phi^+| \rho |\phi^+\rangle$, where $|\phi^+\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$ is one of the Bell states. We find the bounds for a system with the following restrictions.

- We consider 2-to-1 purification protocols, i.e., protocols that take two bipartite entangled states as input and output a single bipartite state. This allows us to use these bounds directly for the analysis of the 1G1B system.
- We restrict the pumping protocols to bilocal Clifford protocols [52, 95], which are a well-known type of purification scheme. We provide more details about this type of protocol in Section 6.10.1.

- We assume that one of the input states is a Werner state (in the 1G1B system, this is the state in the good memory, which suffers from depolarizing noise) and the other input state is Bell-diagonal (in the 1G1B system, this is the state generated via heralded entanglement generation and placed in the bad memory). Mathematically, the input states can be written, respectively, as

$$\rho_W = F\left|\phi^+\right\rangle\!\left\langle\phi^+\right| + \frac{1-F}{3}\left|\psi^+\right\rangle\!\left\langle\psi^+\right| + \frac{1-F}{3}\left|\psi^-\right\rangle\!\left\langle\psi^-\right| + \frac{1-F}{3}\left|\phi^-\right\rangle\!\left\langle\phi^-\right|,$$

$$\rho_{BD} = F_{BD}\left|\phi^+\right\rangle\!\left\langle\phi^+\right| + \lambda_1\left|\psi^+\right\rangle\!\left\langle\psi^+\right| + \lambda_2\left|\psi^-\right\rangle\!\left\langle\psi^-\right| + \lambda_3\left|\phi^-\right\rangle\!\left\langle\phi^-\right|,$$

  with $F, F_{BD}, \lambda_1, \lambda_2, \lambda_3 \in [0,1]$ subjected to the normalization constraint $F_{BD} + \lambda_1 + \lambda_2 + \lambda_3 = 1$, and with the Bell states defined as

$$\left|\phi^+\right\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}},\ \left|\psi^+\right\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}},\ \left|\psi^-\right\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}},\ \left|\phi^-\right\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}}.$$

  Note that any bipartite state can be brought to Bell-diagonal form while preserving the fidelity by means of twirling (adding extra noise) [13, 86].

- We only consider newly generated states with fidelity to some Bell state larger than $1/2$, i.e., we assume $F_{BD} > 1/2$ (note that $F_{BD} > 1/2$ is equivalent to $\lambda_i > 1/2$ for some i, since the states are equivalent upon some Pauli corrections). As shown in 6.10.3, this is a necessary and sufficient condition for the existence of entanglement (otherwise, the state is not useful for purification).

- We assume the Werner state has fidelity $F > 1/4$, since the good memory is initially occupied with a state with fidelity larger than $1/2$, and this fidelity can decay at most to $1/4$ due to depolarizing noise (see Definition 6.3).

In Section 6.10.1, we provide a formal definition of bilocal Clifford protocols. Then, in Section 6.10.2, we prove Lemma 6.2, where bounds are found for the jump function and success probability of bilocal Clifford protocols in a system with the above restrictions.

6.10.1. BILOCAL CLIFFORD PROTOCOLS

Bilocal Clifford protocols [52, 95] take n bipartite states as input and outputs a single bipartite state. They consist of the following steps:

1. $C^T \otimes C^\dagger$ is applied to the state, where C is some Clifford circuit. A Clifford circuit consists of Hadamard gates, phase gates S, and CNOTs [70, 71]. If the state is held by two separate parties, one of them applies C^T and the other one applies $C^\dagger$.

2. All of the qubit pairs except one are measured (in a 2-to-1 protocol, one qubit pair is measured and the other one is kept).

3. Depending on the parity of the measurement outcomes, success or failure is declared. Local unitaries may be performed after a success.

One of the main advantages of bilocal Clifford protocols is that they are relatively simple to execute in practice, since they involve a basic set of gates. Additionally, any stabilizer code C can be mapped to a bilocal Clifford circuit that applies $C^T \otimes C^\dagger$, allowing the analysis of bilocal Clifford circuits from a quantum error-correction perspective [68]. This type of protocol also includes well-known purification protocols, such as DEJMPS [53].

6.10.2. LINEAR BOUNDS FOR BILOCAL CLIFFORD PROTOCOLS

In this appendix, we prove Lemma 6.2, where bounds on the jump function and success probability of every bilocal Clifford protocol are found. Consider pumping two states of the form

$$\rho_W = F\left|\Phi^+\right\rangle\left\langle\Phi^+\right| + \frac{1-F}{3}\left(\left|\Psi^+\right\rangle\left\langle\Psi^+\right| + \left|\Psi^-\right\rangle\left\langle\Psi^-\right| + \left|\Phi^-\right\rangle\left\langle\Psi^-\right|\right) \tag{6.87}$$

$$\rho_{BD} = F_{BD}\left|\Phi^+\right\rangle\left\langle\Phi^+\right| + \lambda_1\left|\Psi^+\right\rangle\left\langle\Psi^+\right| + \lambda_2\left|\Psi^-\right\rangle\left\langle\Psi^-\right| + \lambda_3\left|\Phi^-\right\rangle\left\langle\Psi^-\right|. \tag{6.88}$$

Using the methods from [95], we can find the analytical expressions for the output fidelity and success probability for every bilocal Clifford protocol. The restriction to bilocal Clifford protocols and Bell-diagonal states allows us to do this enumeration of analytical functions efficiently [68, 95]. There are only seven protocols that provide a unique combination of J and p, as shown in Table 6.2. We refer to the i-th jump function and success probability as $J_i(F,\rho_{BD})$ and $p_i(F,\rho_{BD})$, for $i = 1,...,7$.

Table 6.2: **Jump function and success probability for all 2-1 bilocal Clifford protocols**, with input states given in (6.87) and (6.88) .

Protocol	Jump function	Success probability
1	$\frac{(4\lambda_1+3\lambda_2+3\lambda_3-3)F-\lambda_1}{(4\lambda_2+4\lambda_3-2)F-\lambda_2-\lambda_3-1}$	$\frac{2}{3}(1-2\lambda_2-2\lambda_3)F+\frac{1}{3}(1+\lambda_2+\lambda_3)$
2	$\frac{(3\lambda_1+4\lambda_2+3\lambda_3-3)F-\lambda_2}{(4\lambda_1+4\lambda_3-2)F-\lambda_1-\lambda_3-1}$	$\frac{2}{3}(1-2\lambda_3-2\lambda_1)F+\frac{1}{3}(1+\lambda_3+\lambda_1)$
3	$\frac{(3\lambda_1+3\lambda_2+4\lambda_3-3)F-\lambda_3}{(4\lambda_1+4\lambda_2-2)F-\lambda_1-\lambda_2-1}$	$\frac{2}{3}(1-2\lambda_1-2\lambda_2)F+\frac{1}{3}(1+\lambda_1+\lambda_2)$
4	F	$F_{BD}+\lambda_1$
5	F	$F_{BD}+\lambda_2$
6	F	$F_{BD}+\lambda_3$
7	F_{BD}	$\frac{2}{3}F+\frac{1}{3}$

We see that for these particular input states, J_4, J_5 and J_6 produce no change in the fidelity of ρ_W. They also have a non-unity success probability. It would therefore be advantageous to simply perform no action instead of attempting Protocols 4-6. Similarly, J_7 assumes the fidelity of the Bell-diagonal state, which is the same change as performing replacement. Since replacement can be achieved with probability one, it does not make sense to perform Protocol 7. Therefore, the only remaining 'non-trivial' protocols are Protocols 1-3. In the following, we therefore find bounds for the jump function of Protocols 1-3. Notice that there is symmetry in the λ_i: J_2 and p_2 can be obtained by permuting $(\lambda_1,\lambda_2,\lambda_3)$ in J_1 and p_1, and similarly for J_3 and p_3.

In the following, we show Lemma 6.2.

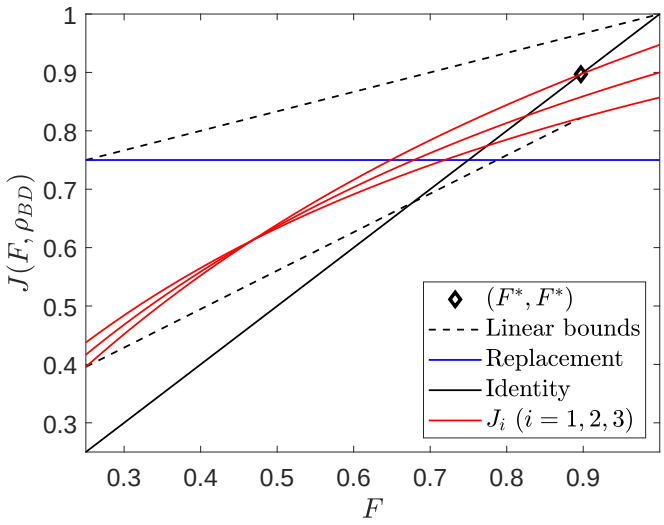

Figure 6.11: **Linear bounds for the jump function of bilocal Clifford protocols (black dashed lines).** The jump functions shown are J_1-J_3 (red lines), J_4-J_6 (identity operation, black line), and J_7 (probabilistic replacement, blue line). F^* is the highest fidelity achievable by pumping a low-fidelity Werner state with the fixed Bell-diagonal state ρ_{BD}. The lower bound holds in the range $[1/4, F^*]$. The upper bound holds in the range $[1/4, 1]$. Here, $F_{\text{BD}} = 0.75$ and $\rho_{\text{BD}} = (0.75, 0.125, 0.833, 0.0417)$.

6

Proof of Lemma 6.2. We firstly show the linear lower bound (i.e., the formulae given in (6.23)). We assume that $\lambda_1 \geq \lambda_2 \geq \lambda_3$. Then, by symmetry in the λ_i, one may retrieve the bound by setting $\lambda_{\text{min}} = \lambda_3$ and $\lambda_{\text{max}} = \lambda_1$. In order to show this bound, we make use of the following collection of results. It is important to note that when showing all of the following results, ρ_{BD} is fixed.

1. **Proposition 6.7, Corollory 6.6, Proposition 6.8** – the formula for F^* is derived (Equation 6.21). This is the maximum achievable fidelity achievable in the 1G1B system, with fixed Bell-diagonal input state ρ_{BD}. Therefore, at any given time t, the fidelity $F(t)$ of the stored link in the 1G1B system (see Definition 6.6) satisfies $F(t) \leq F^*$.

2. At $F = F^*$, Protocol 3 provides the best output fidelity,

$$J_1\left(F^*, \rho_{\text{BD}}\right) \leq J_2\left(F^*, \rho_{\text{BD}}\right) \leq J_3\left(F^*, \rho_{\text{BD}}\right)$$

   (**Proposition 6.7 and Corollary 6.6**).

3. At $F = 1/4$, Protocol 1 provides the best output fidelity, i.e.,

$$J_3\left(F^*, \rho_{\text{BD}}\right) \leq J_2\left(F^*, \rho_{\text{BD}}\right) \leq J_1\left(F^*, \rho_{\text{BD}}\right),$$

   since $J_i(1/4, \rho_{\text{BD}}) = (F_{\text{BD}} + \lambda_i)/2$.

4. For $i = 1, 2, 3$, $J_i(F, \rho_{\text{BD}})$ is a concave function of F (**Proposition 6.9**).

In particular, the third result means that any straight line taken between two points on J_i must lie below the curve itself. The linear lower bound is the linear function connecting the points

$$\left(F^*, J_1(F^*, \rho_{\text{BD}})\right), \quad \left(\frac{1}{4}, J_3\left(\frac{1}{4}, \rho_{\text{BD}}\right)\right), \tag{6.89}$$

which is given by

$$J_{\text{LB}}(F, \rho_{\text{BD}}) = \left(\frac{J_1(F^*, \rho_{\text{BD}}) - \frac{F_{\text{BD}}+\lambda_3}{2}}{F^* - \frac{1}{4}}\right)\left(F - \frac{1}{4}\right) + \frac{F_{\text{BD}} + \lambda_3}{2},$$

where we have used the fact that $J_i(1/4, \rho_{\text{BD}}) = (F_{\text{BD}} + \lambda_i)/2$. Letting $\lambda_{\text{max}} = \lambda_1$ and $\lambda_{\text{min}} = \lambda_3$, this may be rearranged into the form

$$J_{\text{LB}}(F, \rho_{\text{new}}) = a_{\text{l}} F + b_{\text{l}},$$

with a_{l} and b_{l} given in Lemma 6.2 (in (6.23)). When choosing the points in (6.89), we are joining the line corresponding to the lowest of the J_i for both $F = 1/4$ and $F = F^*$. By the concavity property, this is therefore a lower bound for all of the J_i in the region $[1/4, F^*]$. See Figure 6.11 for an illustration of this lower bound.

We now show the upper bound. We choose this to be the linear function connecting the points $(1/4, F_{\text{BD}})$ and $(1, 1)$, which is given by

$$J_{\text{UB}}(F, \rho_{\text{new}}) = \left(\frac{1 - F_{\text{BD}}}{1 - \frac{1}{4}}\right)\left(F - \frac{1}{4}\right) + F_{\text{BD}},$$

and may be rearranged into the form

$$J_{\text{UB}}(F, \rho_{\text{new}}) = a_{\text{u}} F + b_{\text{u}},$$

with a_{u} and b_{u} given in Lemma 6.2 (in (6.25)). We show that this is an upper bound with the following steps. Again, for ease of notation, we exploit the symmetry in λ_i and assume that $\lambda_1 \geq \lambda_2 \geq \lambda_3$.

1. In the domain $F > 0$, the jump functions J_1, J_2 and J_3 intersect at the same point F_{int}. Moreover, for $i = 1, 2, 3$, $J_i(F_{\text{int}}, \rho_{\text{BD}}) = \sqrt{\frac{F_{\text{BD}}}{2}} < F_{\text{BD}}$. (**Proposition 6.7**).

2. In the domain $F \in [F_{\text{int}}, 1]$, the jump function outputting the highest-fidelity outcome out of protocols 1-3 is J_3 (**Corollary 6.6**).

3. For $i = 1, 2, 3$, J_i is an increasing and concave function of F (**Proposition 6.9**).

4. Consider the tangent to J_3 at $F = 1$. This lies below J_{UB} in the range $F \in [1/4, 1]$ (**Proposition 6.10**).

By result (3) from the above list (concavity), we see that the tangent to J_3 at $F = 1$ upper bounds J_3 for all F. By result (2) from the above list, this also upper bounds J_1 and J_2 in the range $F \in [F_{\text{int}}, 1]$. Therefore, by result (4), J_{UB} upper bounds J_1, J_2 and J_3 in the range $F \in [F_{\text{int}}, 1]$. Moreover, for $F < F_{\text{int}}$, by results (1) and (3), $J_i(F, \rho_{\text{BD}}) \leq \sqrt{\frac{F_{\text{BD}}}{2}} < F_{\text{BD}} \leq$

$J_{\mathrm{UB}}(F,\rho_{\mathrm{BD}})$, by the definition of J_{UB} (J_{UB} runs through the point $(1/4,F_{\mathrm{BD}})$ and is increasing). This suffices to show that the upper bound holds.

Finally, we show the bounds for p_i. Recalling that $F_{\mathrm{BD}}+\lambda_1+\lambda_2+\lambda_3=1$, we have

$$\begin{aligned}\frac{\partial}{\partial F}p_1(F,\rho_{\mathrm{BD}})=\frac{2}{3}(1-\lambda_2-\lambda_3)&=\frac{2}{3}(2F_{\mathrm{BD}}+2\lambda_1-1)\\&\geq\frac{2}{3}(2F_{\mathrm{BD}}-1)>0.\end{aligned}$$

Therefore, $p_1(F,\rho_{\mathrm{BD}})$ is an increasing function of F. By symmetry, p_2 and p_3 are also increasing functions of F. Since the fidelity $F(t)$ of the 1G1B system always lies in the region $F(t)\in[1/4,F^*]$, it follows that at any point in time, the success probability p of purification may be bounded with

$$p_i\left(\frac{1}{4},\rho_{\mathrm{BD}}\right)\leq p\leq p_i\left(F^*,\rho_{\mathrm{BD}}\right).$$

□

Below are the collection of results that were used to show the bounds on the jump functions.

Proposition 6.7. *In the domain $F>0$, jump functions 1-3 intersect exactly once at the same point F_{int}, such that $J_i(F_{int},\rho_{\mathrm{new}})=\sqrt{\frac{F_{\mathrm{BD}}}{2}}<F_{\mathrm{BD}}$.*

Proof. We firstly compute the intersection point of jump functions 1 and 2. This occurs at the F value which satisfies

$$\frac{(4\lambda_1+3\lambda_2+3\lambda_3-3)F-\lambda_1}{(4\lambda_2+4\lambda_3-2)F-\lambda_2-\lambda_3-1}=\frac{(3\lambda_1+4\lambda_2+3\lambda_3-3)F-\lambda_2}{(4\lambda_1+4\lambda_3-2)F-\lambda_1-\lambda_3-1},$$

or alternatively, recalling that $F_{\mathrm{BD}}+\lambda_1+\lambda_2+\lambda_3=1$,

$$\frac{(\lambda_1-3F_{\mathrm{BD}})F-\lambda_1}{(2-4F_{\mathrm{BD}}-4\lambda_1)F-2+F_{\mathrm{BD}}+\lambda_1}=(1\leftrightarrow 2),$$

where to obtain the RHS we exchange labels 1 and 2 of the LHS. This is equivalent to

$$((\lambda_1-3F_{\mathrm{BD}})F-\lambda_1)((2-4F_{\mathrm{BD}}-4\lambda_2)F-2+F_{\mathrm{BD}}+\lambda_2)-(1\leftrightarrow 2)=0,$$

which simplifies to

$$(\lambda_1-\lambda_2)\left((2-16F_{\mathrm{BD}})F^2+(8F_{\mathrm{BD}}-4)F+2-F_{\mathrm{BD}}\right)=0. \tag{6.90}$$

Then, if $\lambda_1\neq\lambda_2$, the points of intersection depend only on F_{BD} and therefore are symmetric in λ_1, λ_2 and λ_3. The points of intersection are given by

$$F=\frac{4F_{\mathrm{BD}}-2\pm 3\sqrt{2F_{\mathrm{BD}}}}{2(8F_{\mathrm{BD}}-1)} \tag{6.91}$$

and recalling that $F_{\text{BD}} \in (1/2, 1]$, the solution lying in the domain of interest ($F > 0$) is

$$F_{\text{int}} = \frac{4F_{\text{BD}} - 2 + 3\sqrt{2F_{\text{BD}}}}{2(8F_{\text{BD}} - 1)}.$$

Then, since F_{int} is symmetric in λ_1, λ_2 and λ_3, all jump functions intersect at the point F_{int}. One may also show that

$$J_i(F_{\text{int}}, \rho_{\text{new}}) = \sqrt{\frac{F_{\text{BD}}}{2}},$$

, e.g., using software such as Mathematica. Since $F_{\text{BD}} < 1/2$, we have

$$\sqrt{\frac{1}{2}} < \sqrt{F_{\text{BD}}} \Leftrightarrow \sqrt{\frac{F_{\text{BD}}}{2}} < F_{\text{BD}}.$$

□

We now continue with the following corollary.

Corollary 6.6. *Suppose that $\lambda_1 \geq \lambda_2 \geq \lambda_3$. Then, for $F \geq F_{int}$,*

$$J_3(F, \rho_{\text{BD}}) \geq J_2(F, \rho_{\text{BD}}) \geq J_1(F, \rho_{\text{BD}}), \tag{6.92}$$

Proof. From Proposition 6.7, J_1, J_2 and J_3 will not intersect again for $F > F_{\text{int}}$. Therefore, their ordering remains the same for all $F > F_{\text{int}}$. The jump function outputting the largest fidelity in this range will therefore also have the largest limit as $F \to \infty$. We see that

$$\lim_{F\to\infty} J_i(F, \rho_{\text{BD}}) = \frac{3F_{\text{BD}} - \lambda_i}{4F_{\text{BD}} + \lambda_i - 2},$$

which is a decreasing function of λ_i. Therefore, $\lambda_3 = \min\{\lambda_1, \lambda_2, \lambda_3\}$ gives the largest limit, and J_1 satisfies (6.92). □

From Proposition 6.7 and Corollary 6.6, we know which of J_1, J_2 and J_3 provide the best fidelity for $F \in [1/2, 1]$. With the following proposition, we see that for some lower fidelities, it is better to replace with the bad link rather than choose to pump.

Proposition 6.8. *The largest fidelity obtainable by pumping a low-fidelity Werner state with ρ_{BD} and bilocal Clifford protocols is*

$$F^* = \frac{2F_{\text{BD}} - 1 + \sqrt{(2F_{\text{BD}} - 1)^2 + 2\lambda_{\min}(2F_{\text{BD}} - 1 + 2\lambda_{\min})}}{2(2F_{\text{BD}} - 1 + 2\lambda_{\min})}, \tag{6.93}$$

where $\lambda_{\min} = \min\{\lambda_1, \lambda_2, \lambda_3\}$.

Proof. Consider applying pumping protocol $i \in \{1,2,3\}$. This stops improving the Werner state fidelity at the value of F such that

$$\begin{aligned} F^* &= J_i(F^*, \rho_{\text{BD}}) \\ \Leftrightarrow F^* &= \frac{(\lambda_i - 3F_{\text{BD}})F^* - \lambda_i}{(2 - 4F_{\text{BD}} - 4\lambda_i)F^* - 2 + F_{\text{BD}} + \lambda_i} \\ \Leftrightarrow 0 &= (2 - 4F_{\text{BD}} - 4\lambda_i)F^2 + (4F_{\text{BD}} - 2)F + \lambda_i, \end{aligned}$$

which has solutions

$$F = \frac{2F_{\text{BD}} - 1 \pm \sqrt{(2F_{\text{BD}} - 1)^2 + 2\lambda_i(2F_{\text{BD}} - 1 + 2\lambda_i)}}{2(2F_{\text{BD}} - 1 + 2\lambda_i)},$$

one of which is positive and one negative. Recalling that for $F > \frac{1}{2}$, the jump function taking the largest value is J_i with $\lambda_i = \lambda_{\min}$, means that the maximum fidelity achievable is (6.93). □

Proposition 6.9. *For any ρ_{BD} with $F_{\text{BD}} > 1/2$, , for $i = 1,2,3$ $J_i(F, \rho_{\text{BD}})$ is a strictly concave and increasing function of F.*

Proof. We differentiate J_i. Firstly, consider derivatives of functions of the form

$$y = \frac{ax + b}{cx + d}.$$

This may be rewritten as

$$y = \frac{a}{c} + \frac{b - \frac{ad}{c}}{cx + d}.$$

Then,

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{ad - bc}{(cx + d)^2}, \quad \frac{\mathrm{d}^2 y}{\mathrm{d}x^2} = -2c\frac{ad - bc}{(cx + d)^3}. \tag{6.94}$$

To check the sign of these functions, we must therefore check the sign of $ad - bc$. Recalling that J_i may be rewritten as

$$J_i(F, \rho_{\text{BD}}) = \frac{(3F_{\text{BD}} - \lambda_i)F + \lambda_i}{(4F_{\text{BD}} + 4\lambda_i - 2)F + 2 - F_{\text{BD}} - \lambda_i},$$

in this case,

$$\begin{aligned} a &= 3F_{\text{BD}} - \lambda_i > \frac{3}{2} - \frac{1}{2} = 1 \\ b &= \lambda_i < \frac{1}{2} \\ c &= 4(F_{\text{BD}} + \lambda_i) - 2 \leq 4 \cdot 1 - 2 = 2 \\ d &= 2 - (F_{\text{BD}} + \lambda_i) \geq 2 - 1 = 1 \end{aligned}$$

and it follows that $ad - bc > 1 \cdot 1 - 2 \cdot 1/2 = 0$. Then, since

$$c = 4F_{\mathrm{BD}} + 4\lambda_i - 2 > 4 \cdot \frac{1}{2} + 4\lambda_i - 2 = 4\lambda_i \geq 0,$$

it follows from (6.94) that

$$\frac{\partial}{\partial F} J_i(F, \rho_{\mathrm{BD}}) > 0, \quad \frac{\partial^2}{\partial F^2} J_i(F, \rho_{\mathrm{BD}}) < 0.$$

Therefore, J_i is a strictly concave and increasing function of F. □

Proposition 6.10. *Suppose that $\lambda_1 \geq \lambda_2 \geq \lambda_3$. Consider the tangent to $J_3(F, \rho_{\mathrm{BD}})$ at $F = 1$. Denote this by $J_{\tan}(F, \rho_{\mathrm{BD}})$. Then, this lies below J_{UB} for all $F \in [1/4, 1]$, i.e.,*

$$J_{\tan}(F, \rho_{\mathrm{BD}}) \leq J_{\mathrm{UB}}(F, \rho_{\mathrm{BD}}),$$

where

$$J_{\mathrm{UB}}(F, \rho_{\mathrm{BD}}) = \frac{4(1 - F_{\mathrm{BD}})}{3} F + \frac{4F_{\mathrm{BD}} - 1}{3}$$

is the linear upper bound from Lemma 6.2.

Proof. We firstly compute the formula for the tangent to J_i at $F = 1$. Recalling the formula (6.94), this has gradient

$$\frac{\partial}{\partial F} J_3(F, \rho_{\mathrm{BD}})\big|_{F=1} = \frac{ad - bc}{(c + d)^2} = \frac{6F_{\mathrm{BD}} - 3(F_{\mathrm{BD}} + \lambda_3)^2}{(3(F_{\mathrm{BD}} + \lambda_3))^2} = \frac{2F_{\mathrm{BD}}}{3(F_{\mathrm{BD}} + \lambda_3)^2} - \frac{1}{3}.$$

Since the tangent runs through the point $(1, J_3(1, \rho_{\mathrm{BD}}))$, it has formula

$$J_{\tan}(F, \rho_{\mathrm{BD}}) = \left(\frac{2F_{\mathrm{BD}}}{3(F_{\mathrm{BD}} + \lambda_3)^2} - \frac{1}{3} \right)(F - 1) + \frac{F_{\mathrm{BD}}}{F_{\mathrm{BD}} + \lambda_3},$$

where we have used $J_i(1, \rho_{\mathrm{BD}}) = F_{\mathrm{BD}}/(F_{\mathrm{BD}} + \lambda_i)$. We note that at $F = 1$,

$$J_{\mathrm{UB}}(1, \rho_{\mathrm{BD}}) = 1 \geq \frac{F_{\mathrm{BD}}}{F_{\mathrm{BD}} + \lambda_3} = J_{\tan}(1, \rho_{\mathrm{BD}}).$$

Therefore, to show the proposition, it suffices to show that

$$J_{\mathrm{UB}}\left(\frac{1}{4}, \rho_{\mathrm{BD}} \right) \geq J_{\tan}\left(\frac{1}{4}, \rho_{\mathrm{BD}} \right), \tag{6.95}$$

since both J_{UB} and $J_{\tan}$ are linear in F and therefore intersect at most once. Now,

$$\begin{aligned} J_{\mathrm{UB}}\left(\frac{1}{4}, \rho_{\mathrm{BD}} \right) - J_{\tan}\left(\frac{1}{4}, \rho_{\mathrm{BD}} \right) &= F_{\mathrm{BD}} - \left(\frac{2F_{\mathrm{BD}}}{3(F_{\mathrm{BD}} + \lambda_3)^2} - \frac{1}{3} \right)\left(-\frac{3}{4} \right) - \frac{F_{\mathrm{BD}}}{F_{\mathrm{BD}} + \lambda_3} \\ &= F_{\mathrm{BD}} - \frac{1}{4} + \frac{F_{\mathrm{BD}}}{2(F_{\mathrm{BD}} + \lambda_3)^2} - \frac{F_{\mathrm{BD}}}{F_{\mathrm{BD}} + \lambda_3}. \end{aligned}$$

Now, let $x := F_{\mathrm{BD}} + \lambda_3$, and

$$h(x) := F_{\mathrm{BD}} - \frac{1}{4} + \frac{F_{\mathrm{BD}}}{2x^2} - \frac{F_{\mathrm{BD}}}{x}.$$

By the assumption that $\lambda_3 = \min\{\lambda_1, \lambda_2, \lambda_3\}$ and the condition $F_{\mathrm{BD}} + \lambda_1 + \lambda_2 + \lambda_3 = 1$, it follows that

$$\lambda_3 \in \left[0, \frac{1-F_{\mathrm{BD}}}{3}\right], \quad \text{and} \quad x \in \left[F_{\mathrm{BD}}, \frac{1+2F_{\mathrm{BD}}}{3}\right]. \tag{6.96}$$

To prove the proposition, it therefore suffices to show positivity of h for x in the range (6.96). We start by establishing the monotonicity of h:

$$\frac{\partial}{\partial x} h(x) = -\frac{F_{\mathrm{BD}}}{x^3} + \frac{F_{\mathrm{BD}}}{x^2} = -\frac{F_{\mathrm{BD}}}{x^3}(1-x) \leq 0,$$

since $x = F_{\mathrm{BD}} + \lambda_3 \leq 1$. We therefore see that h is decreasing. To show that h is positive in the range (6.96), it therefore suffices to show that

$$h\left(\frac{1+2F_{\mathrm{BD}}}{3}\right) \geq 0.$$

We have

$$\begin{aligned} h\left(\frac{1+2F_{\mathrm{BD}}}{3}\right) &= F_{\mathrm{BD}} - \frac{1}{4} + \frac{9F_{\mathrm{BD}}}{2(1+2F_{\mathrm{BD}})^2} - \frac{3F_{\mathrm{BD}}}{1+2F_{\mathrm{BD}}} \\ &= F_{\mathrm{BD}} - \frac{1}{4} + 6F_{\mathrm{BD}}\left(\frac{\frac{3}{4} - \frac{1}{2}(1+2F_{\mathrm{BD}})}{(1+2F_{\mathrm{BD}})^2}\right) \\ &= F_{\mathrm{BD}} - \frac{1}{4} + 6F_{\mathrm{BD}}\frac{\frac{1}{4} - F_{\mathrm{BD}}}{(1+2F_{\mathrm{BD}})^2} \\ &= \left(F_{\mathrm{BD}} - \frac{1}{4}\right)\left(1 - \frac{6F_{\mathrm{BD}}}{(1+2F_{\mathrm{BD}})^2}\right). \end{aligned}$$

Then, since $F_{\mathrm{BD}} > \frac{1}{2}$, we have

$$\begin{aligned} h\left(\frac{1+2F_{\mathrm{BD}}}{3}\right) > 0 &\Leftrightarrow 1 - \frac{6F_{\mathrm{BD}}}{(1+2F_{\mathrm{BD}})^2} > 0 \\ &\Leftrightarrow (1+2F_{\mathrm{BD}})^2 > 6F_{\mathrm{BD}} \\ &\Leftrightarrow 4F_{\mathrm{BD}}^2 - 2F_{\mathrm{BD}} + 1 > 0 \\ &\Leftrightarrow (1-2F_{\mathrm{BD}})^2 + 2F_{\mathrm{BD}} > 0, \end{aligned}$$

which holds. We therefore see that

$$h(x) \geq h\left(\frac{1+2F_{\mathrm{BD}}}{3}\right) > 0$$

for all x in the range (6.96), and therefore

$$J_{\mathrm{UB}}\left(\frac{1}{4}, \rho_{\mathrm{BD}}\right) - J_{\tan}\left(\frac{1}{4}, \rho_{\mathrm{BD}}\right) > 0.$$

This suffices to show the proposition. □

6.10.3. ADDITIONAL PROOFS

Lemma 6.5. *A Bell-diagonal state*

$$\rho = \lambda_0 \left|\phi^+\right\rangle\!\left\langle\phi^+\right| + \lambda_1 \left|\psi^+\right\rangle\!\left\langle\psi^+\right| + \lambda_2 \left|\psi^-\right\rangle\!\left\langle\psi^-\right| + \lambda_3 \left|\phi^-\right\rangle\!\left\langle\phi^-\right|,$$

with $\lambda_0 + \lambda_1 + \lambda_2 + \lambda_3 = 1$, *is entangled if and only if* $\lambda_i > 1/2$ *for some* i.

Proof of Lemma 6.5. We will analyze the entanglement of a Bell-diagonal state using the Peres-Horodecki criterion, which states that a bipartite, 2×2 dimensional quantum state ρ is entangled if and only if the partial transpose of ρ has at least one negative eigenvalue [85, 143]. A Bell-diagonal state can be written in the Bell basis as

$$\rho = \lambda_0 \left|\phi^+\right\rangle\!\left\langle\phi^+\right| + \lambda_1 \left|\psi^+\right\rangle\!\left\langle\psi^+\right| + \lambda_2 \left|\psi^-\right\rangle\!\left\langle\psi^-\right| + \lambda_3 \left|\phi^-\right\rangle\!\left\langle\phi^-\right|.$$

In the computational basis, $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$, the Bell-diagonal state can be written as

$$\rho = \begin{pmatrix} \lambda_0+\lambda_3 & 0 & 0 & \lambda_0-\lambda_3 \\ 0 & \lambda_1+\lambda_2 & \lambda_1-\lambda_2 & 0 \\ 0 & \lambda_1-\lambda_2 & \lambda_1+\lambda_2 & 0 \\ \lambda_0-\lambda_3 & 0 & 0 & \lambda_0+\lambda_3 \end{pmatrix}.$$

The partial transpose of this density matrix is given by

$$\rho^{\mathrm{PT}} = \begin{pmatrix} \lambda_0+\lambda_3 & 0 & 0 & \lambda_1-\lambda_2 \\ 0 & \lambda_1+\lambda_2 & \lambda_0-\lambda_3 & 0 \\ 0 & \lambda_0-\lambda_3 & \lambda_1+\lambda_2 & 0 \\ \lambda_1-\lambda_2 & 0 & 0 & \lambda_0+\lambda_3 \end{pmatrix}.$$

The eigenvalues of the partial transpose are $\xi_i = 1-2\lambda_i$, $i = 1,2,3,4$. One of the eigenvalues is negative iff $\lambda_i > 1/2$ for some i. Therefore, according to the Peres-Horodecki criterion, the state is entangled iff $\lambda_i > 1/2$ for some i. Since these λ_i correspond to the fidelity of ρ to one of the Bell states (e.g., $F(\rho, \left|\phi^+\right\rangle) \equiv \left\langle\phi^+\right| \rho \left|\phi^+\right\rangle = \lambda_0$), we conclude that the state is entangled iff the fidelity to one of the Bell states is larger than $1/2$. □

6.11. [APPENDIX] NUMERICAL SIMULATIONS

In our analytical calculations, we assumed a purification protocol with constant success probability (which implies a linear jump function, as shown in Appendix 6.7). This allowed us to derive bounds for the performance of any 1G1B entanglement buffering system that uses bilocal Clifford protocols. However, the success probability of these purification protocols is in general linear in the fidelity of the buffered state (see Appendix 6.7). In this appendix, we compare the analytical bounds, which assume a constant success probability, to the actual values obtained via a simulation that considers the true (linear, non-constant) success probability.

Our discrete-event simulation keeps track of the buffered link, which decoheres until an event is triggered. These events could correspond to a consumption request (which

consumes the buffered memory) or a successful entanglement generation (which is followed by pumping, with probability q). When purification is performed, it succeeds with a probability that depends linearly on the fidelity of the buffered link (see Appendix 6.7).

To compute the average consumed fidelity and the availability, we run the simulation N_{samples} times. In each realization i of the process, we let the system evolve over t_{sim} units of time until convergence to a steady state, and we record the fidelity of the buffered link $F_i(t_{\text{sim}})$ (if the memory is empty, the fidelity is set to zero, as was specified in Definition 6.6). Then, we estimate the average consumed fidelity as the average fidelity of the buffered link at t_{sim} (conditional on the buffered link being present):

$$\overline{F} \approx \overline{F}' \equiv \frac{\sum_{i=1}^{N_{\text{samples}}} F_i(t_{\text{sim}})}{N'_{\text{samples}}}, \tag{6.97}$$

where

$$N'_{\text{samples}} = \sum_{i=1}^{N_{\text{samples}}} \mathbb{1}_{F_i(t_{\text{sim}})>0} \tag{6.98}$$

is the number of samples in which $F_i(t_{\text{sim}}) > 0$ ($\mathbb{1}$ is the indicator function). We measure the error in the estimate using the standard error:

$$\varepsilon_F = \sqrt{\frac{\sum_{i=1}^{N'_{\text{samples}}} \left(F_i(t_{\text{sim}}) - \overline{F}'\right)^2}{N'_{\text{samples}}\left(N'_{\text{samples}} - 1\right)}}, \tag{6.99}$$

which corresponds to the square root of the unbiased sample variance divided by the number of samples. The availability is estimated as the proportion of samples in which there is a buffered link at time t_{sim}:

$$A \approx A' \equiv \frac{1}{N_{\text{samples}}} \sum_{i=0}^{N_{\text{samples}}} \mathbb{1}_{F_i(t_{\text{sim}})>0}, \tag{6.100}$$

Note that A' is the average of a binary random variable. We can therefore model this random variable as Bernoulli-distributed with probability of success A'. This yields a variance $A'(1-A')$, which allows us to compute the standard error as

$$\varepsilon_A = \sqrt{\frac{A'(1-A')}{N_{\text{samples}}}}. \tag{6.101}$$

Next, we study again the example from Figure 6.7, and we compare the bounds discussed in the main text with the results from our simulation. In Figure 6.12, we show the same lower and upper bounds (yellow and dark blue lines, respectively) from Figure 6.7. We simulated three buffering systems, each of them using the unique bilocal Clifford protocols 1, 2, and 3 from Table 6.2 (we neglect protocols 4-7 since they are trivial). We emphasize that these simulations consider the true probabilities of success (which are linear but non-constant in the fidelity of the buffered link) and the true jump functions (rational in the fidelity of the buffered link) of the purification protocols. Figure 6.12

shows the availability and average consumed fidelity attained by each of these systems, for different values of q. We first note that protocols 2 (blue circles) and 3 (red crosses) are equivalent. This is due to the symmetry of the newly generated state considered in this example, $\rho_{\text{new}} = F_{\text{new}} |\phi^+\rangle\langle\phi^+| + (1 - F_{\text{new}}) \left(|\psi^+\rangle\langle\psi^+| + |\psi^-\rangle\langle\psi^-|\right)/2$. More importantly, the performance of the simulated systems lies within the analytical bounds, which were derived assuming a constant probability of success. This serves as empirical evidence that our simplified model is still useful when lifting the assumption about a constant probability of success, and can guide the design of more complex and realistic buffering systems.

CODE AVAILABILITY

Our code can be found at https://github.com/AlvaroGI/buffering-1G1B.

AUTHOR CONTRIBUTIONS

ÁGI and BD conceived and defined the project. BD proved Propositions 6.1 and 6.2, Theorem 6.1, and Lemmas 6.1 and 6.2. ÁGI proved Proposition 6.3. ÁGI analyzed bilocal Clifford protocols in the context of 1G1B and coded the Monte Carlo simulation used to validate analytical results. BD and ÁGI wrote the paper, which was published as ref. [50]. SW provided active feedback at every stage of the project.

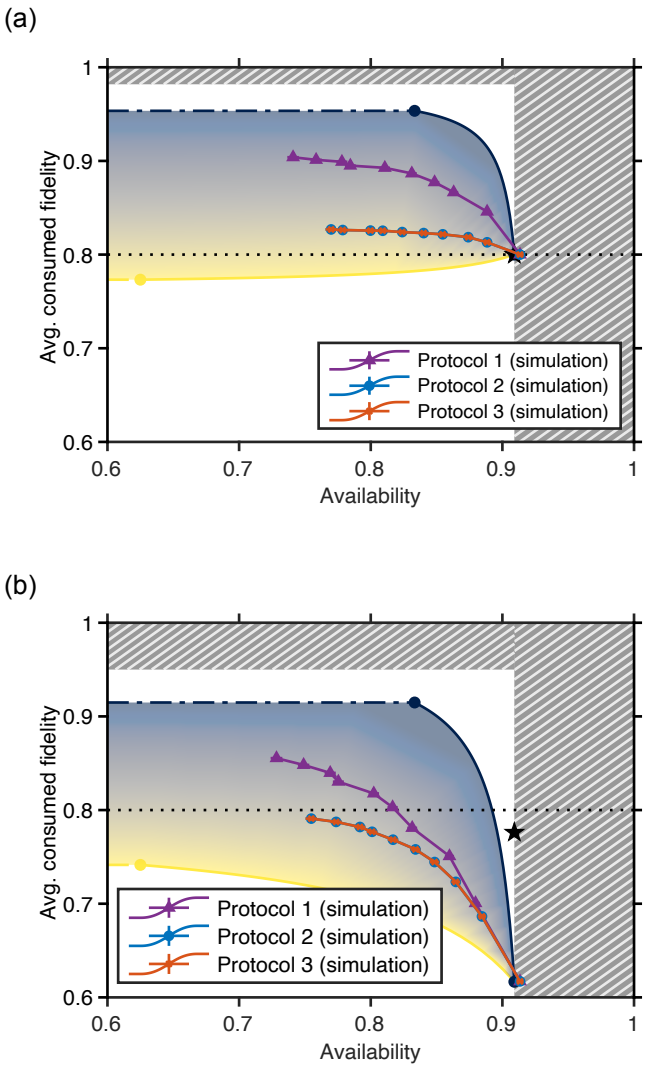

Figure 6.12: **Bounds derived assuming a constant probability of success still apply when the assumption is lifted.** (a) Noiseless memories ($\Gamma = 0$) or (b) noisy memories ($\Gamma = 5 \cdot 10^{-2}$ a.u.). For a given target availability, the average consumed fidelity is within the blue/yellow region (see Corollary 6.1). Availability is maximized for $q = 0$ (q is the probability of purification after successful entanglement generation), and it decreases for increasing q. White regions cannot be achieved by bilocal Clifford protocols. Striped regions cannot be achieved by any pumping protocol. Black star: performance of the replacement protocol (buffered link is replaced by new links). Dotted line: fidelity of newly generated entangled links. Solid lines with markers: performance of the 1G1B system obtained via simulation, using the true jump functions and true probabilities of success of purification protocols 1, 2, and 3 from Table 6.2, ($q = 0$ for the rightmost data point, decreasing in intervals of 0.111 until reaching $q = 1$ in the leftmost data point). The simulation considers a linear probability of success, unlike the analytical calculations, in which this probability is assumed to be constant. Parameters used in this example (times and rates in the same arbitrary units): $\lambda = 1$, $\mu = 0.1$, $F_{\text{new}} = 0.8$, $\rho_{\text{new}} = F_{\text{new}} |\phi^+\rangle\langle\phi^+| + (1 - F_{\text{new}}) \left(|\psi^+\rangle\langle\psi^+| + |\psi^-\rangle\langle\psi^-|\right)/2$. Numerical parameters used in the simulation: $t_{\text{sim}} = 50$, $N_{\text{samples}} = 10^4$.

7

Entanglement Buffering with Multiple Quantum Memories

Álvaro G. Iñesta*, Bethany Davies*, Sounak Kar and Stephanie Wehner

Now, here, you see, it takes all the running you can do,
to keep in the same place.
If you want to get somewhere else,
you must run at least twice as fast as that!

— the Red Queen

Entanglement buffers are systems that maintain high-quality entanglement, ensuring it is readily available for consumption when needed. In this work, we study the performance of a two-node buffer, where each node has one long-lived quantum memory for storing entanglement and multiple short-lived memories for generating fresh entanglement. Newly generated entanglement may be used to purify the stored entanglement, which degrades over time. Stored entanglement may be removed due to failed purification or consumption. We derive analytical expressions for the system performance, which is measured using the entanglement availability and the average fidelity upon consumption. Our solutions are computationally efficient to evaluate, and they provide fundamental bounds to the performance of purification-based entanglement buffers. We show that purification must be performed as frequently as possible to maximize the average fidelity of entanglement upon consumption, even if this often leads to the loss of high-quality entanglement due

* These authors contributed equally.
This chapter has been published separately in ref. [90].

to purification failures. Moreover, we obtain heuristics for the design of good purification policies in practical systems. A key finding is that simple purification protocols, such as DEJMPS, often provide superior buffering performance compared to protocols that maximize output fidelity.

7.1. INTRODUCTION

Entanglement is a fundamental resource for many quantum network applications, including some quantum key distribution protocols [12, 59], distributed quantum sensing [61, 123, 152, 199], and coordination tasks where communication is either prohibited or insufficiently fast [20, 27]. Pre-distributing entanglement between remote parties would eliminate the need to generate and distribute entangled states on demand, saving time and resources [35, 66, 94, 149]. However, entanglement degrades over time due to decoherence, preventing long-term storage.

Entanglement buffers are systems that store entanglement until it is needed for an application. Passive buffers, which store entanglement in quantum memories, are constrained by the coherence time of these memories [5]. To overcome this limitation, purification-based entanglement buffers have been proposed [50, 60] (note that ref. [50] is contained in Chapter 6). These systems store entangled states and employ purification protocols to ensure the states remain high quality, mitigating the effects of decoherence. Purification protocols take m low-quality entangled states as input and produce n higher-quality states as output, typically with $m > n$ [15, 53, 56, 205]. These protocols often involve some probability of failure, in which case all the input states are lost and no entanglement is produced. Here, we focus on purification-based buffers.

As proposed in ref. [50] (Chapter 6), the performance of an entanglement buffer can be measured with two quantities: the availability (probability that entanglement is available for consumption when requested, see Definition 7.2) and the average consumed fidelity (average quality of entanglement at the time of consumption, see Definition 7.3). As well as having practical utility, entanglement buffers are a useful theoretical tool in order to understand the impact of several important interacting processes that occur in a quantum network: ongoing generation, purification, and consumption of entanglement. Of major interest is the impact of the entanglement purification protocol on the performance of the system. Since the success probability of entanglement purification typically depends on the fidelity of the input states, any rate and fidelity metrics are inherently coupled in systems making use of purification. This coupling adds complexity to analytical calculations. Consequently, most analytical studies on the performance of quantum networking systems exclude purification, and its impact on performance is typically explored with numerical methods [79, 189]. Nevertheless, as is a main result in this work, for entanglement buffering systems closed-form solutions are obtainable for a fully general purification protocol. One may then efficiently compute the performance of a particular purification policy, as well as make formal statements about how often purification should be applied to the buffered entanglement.

Here, we study the *1GnB system*: a purification-based entanglement buffer with one good (long-lived) memory and n bad (short-lived) memories. The good memory can

store entanglement, which can be consumed at any time by an application. In contrast, bad memories can generate entanglement concurrently but cannot store it; they act as communication qubits. For instance, carbon-13 nuclear spins in diamond can serve as good memories with coherence times up to 1 min [19], while electron spins in nitrogen-vacancy centers may function as communication qubits, with coherence times generally below 1 s [1].

Each time entanglement is generated in some of the bad memories, the system may choose to immediately use it to purify the entanglement stored in the good memory. If purification is not attempted, the newly generated entanglement is discarded. We illustrate the 1GnB system in Figure 7.1. Note that the physical platform must enable easy access to stored entanglement for consumption and purification. However, network activities, such as repeated entanglement generation attempts and purification, may introduce additional noise, reducing memory lifetimes. For example, in ref. [147], even when the carbon-13 nuclear spin used as a storage qubit is protected from network noise by applying stronger magnetic fields, it exhibits a shortened lifetime of approximately 11.6 ms.

The 1GnB buffering system is a generalization of the 1G1B system that was originally proposed in [50] (Chapter 6). 1G1B is a system with only one good quantum memory and one bad memory. Here, we generalize the work from Chapter 6 in three main ways. Firstly, we now consider several (n) bad memories. Including several bad memories in our model now means that there is the possibility of generating multiple entangled links in the same entanglement generation attempt, for example via frequency [5, 37, 196] or time multiplexing [108, 131], which are commonly proposed ways of improving the rate of entanglement generation [43, 134, 182]. Moreover, the simultaneous generation of multiple links opens up the use of stronger purification protocols, thereby providing an improvement to system fidelity metrics as well as the rate. Note again that the physical implementation of the buffer must allow for such multiplexing and for purification of the generated entanglement. The second generalization from previous work is that we now model the system in discrete time rather than continuous time, which is more accurate to real-world systems, as entanglement generation typically happens in discrete attempts (see, e.g., refs. [9, 16, 180, 207]). Finally, we now derive our solutions for a fully arbitrary purification protocol. In particular, the solutions for performance metrics presented in ref. [50] (Chapter 6) only apply for purification protocols with a constant probability of success (i.e., the success probability must be independent of the fidelity of the buffered quantum state). However, in this work, we remove this assumption and derive closed-form solutions for the availability and the average consumed fidelity of buffers that use arbitrary purification protocols. This is in contrast to [60], where although performance metrics are derived analytically and the probability of success is not necessarily constant, their computation requires solving a linear system of equations, which has dimension that scales with system parameters such as the memory lifetime.

Here, we firstly provide analytical expressions for the availability, A, and the average consumed fidelity, $\overline{F}$, of the 1GnB system (see model description in Section 7.2). Then, we use these expressions to find fundamental limits to the performance of entanglement buffers. Lastly, we investigate how the 1GnB system should be operated: because there is a large amount of freedom in the choice of purification protocols, it is not clear what purification strategies should be employed to maximize A and $\overline{F}$. For example, would it

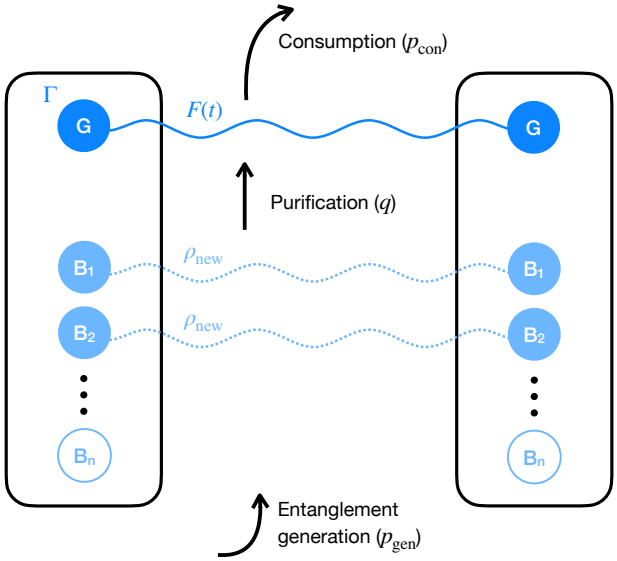

Figure 7.1: **Illustration of the 1GnB buffering system.** Entanglement generation is attempted in every bad memory ($B_1, \ldots, B_n$) simultaneously in each time slot. Each memory succeeds with probability p_{gen}. The good memory, G, stores entanglement, which decoheres at rate Γ. When G is full and new entanglement is generated in any of the B memories, a purification subroutine is applied with probability q. Entanglement is consumed from G with probability p_{con} in each time slot.

be beneficial to use a purification subroutine that provides a larger fidelity boost (which could increase $\overline{F}$) if this comes at the cost of a higher probability of failure (which means losing high-quality entanglement more frequently, decreasing A and maybe also $\overline{F}$)? Our main findings are the following:

- MONOTONIC PERFORMANCE – We show that, to maximize the average consumed fidelity, purification must be performed as much as possible, i.e., every time entanglement is generated in any of the bad memories. This holds even if the purification protocol has a large probability of failure. Nevertheless, there is a tradeoff between both performance metrics, since the availability decreases when purification is performed more frequently.

- FUNDAMENTAL BOUNDS – We provide upper and lower bounds for the availability and the average consumed fidelity of a 1GnB system, which constitute fundamental limits to the impact that a purification policy can have on the performance.

- SIMPLE CAN BE BETTER THAN OPTIMAL – Simple purification protocols can greatly outperform advanced purification protocols that maximize the fidelity of the output entangled state. For example, we find that a buffering system using the 2-to-1 purification protocol from ref. [53] (known as DEJMPS) can outperform a system using the n-to-1 optimal bilocal Clifford protocol from ref. [95], in terms of both availability and average consumed fidelity.

7.2. THE 1GnB SYSTEM

In this section, we provide a short description of the entanglement buffering setup (see Figure 7.1). The goal of the system is to buffer bipartite entanglement shared between two nodes. These nodes could be, for example, two end users in a quantum network or two processors in a quantum computing cluster. We refer to bipartite entanglement as an *entangled link* between the two nodes. In the 1GnB system:

- Each node has *one long-lived memory* (good, G) and n *short-lived memories* (bad, B).
- The G memories are used to store the entangled link. We assume *the link stored in memory is a Werner state* (any bipartite state can be transformed into a Werner state with the same fidelity by applying extra noise, a process known as *twirling* [13, 86]). Such a state can be parametrized with its fidelity to the target maximally entangled state, F.
- The entangled link stored in G is subject to *depolarizing noise* with memory lifetime $1/\Gamma$, which causes an exponential decay in fidelity with rate Γ. That is, if the link in memory has an initial fidelity F, after time t this reduces to

$$F \mapsto \left(F - \frac{1}{4}\right) e^{-\Gamma t} + \frac{1}{4}. \tag{7.1}$$

- Before each entanglement generation attempt, the system checks if a new *consumption request* has arrived. The arrival of a new consumption request in each time step occurs with probability p_{con}. If there is a link stored in memory G when a consumption request arrives, the link is immediately consumed and therefore removed from the memory. This takes up the entire time step. If there is no link available, the request is discarded and the system proceeds with the entanglement generation attempt.
- The B memories are used to generate new entangled links. In the literature, these are usually called communication or broker qubits [11]. This communication qubit can be, for example, the electron spin in a nitrogen-vacancy center [16, 114, 161]. Every time step that is not taken up by consumption, *entanglement generation* is attempted in all n bad memories simultaneously, e.g., using frequency or spatial multiplexing, and each of them independently generates an entangled link with probability p_{gen}. This means that, after each multiplexed attempt, the number of successfully generated links follows a binomial distribution with parameters (n, p_{gen}). Each of these new links is of the form ρ_{new}, which is an arbitrary state that depends on the entanglement generation protocol employed (see, e.g., refs. [9, 33, 97, 180]).
- When $k \geq 1$ entangled links are generated in the B memories and the G memory is empty, one of the links is *transferred* to the G memory. If the G memory is occupied, the new links may be used to *purify* the link in memory. The system decides to attempt purification with probability q. If the system does not decide to purify, the new links are discarded. If the system decides to attempt purification and this

Table 7.1: **Parameters of the 1G1B system.** See main text for further details.

Hardware	
n	Number of short-lived memories.
p_{gen}	Probability of successful entanglement generation attempt.
ρ_{new}	Bipartite entangled state produced after a successful entanglement generation. This state has fidelity F_{new}.
Γ	Rate of decoherence.
Application	
p_{con}	Probability of consumption request.
Purification policy	
q	Probability of attempting purification immediately after a successful entanglement generation attempt (otherwise the new links are discarded).
$J_k(F)$	Jump function. Given a buffered link with fidelity F, $J_k(F)$ is the fidelity immediately following a successful purification using k newly generated links. Rational function with coefficients a_k, b_k, c_k, d_k – see (7.3).
$p_k(F)$	Probability of successful purification using k newly generated links. Linear function with coefficients c_k, d_k – see (7.4).

succeeds, then the resultant link in the G memory is twirled, converting it into the form of a Werner state with the same fidelity.

Table 7.1 summarizes all variables of the system. Next, we discuss how to model the purification strategy.

7.2.1. PURIFICATION POLICY

The main degree of freedom in the 1GnB system is the choice of purification protocol. This is given by the purification policy.

Definition 7.1. *The* purification policy π *is a function that indicates the purification protocol that must be used when k links are generated in the B memories,*

$$\pi : k \in \{1, \dots, n\} \mapsto \pi(k) \in \mathscr{P}_{k+1}, \tag{7.2}$$

where $\mathscr{P}_m$ is the set of all m-to-1 purification protocols.

Protocol $\pi(k)$ of purification policy π is the $(k+1)$-to-1 purification protocol that is used when k new links are generated in the B memories (examples of basic protocols can be found in refs. [15, 52, 53]; see ref. [57] for a survey). The purification protocol updates the fidelity of the buffered link from F to $J_k(F)$, where

$$J_k(F) = \frac{1}{4} + \frac{a_k(\rho_{\text{new}})\left(F - \frac{1}{4}\right) + b_k(\rho_{\text{new}})}{c_k(\rho_{\text{new}})\left(F - \frac{1}{4}\right) + d_k(\rho_{\text{new}})}. \tag{7.3}$$

We call J_k the *jump function of protocol* $\pi(k)$. The protocol succeeds with probability

$$p_k(F) = c_k(\rho_{\text{new}})\left(F - \frac{1}{4}\right) + d_k(\rho_{\text{new}}), \tag{7.4}$$

otherwise all of the links (including the buffered one) are discarded and the G memory becomes empty. In Appendix 6.7, the forms (7.3) and (7.4) for the output fidelity and success probability are justified, given that the buffered link is a Werner state with fidelity F and any other input state is given by the same arbitrary density matrix ρ_{new}. We therefore see that the action of any purification protocol on the fidelity of the buffered link is determined by the four parameters $a_k(\rho_{\text{new}})$, $b_k(\rho_{\text{new}})$, $c_k(\rho_{\text{new}})$, $d_k(\rho_{\text{new}})$. In Appendix 7.8, we discuss the values that these coefficients can take. As an example, we also provide the explicit form of these coefficients for the well-known 2-to-1 DEJMPS protocol [53].

Lastly, note that purification policy π employs protocol $\pi(k)$ when k new links are generated. However, this does not mean that all the new links are used in the protocol. For example, a policy may simply replace the link in memory with a newly generated link and ignore the rest of the new links.

7.2.2. FIDELITY OF THE BUFFERED ENTANGLEMENT

Given the system description, we now view 1GnB as a discrete-time stochastic process. In particular, at time t the state of the system is the fidelity $F(t)$ of the buffered link, as this is the only quantity that can change over time. If there is no link in the buffered memory at time t, we let $F(t) = 0$. This is for notational convenience, as recalling the decoherence (7.1), one can never reach zero fidelity if there is a link present.

We now describe the behavior of $F(t)$ when moving from time t to time $t+1$:

- Let us consider first $F(t) = 0$. If entanglement generation is unsuccessful, in the next time step the fidelity will remain at that value: $F(t+1) = 0$. If entanglement generation is successful, in the next time step the fidelity will be F_{new}, where $F_{\text{new}} = \langle\Phi_{00}|\,\rho_{\text{new}}\,|\Phi_{00}\rangle$ is the fidelity of freshly generated links. We will assume that $F_{\text{new}} > 1/4$.

- If $F(t) > 0$, then in the next time step this could evolve in one of the following ways: (i) if no purification is attempted then the fidelity simply decoheres by one unit of time according to (7.1); (ii) if k new links are generated and purification is successfully performed, the fidelity decoheres by one time step and is then mapped according to the corresponding jump function (7.3); (iii) if a consumption request has arrived or if purification fails, the link is removed and the system becomes empty.

In Figure 7.2, we illustrate an example of how the fidelity may evolve.

In the following subsection, we define the two performance metrics: the availability and the average consumed fidelity. We then present simple closed-form solutions for these two performance metrics in the 1GnB system.

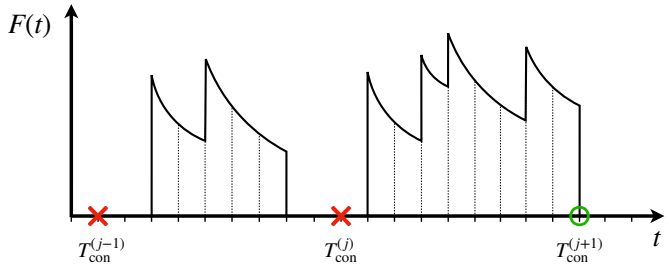

Figure 7.2: **Example dynamics of the 1GnB system.** Here, the fidelity $F(t)$ of the link in the G memory is plotted against time. The vertical lines represent discretization of time. The jumps in fidelity occur when the link is purified successfully. In between purifications, the link is subject to decoherence and the fidelity decreases. The link in the G memory is removed due to either failed purification or consumption. When there is no link in memory, $F(t) = 0$. The j-th consumption request arrives at time $T_{\text{con}}^{(j)}$. The green tick (red crosses) represent when a consumption request is (is not) served.

7.2.3. BUFFERING PERFORMANCE

The first step towards the design of useful entanglement buffers is to determine a suitable way to measure performance. Here, we define two performance metrics for entanglement buffers – these quantities were proposed in ref. [50] (Chapter 6), where they were used to study the 1G1B system. Then, we provide exact, closed-form expressions for these two performance metrics in the 1GnB system.

Our first metric is the *availability*. A user is able to consume entanglement only when there is a link available in memory G at the time of requesting the entanglement. Therefore, an important performance measure is the probability that entanglement is available when a consumption request arrives.

Definition 7.2 (Availability)**.** *The availability A is the probability that there is an entangled link present in memory G when a consumption request arrives. This is defined as*

$$A = \lim_{m\to\infty} \frac{1}{m} \sum_{j=1}^{m} \mathbb{1}_{\textit{link exists}}(T_{\text{con}}^{(j)}), \tag{7.5}$$

where $T_{\text{con}}^{(j)}$ is the arrival time of the j-th consumption request, and $\mathbb{1}_{\textit{link exists}}(t)$ is and indicator function that takes the values one if there is a link stored in memory G at time t, and zero otherwise.

The availability may be seen as a rate metric: it determines the rate at which entanglement can be consumed. The second performance metric is the *average consumed fidelity*, which captures the average quality of consumed entanglement.

Definition 7.3 (Average consumed fidelity)**.** *The average consumed fidelity is the average fidelity of the entangled link upon consumption, conditional on a link being present. More specifically,*

$$\overline{F} = \lim_{m\to\infty} \frac{\sum_{j=1}^{m} \mathbb{1}_{\textit{link exists}}\left(T_{\text{con}}^{(j)}\right) \cdot F^{-}(T_{\text{con}}^{(j)})}{\sum_{j=1}^{m} \mathbb{1}_{\textit{link exists}}(T_{\text{con}}^{(j)})}, \tag{7.6}$$

where

$$F^{-}(t)=\begin{cases} e^{-\Gamma}\left(F(t-1)-\frac{1}{4}\right)+\frac{1}{4}, \text{ if } F(t-1)>0, \\ 0, \text{ if } F(t-1)=0. \end{cases} \tag{7.7}$$

is the fidelity of the link stored in memory G at the end of the previous timestep at time $t-1$ (and therefore consumed at time t), and $T_{\text{con}}^{(j)}$ is the arrival time of the j-th consumption request.

The indicator function in the numerator of (7.6) is included for clarity, but is not necessary: if there is no link in memory at time t, then $F(t)=0$ by definition. We note that the Definitions 7.2 and 7.3 are presented differently to how they were in Chapter 6 and in ref. [50]. This is because the new definitions have a clearer operational meaning, as they are from the viewpoint of the consumer. However, in Appendix 7.6 we show that these metrics are equivalent for the 1GnB system.

As our first main result, we derive analytical solutions for the availability and the average consumed fidelity in the 1GnB system.

Theorem 7.1 (Formula for the availability)**.** *The availability of the 1GnB system is given by*

$$A=\frac{\mathbb{E}[T_{\text{occ}}]}{\mathbb{E}[T_{\text{gen}}]+\mathbb{E}[T_{\text{occ}}]}, \text{ almost surely,} \tag{7.8}$$

where T_{gen} is the time to generate new entangled links and T_{occ} is the time from when the G memory becomes occupied until it is emptied due to consumption or to failed purification. The expected values are given by

$$\mathbb{E}[T_{\text{gen}}]=\frac{1}{1-\left(1-p_{\text{gen}}\right)^{n}} \tag{7.9}$$

and

$$\mathbb{E}[T_{\text{occ}}]=\frac{1-\tilde{A}+\tilde{C}(F_{\text{new}}-\frac{1}{4})}{\left[(1-\tilde{A})(1-\tilde{D})-\tilde{B}\tilde{C}\right]\tilde{P}}, \tag{7.10}$$

with

$$\tilde{A}:=\frac{q(1-p_{\text{con}})\tilde{a}}{e^{\Gamma}-\left(1-q+q\left(1-p_{\text{gen}}\right)^{n}\right)(1-p_{\text{con}})}, \quad \tilde{B}:=\frac{q(1-p_{\text{con}})\tilde{b}}{p_{\text{con}}+q\left(1-\left(1-p_{\text{gen}}\right)^{n}\right)(1-p_{\text{con}})},$$
$$\tilde{C}:=\frac{q(1-p_{\text{con}})\tilde{c}}{e^{\Gamma}-\left(1-q+q\left(1-p_{\text{gen}}\right)^{n}\right)(1-p_{\text{con}})}, \quad \tilde{D}:=\frac{q(1-p_{\text{con}})\tilde{d}}{p_{\text{con}}+q\left(1-\left(1-p_{\text{gen}}\right)^{n}\right)(1-p_{\text{con}})},$$
$$\tilde{P}:=p_{\text{con}}+q\left(1-\left(1-p_{\text{gen}}\right)^{n}\right)(1-p_{\text{con}}),$$

and

$$\tilde{a}:=\sum_{k=1}^{n}a_k\cdot\binom{n}{k}(1-p_{\text{gen}})^{n-k}p_{\text{gen}}^{k}, \quad \tilde{b}:=\sum_{k=1}^{n}b_k\cdot\binom{n}{k}(1-p_{\text{gen}})^{n-k}p_{\text{gen}}^{k},$$
$$\tilde{c}:=\sum_{k=1}^{n}c_k\cdot\binom{n}{k}(1-p_{\text{gen}})^{n-k}p_{\text{gen}}^{k}, \quad \tilde{d}:=\sum_{k=1}^{n}d_k\cdot\binom{n}{k}(1-p_{\text{gen}})^{n-k}p_{\text{gen}}^{k}.$$

Proof. See Appendix 7.7. □

From Theorem 7.1, we see that the availability depends on all the parameters of the system (listed in Table 7.1), including the noise level Γ. The latter may come as a surprise, since one would expect noise to have an impact on the average consumed fidelity but maybe not on the availability, which is only affected by processes that fill or deplete the G memory. These processes are entanglement generation, failed purification, and consumption. In our model, the probability of failed purification depends via (7.4) on the fidelity of the buffered link, which is in turn affected by the level of noise. As a consequence, noise has an indirect effect on the availability.

Theorem 7.2 (Formula for the average consumed fidelity)**.** *The average consumed fidelity of the 1GnB system is given by*

$$\overline{F} = \frac{\tilde{w}F_{\text{new}} + \tilde{x}}{\tilde{y}F_{\text{new}} + \tilde{z}}, \text{ almost surely,} \tag{7.11}$$

with

$$\begin{aligned}
\tilde{w} &:= p_{\text{con}} + q(1-p_{\text{con}})\left(p^*_{\text{gen}} + \frac{1}{4}\tilde{c} - \tilde{d}\right),\\
\tilde{x} &:= \frac{1}{4}\left[e^{\Gamma} - 1 + q(1-p_{\text{con}})\left(-\tilde{a} + 4\tilde{b} - \frac{1}{4}\tilde{c} + \tilde{d}\right)\right],\\
\tilde{y} &:= q(1-p_{\text{con}})\tilde{c},\\
\tilde{z} &:= e^{\Gamma} - 1 + p_{\text{con}} + q(1-p_{\text{con}})\left(p^*_{\text{gen}} - \tilde{a} - \frac{1}{4}\tilde{c}\right),
\end{aligned}$$

where $p^*_{\text{gen}} = 1-(1-p_{\text{gen}})^n$*, and* $\tilde{a}$*,* $\tilde{b}$*,* $\tilde{c}$*, and* $\tilde{d}$ *are given in Theorem 7.1.*

Proof. See Appendix 7.7. □

We note that both A and $\overline{F}$ have been defined as random variables in Definitions 7.2 and 7.3. However, as shown in Theorems 7.1 and 7.2, these quantities are almost surely deterministic functions of the system parameters. For clarity and convenience, we will adopt a slight abuse of notation and treat A and $\overline{F}$ as deterministic functions. This convention will be maintained throughout the remainder of the text.

7.3. BUFFERING SYSTEM DESIGN

In this section, we discuss our main findings after analysing the performance of the 1GnB system. In Subsection 7.3.1, we study the impact of a general purification protocol on the system performance. In particular, it is shown that the availability and the average consumed fidelity are monotonic in the parameter q that determines how frequently the system attempts purification. In the remaining subsections, we investigate how the choice of purification policy impacts the performance of the buffering system, and we provide heuristic rules for the design of a good purification policy.

7.3.1. MONOTONIC PERFORMANCE

Each time a B memory successfully generates entanglement, there is the opportunity to purify the buffered link. This is controlled by the parameter q, which is the probability that, after some fresh links are successfully generated, they are used to attempt purification (otherwise they are discarded). If purification is never attempted ($q = 0$), the fidelity of the buffered link will never be increased, although the buffered link will never be lost to failed purification. If purification is always attempted ($q = 1$), the availability and average consumed fidelity might be affected as follows:

- Purifying more often means risking the loss of buffered entanglement more frequently, since purification can fail. This suggests availability may be decreasing in q. However, many purification protocols have a probability of success that is increasing in the fidelity of the buffered link, F. This means that, when purification is applied more frequently to maintain a high-fidelity link, subsequent purification attempts are more likely to succeed. Consequently, it is not clear that the availability is decreasing in q.
- The fidelity of the buffered link increases after applying several purification rounds. However, if purification is applied too greedily, we may lose a high-quality link and we would have to restart the system with a lower-quality link. If a consumption request then arrives, it would only be able to consume low-quality entanglement. Hence, it is not clear that the average consumed fidelity is increasing in q.

In the following, we address the previous discussion and show that, if purification is always attempted ($q = 1$), the availability is actually minimized, while the average consumed fidelity is maximized. More generally, we show that A and $\overline{F}$ are both monotonic in q, given some reasonable conditions on the jump functions J_k. The following results (Propositions 7.1 and 7.2) may be used to answer an important question about the 1GnB system: *how frequently should we purify the buffered state in order to maximize A (or $\overline{F}$)?* That is, *what value of q optimizes our performance metrics?*

Proposition 7.1. *The availability is a non-increasing function of q, i.e.,*

$$\frac{\partial A}{\partial q} \leq 0. \tag{7.12}$$

Proof. See Appendix 7.10. □

As previously explained, the monotonicity of the availability in q is not a trivial result, and it has fundamental implications. It allows us to derive upper and lower bounds that apply to 1GnB systems using *any* purification policy.

Corollary 7.1. *The availability is bounded as*

$$\frac{p^*_{\text{gen}} \cdot (\gamma + p_{\text{con}})}{\xi + \xi' \cdot p^*_{\text{gen}} + \xi'' \cdot (p^*_{\text{gen}})^2} \leq A \leq \frac{p^*_{\text{gen}}}{p^*_{\text{gen}} + p_{\text{con}}}, \tag{7.13}$$

with $p^*_{\text{gen}} := 1 - (1 - p_{\text{gen}})^n$, $\gamma := e^{\Gamma} - 1$, $\xi := \gamma p_{\text{con}} + p^2_{\text{con}}$, $\xi' := 1 + 2\gamma + (2 - \gamma) p_{\text{con}} - 2p^2_{\text{con}}$, and $\xi'' := 2(1 - p_{\text{con}})^2$. Moreover, the upper bound is tight, and is achieved when $q = 0$ for any purification policy.

7

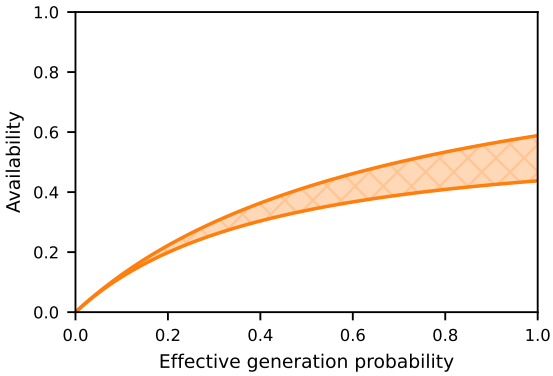

Figure 7.3: **The upper bound on the availability is tight and it converges to the lower bound in the limit of small generation probabilities.** Upper and lower bounds on the availability from (7.13), versus the effective generation probability $p^*_{\text{gen}} = 1-(1-p_{\text{gen}})^n$. The availability can only take values within the shaded region. In this example we use $\Gamma = 1$ and $p_{\text{con}} = 0.7$.

Proof. See Appendix 7.10. □

We refer to p^*_{gen} as the *effective generation probability*, since it corresponds to the probability that at least one new link is generated in a single (multiplexed) attempt. The upper bound from (7.13) only depends on the effective generation probability and the probability of consumption. This bound is achievable with any purification policy: to maximize the availability, it suffices to never purify ($q = 0$). A special case are deterministic policies (those with $p_k(F) = 1, \forall k$), which achieve this bound for any q. This upper bound coincides with the tight upper bound found in previous work for a 1G1B system [50]. Note that the 1G1B analysis from ref. [50] (Chapter 6) was done in continuous time, where rates were used instead of probabilities. In this framework, the maximum availability was $\lambda/(\lambda+\mu)$, where λ was the (non-multiplexed) entanglement generation rate and μ was the consumption rate.

Unlike the upper bound, we note that the lower bound from (7.13) may not be tight. We believe that the availability at $q = 1$ of a policy that always fails purification ($c_k = d_k = 0$, $\forall k$) constitutes a tight lower bound for any other purification policy. We leave this proof as future work.

Figure 7.3 shows the upper and lower bounds for the availability from (7.13) versus p^*_{gen} for two different noise levels. As discussed, only the lower bound is affected by noise. In particular, we have observed that the gap between the bounds is reduced when the noise level increases. Another remarkable feature is that, when p^*_{gen} approaches zero, both upper and lower bounds are equal to $p^*_{\text{gen}}/p_{\text{con}}$ to first order in p_{gen}. Hence, in the limit of small effective generation probabilities, the availability also satisfies

$$A \approx \frac{p^*_{\text{gen}}}{p_{\text{con}}}. \tag{7.14}$$

Proposition 7.2. *The average consumed fidelity is a non-decreasing function of q, i.e.,*

$$\frac{\partial \overline{F}}{\partial q} \geq 0, \tag{7.15}$$

if $J_k(F_{\text{new}}) \geq F_{\text{new}}$, $\forall k \in \mathbb{N}$.

Proof. See Appendix 7.11. □

As previously explained, the monotonicity of $\overline{F}$ in q is not a trivial result. In fact, this behavior is only certain for purification policies composed of protocols that can increase the fidelity of a newly generated link. That is, when k new links are generated, the protocol applied satisfies $J_k(F_{\text{new}}) \geq F_{\text{new}}$. This is a reasonable condition: otherwise, we would be applying purification protocols that decrease the fidelity of new links.

Proposition 7.2 also allows us to derive useful upper and lower bounds that apply to 1GnB systems using any purification policy.

Corollary 7.2. *The average consumed fidelity is bounded as*

$$\frac{\gamma + 4F_{\text{new}} p_{\text{con}}}{4\gamma + 4p_{\text{con}}} \leq \overline{F} \leq \frac{\gamma + 4F_{\text{new}} p_{\text{con}} + 3(1 - p_{\text{con}}) p^*_{\text{gen}}}{4\gamma + 4p_{\text{con}}}, \tag{7.16}$$

with $\gamma := e^{\Gamma} - 1$. Moreover, the lower bound is tight and is achieved when $q = 0$ for any purification policy.

Proof. See Appendix 7.11. □

We see that the tight lower bound from (7.16) does not depend on the number of memories n, the probability of successful entanglement generation p_{gen}, or the purification policy. This is because this bound corresponds to $q = 0$. In such a case, no purification is applied, and the consumed fidelity only depends on the initial fidelity (F_{new}) and the amount of decoherence experienced until consumption (given by Γ and p_{con}).

The bounds on $\overline{F}$ can be used to determine if the parameters of the system need an improvement to meet specific quality-of-service requirements. For example, let us consider Figure 7.4, which shows the bounds for $p_{\text{con}} = 0.7$ and two different values of Γ. If noise is strong ($\Gamma = 1$ in this example), we observe that values of p^*_{gen} below 0.5 yield $\overline{F} < 1/2$, which means that, on average, the consumed link will not be entangled (see Section 6.10.3). Hence, if the consumption request rate is $p_{\text{con}} = 0.7$, we need to increase p^*_{gen} beyond 0.5 (by increasing the number of B memories, n, or the probability of successful entanglement generation, p_{gen}) or to decrease the noise experienced in memory G in order to provide a useful average state. When the noise level is $\Gamma = 0.1$, Figure 7.4 shows that $\overline{F} > 0.85$. Moreover, for $p^*_{\text{gen}} > 0.3$, the upper bound is above F_{new}, which means that a smart choice of purification policy may allow us to buffer entanglement with $\overline{F} > F_{\text{new}}$. Ultimately, this means that, in this regime, an entanglement buffer with faulty memories may be able to keep entanglement at higher fidelities than a perfect memory.

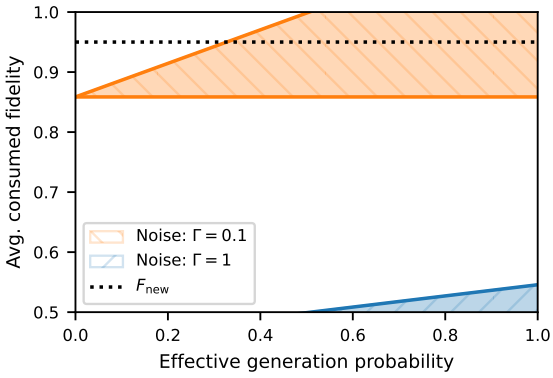

Figure 7.4: **The upper bound on the average consumed fidelity marks unachievable values for any purification policy.** Upper and lower bounds on the average consumed fidelity $\overline{F}$ from (7.16), versus the effective generation probability $p^*_{\text{gen}} = 1-(1-p_{\text{gen}})^n$. $\overline{F}$ can only take values within the shaded region. In this example we use $p_{\text{con}} = 0.7$.

7.4. CHOOSING A PURIFICATION POLICY

In previous studies of entanglement buffering, the choice of purification policy was restricted by the properties of the system. For example, in ref. [50] (Chapter 6 the 1G1B system was studied, where only 2-to-1 purification protocols can be implemented, and the jump function was assumed to be linear in the fidelity of the buffered link. Other works include simplifying assumptions (e.g., in ref. [60], a buffer is studied that employs the purification protocol proposed in ref. [162]). The 1GnB buffering system offers more freedom in the choice of purification protocols. In a 1GnB buffer, each entanglement generation attempt is multiplexed and can generate up to n new links at a time. When $k \leq n$ new links are produced, any $(k+1)$-to-1 purification protocol can in principle be implemented. This provides an extra knob that can be used to tune the performance of the system to the desired values. In this section, we investigate the impact that specific purification policies have on the system and we provide guidelines on how to choose a suitable purification policy. Note that an exhaustive optimization problem would be extremely computationally expensive to solve due to the large space of purification policies – optimizing over a_k, b_k, c_k, d_k is not easy, since it is not certain that every combination of those parameters corresponds to an implementable purification circuit.

7.4.1. SIMPLE POLICIES: IDENTITY, REPLACEMENT, AND CONCATENATION

There are two trivial deterministic policies ($p_k = 1$, $\forall k$) that we will use as a baseline:

- In the *identity policy*, the system does not perform any operation on the buffered link, which yields an output fidelity $J_k(F) = F$, $\forall k > 0$. This is equivalent to setting $q = 0$. As discussed in Section 7.3.1, the identity policy therefore maximizes the availability and minimizes the average consumed fidelity.
- In the *replacement policy*, the system replaces the buffered entangled link by a new

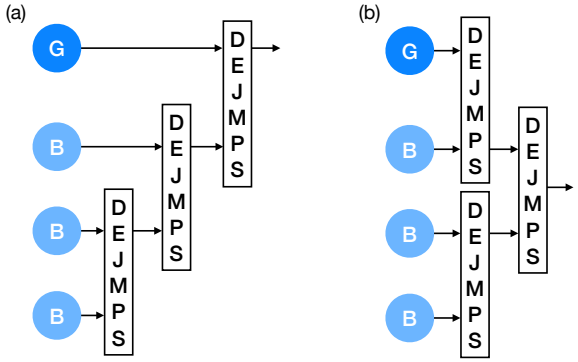

Figure 7.5: **The ordering in a concatenated policy matters.** Example of two different orderings when the buffered link (G) and three newly generated links (B) are used. We call ordering (a) "concatenated DEJMPS". Ordering (b) is often called "nested" [22].

link, yielding an output fidelity $J_k(F) = F_{\text{new}}$, $\forall k > 0$. This corresponds to $a_k = 0$, $b_k = F_{\text{new}} - 1/4$, $c_k = 0$, and $d_k = 1$. Since this policy is deterministic, from the discussion in Section 7.3.1 we find that the replacement policy also provides maximum availability for any value of q. Since $\overline{F}$ is maximized for $q = 1$ (Proposition 7.2), we will only consider a replacement policy that always chooses to replace the link in memory when a new link is generated. That is, the replacement policy implicitly assumes $q = 1$.

Another simple strategy is the *DEJMPS policy*. This policy consists in applying the well-known 2-to-1 DEJMPS purification protocol [53] using the buffered link and a newly generated link as inputs. If more than one link is successfully generated, we use only one of them and discard the rest. We provide the purification coefficients a_k, b_k, c_k, and d_k for this policy in Appendix 7.8.1. One of the main drawbacks of the DEJMPS policy is that it does not take full advantage of the multiplexed entanglement generation, as it only uses one of the newly generated links and discards the rest. A technique that could improve the performance of the policy is *concatenation*, which consists in applying DEJMPS to all links (the buffered one and the newly generated ones) subsequently until only one link remains, which will be stored in memory G. Note that the concatenation of DEJMPS subroutines can be applied using different orderings of the links (see Figure 7.5). This order determines the output fidelity and probability of success [183], which affects the performance of the buffering system. In what follows, we consider the *concatenated DEJMPS* policy, where DEJMPS is applied sequentially to all the newly generated links and the buffered link is used in the last application of DEJMPS, as in Figure 7.5a. In our analysis, we found that different orderings provided qualitatively similar behavior of our two performance metrics (see Appendix 7.12.1 for further details).

Figure 7.6 shows the performance of several policies: identity, replacement, DEJMPS, and concatenated DEJMPS $\times N$. The latter is a policy that concatenates DEJMPS up to N times and discards any extra links: if $k \leq N$ links are generated then k concatenations are performed, and if $k > N$ links are generated, N concatenations are performed. We

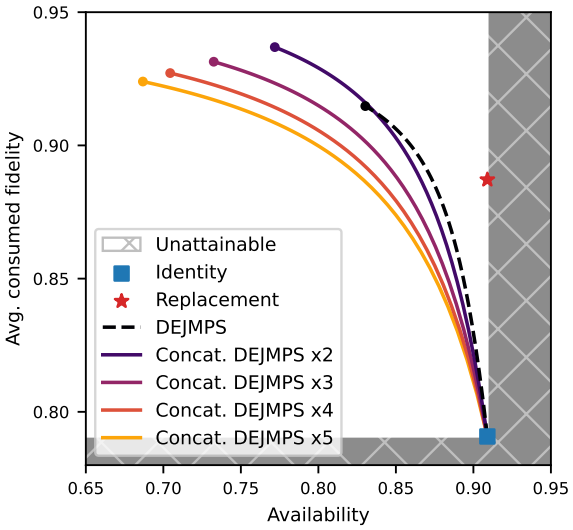

Figure 7.6: **Concatenating simple purification policies decreases A but may increase $\overline{F}$.** Performance of 1GnB systems with different purification policies, in terms of availability A and average consumed fidelity $\overline{F}$. The shaded area corresponds to unattainable values of A and $\overline{F}$ (see (7.13) and (7.16)). Lines and markers show the combinations of A and $\overline{F}$ achievable by different purification policies: identity (square marker), replacement (star marker), DEJMPS (dashed line), and concatenated DEJMPS (solid lines). Concatenation can boost $\overline{F}$ (e.g., the maximum $\overline{F}$ of twice-concatenated DEJMPS is larger than DEJMPS), but excessive concatenation may eventually lead to a drop in $\overline{F}$. Parameter values used in this example: $n = 10$, $p_{\text{gen}} = 0.5$, $F_{\text{new}} = 0.9$ (ρ_{new} is a Werner state), $p_{\text{con}} = 0.1$, and $\Gamma = 0.02$.

note that concatenated DEJMPS ×1 is just the same as the DEJMPS policy. DEJMPS and concatenated DEJMPS are plotted for $q \in [0, 1]$. The maximum average consumed fidelity is indicated with a dot, and it is achieved when $q = 1$. The first observation from this figure is that a higher level of concatenation decreases the availability. This is because it requires multiple DEJMPS subroutines to succeed, which decreases the overall probability of successful purification. However, a higher level of concatenation can significantly increase the average consumed fidelity $\overline{F}$. For example, the maximum $\overline{F}$ that DEJMPS can achieve is 0.915, while concatenated DEJMPS ×2 leads to $\overline{F} = 0.937$ (for $q = 1$). Nevertheless, for the parameter values explored, we also find that increasing the number of concatenations beyond two often reduces both A and $\overline{F}$. This behavior is shown more explicitly in Figure 7.7, where we plot the maximum $\overline{F}$ versus the maximum number of concatenations N. In this example, the number of B memories is $n = 10$, and therefore it is only possible to perform up to 10 concatenated applications of DEJMPS. We observe that $\overline{F}$ is maximized for two concatenations. The same was observed for different parameter values – in some edge cases, $\overline{F}$ increases with more concatenations, although the increase is marginal (see Appendix 7.12.2 for further details). In conclusion, this result shows that even if many new links are successfully generated in parallel, it can sometimes be beneficial to use only one or two of them for purification while discarding the rest.

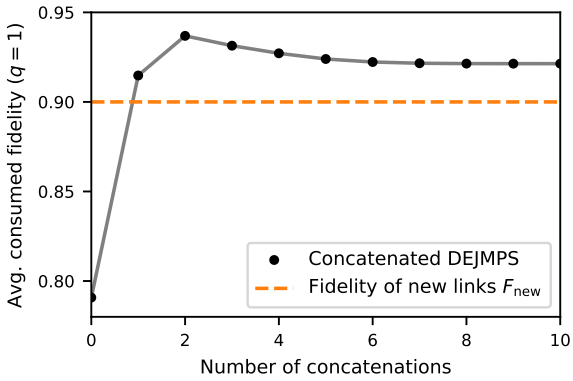

Figure 7.7: **Excessive concatenation worsens the performance.** Maximum average consumed fidelity $\overline{F}$ achieved by a purification policy that concatenates DEJMPS a limited number of times. Zero concatenations corresponds to an identity policy (no purification is performed). One concatenation corresponds to the DEJMPS policy. Excessive concatenation may decrease $\overline{F}$. Parameter values used in this example: $n = 10$, $p_{\text{gen}} = 0.5$, $F_{\text{new}} = 0.9$ (ρ_{new} is a Werner state), $p_{\text{con}} = 0.1$, and $\Gamma = 0.02$.

7.4.2. SIMPLE POLICIES CAN OUTPERFORM COMPLEX POLICIES

In the previous section, we found that implementing a simple 2-to-1 protocol, even when multiple links are generated in the B memories, can provide a better performance than using all of the newly generated links for purification with concatenated 2-to-1 protocols. A follow-up question arises: *what if we employ more sophisticated* $(k+1)$*-to-1 protocols instead of simply concatenating 2-to-1 protocols? Can we then improve the performance of the buffer?* This is the question that we explore next.

Much recent work has focused on the search for optimal purification protocols [95, 106, 159], where optimal protocols are typically defined as those which maximize the output fidelity, or in some cases the success probability. Here, we evaluate the performance of a 1GnB system with some of these protocols, and we find a surprising result: simple protocols like DEJMPS can vastly outperform these more complex protocols in terms of buffering performance. In particular, we consider the bilocal Clifford protocols that maximize the output fidelity, given in ref. [95]. We refer to this policy as the *optimal bilocal Clifford (optimal-bC) policy*. In Appendix 7.8.2, we discuss the details of this policy and provide its purification coefficients a_k, b_k, c_k, and d_k.

Figure 7.8 shows the performance of the optimal-bC policy in comparison to DEJMPS and twice-concatenated DEJMPS. The optimal-bC policy provides a significantly lower availability, A, without providing any advantage in average consumed fidelity, $\overline{F}$. In other words, for any desired A, using DEJMPS or twice-concatenated DEJMPS always provides a larger $\overline{F}$ than the optimal-bC policy. If we want to increase A as much as possible, the replacement policy is better than any other, as discussed earlier. We say that the performance of DEJMPS, twice-concatenated DEJMPS, and replacement forms the *Pareto frontier* [126], which informally is the set of best achievable values for A and $\overline{F}$ for this collection of protocols. We tested different parameter combinations and found that the Pareto frontier was often made of DEJMPS, concatenated DEJMPS, and replacement.

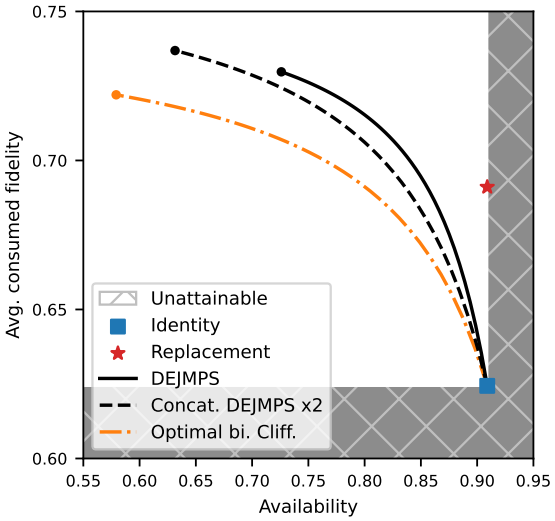

Figure 7.8: **Simple policies perform better despite discarding freshly generated entanglement.** Performance of 1GnB systems with different purification policies, in terms of availability A and average consumed fidelity $\overline{F}$. The shaded area corresponds to unattainable values of A and $\overline{F}$ (see (7.13) and (7.16)). Lines and markers show the combinations of A and $\overline{F}$ achievable by different purification policies: identity (square marker), replacement (star marker), DEJMPS (solid line), twice-concatenated DEJMPS (dashed line), and optimal-bC (dotted line). Parameter values used in this example: $n = 5$, $p_{\text{gen}} = 0.8$, $F_{\text{new}} = 0.7$ (ρ_{new} is a Werner state), $p_{\text{con}} = 0.1$, and $\Gamma = 0.02$.

The reason for these simple policies to outperform the optimal-bC policy is that the optimal bilocal Clifford protocols maximize the output fidelity at the expense of a reduced probability of success. At some point, the sacrifice in the probability of success can outweigh the benefit of a larger output fidelity, thereby reducing the overall performance of the buffer in terms of both A and $\overline{F}$.

Our comparison between simple and optimal purification protocols is by no means an exhaustive study. However, it shows that purification protocols that maximize only the output fidelity (or probability of success) must not be blindly used in more complex systems involving many impacting factors such as decoherence and consumption, such as entanglement buffers. In fact, we find that discarding some of the newly generated links and applying a 2-to-1 protocol can provide larger A and $\overline{F}$ than using all of the links in a more sophisticated purification subroutine. Note that this does not mean that multiplexed entanglement generation is not useful: even if we only employ 2-to-1 protocols, multiplexing boosts the effective entanglement generation rate, which allows for a more frequent purification of the buffered link.

Additionally, we also tested other complex policies that use (suboptimal) k-to-1 protocols, such as the *513 EC policy*, which uses a 5-to-1 protocol based on a $[[5,1,3]]$ quantum error correcting code. In Appendix 7.9, we explain this policy in detail and show that it can outperform DEJMPS and twice-concatenated DEJMPS in some parameter regions.

7.4.3. FLAGS CAN IMPROVE PERFORMANCE

As discussed in the previous sections, concatenating protocols multiple times does not necessarily improve the performance of the buffer (neither in terms of A nor $\overline{F}$). The reason is that, when concatenating, a single failure in one of the purification subroutines (in our examples, DEJMPS) leads to failure of the whole concatenated protocol. This can be easily solved: instead of considering the concatenated protocol as a black box that only succeeds when all subroutines succeed, *what if we condition the execution of each subroutine on the success/failure of previous subroutines?* Consider for example the concatenated protocol from Figure 7.5a. If any of the DEJMPS subroutines fails, the whole protocol fails and the buffered link has to be discarded. However, we can fix this by raising a failure flag whenever any of the first two subroutines fails. If this flag is raised, the third subroutine is not executed and we leave the buffered link untouched. The flagged version of a concatenated protocol has a larger probability of success, but may also have a lower output fidelity. This means that it is not clear a priori what is the impact of flags on the buffer performance. Next, we analyze a simple case in which we conclude that flags can be either beneficial or detrimental depending on the parameters of the system, such as the level of noise Γ, and not only on the purification policy itself.

Let us consider a policy that operates as follows. For simplicity, we assume that newly generated states ρ_{new} are Werner states with fidelity F_{new}. When k new links are generated and there is already a link stored in memory G:

1. If $k = 1$, we apply the replacement protocol, which has coefficients $a_1 = 0$, $b_1 = F_{\text{new}} - 1/4$, $c_1 = 0$, and $d_1 = 1$.

2. If $k \geq 2$, we apply the DEJMPS protocol to two of the fresh links and discard the rest. Then, we replace the link in memory with the output from the DEJMPS subroutine, without checking whether it was successful or not. This means that the output fidelity of the protocol is the same as the output fidelity from the DEJMPS subroutine. Since replacement is deterministic, the success probability of this protocol is also the same as the success probability of the DEJMPS subroutine. The purification coefficients for $k \geq 2$ are therefore given by $a_k = 0$, $b_k = a(\rho_{\text{new}}) \cdot (F_{\text{new}} - 1/4) + b(\rho_{\text{new}})$, $c_k = 0$, and $d_k = c(\rho_{\text{new}}) \cdot (F_{\text{new}} - 1/4) + d(\rho_{\text{new}})$, where a, b, c, and d are the coefficients of the DEJMPS protocol (given in Appendix 7.8.1).

Now, let us consider a flagged variant of the previous policy, with coefficients a'_k, b'_k, c'_k, and d'_k. It works as follows:

1. When $k = 1$ new links are generated, we still apply the replacement protocol.

2. When $k \geq 2$ links are generated, the DEJMPS protocol is applied to two of the fresh links, and the rest are discarded. Then, the link in memory is replaced with the output from the DEJMPS subroutine, but only if the subroutine succeeds (otherwise, the buffered link is left untouched). This protocol is now fully deterministic, since we always end up with a buffered link (either the old buffered link or the output of a successful DEJMPS subroutine). Consequently, $c'_k = 0$, and $d'_k = 1$. The output fidelity of this protocol can be computed as the weighted average between the fidelity of the link in memory and the output fidelity of the DEJMPS subroutine –

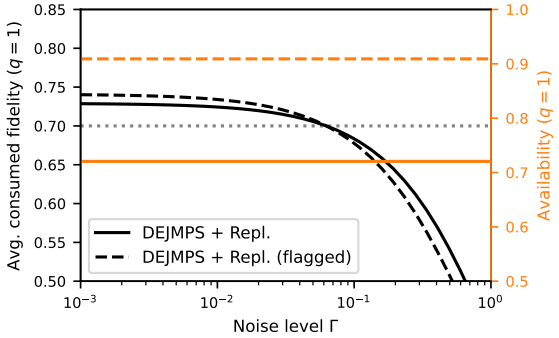

Figure 7.9: **Flagged protocols boost the availability but may decrease the average consumed fidelity.** Availability A and average consumed fidelity $\overline{F}$ versus the noise level Γ, for a 'DEJMPS + Replacement' policy and its flagged version. In the first policy, the buffered link is lost when a DEJMPS subroutine fails. The second policy incorporates a flag that prevents this from happening – it succeeds deterministically at the expense of a lower output fidelity. The flagged policy yields larger A, but may decrease $\overline{F}$ in some parameter regimes (e.g., when Γ is large). Parameter values used in this example: $n = 2$, $p_{\text{gen}} = 1$, $F_{\text{new}} = 0.7$ (ρ_{new} is a Werner state [197]), and $p_{\text{con}} = 0.1$.

the first term must be weighted by the probability of failure of the subroutine, and the second term by the probability of success. Then, the remaining purification coefficients can be computed as $a'_k = 1 - c(\rho_{\text{new}}) \cdot (F_{\text{new}} - 1/4) - d(\rho_{\text{new}})$ and $b'_k = a(\rho_{\text{new}}) \cdot (F_{\text{new}} - 1/4) + b(\rho_{\text{new}})$, where a, b, c, and d are the coefficients of the DEJMPS protocol (given in Appendix 7.8.1).

By introducing the flags, we have created a protocol with probability of success $p'_k = 1 \geq p_k$, where p_k is the probability of success of the original protocol. However, it can be shown that the output fidelity of the flagged protocol is $J'_k(F) \leq J_k(F)$, where J_k is the jump function of the original protocol. This holds when DEJMPS can improve the fidelity of the newly generated links, i.e., when $J(F_{\text{new}}) \geq F_{\text{new}}$, where J is the jump function of DEJMPS. The opposite regime is not interesting, since DEJMPS is decreasing the fidelity of the links and we would be better off not purifying.

As shown in the previous example, internal flags increase the probability of success of purification protocols, which should boost the availability of the buffer. However, flags may have the side effect of reducing the output fidelity, and therefore it is not clear what is their impact on the average consumed fidelity. In Figure 7.9, we show the performance of a 1GnB system using the policy described above, versus the level of noise in memory G. We show A (orange lines) and $\overline{F}$ (black lines) for the original policy (solid lines) and the flagged policy (dashed lines). As expected, the availability is larger for the flagged policy. The behavior of $\overline{F}$ is more interesting. When the level of noise is low, the flagged policy provides better performance, since it prevents high-quality entanglement from being lost to a failed purification. However, when noise is strong, flagging becomes detrimental in terms of $\overline{F}$: the buffer is likely to store low-quality entanglement due to the strong noise, and flags prevent the buffered link from being discarded earlier due to failed purification and being replaced by a fresh link.

In conclusion, internal flags are a solid tool to improve the availability of entanglement buffers based on concatenated purification protocols. However, they may decrease the average consumed fidelity in some parameter regimes. Hence, flagged purification policies should not be assumed to be better than their non-flagged counterparts, and their performance should be carefully evaluated before being adopted.

7.5. OUTLOOK

In this chapter, we have studied the behavior of entanglement buffers with one long-lived memory and n short-lived memories (1GnB system). In particular, we have provided analytical expressions for the two main performance measures: the availability and the average consumed fidelity. These expressions provide valuable insights, such as the fundamental limits to the performance of 1GnB systems discussed earlier.

Since our analytical solutions are not computationally expensive to evaluate, we expect our buffering setup to be easy to incorporate in more complex network architectures, such as quantum repeater chains or even large-scale quantum networks. Additionally, larger buffering systems with multiple long-lived memories, e.g., an mGnB setup, can be implemented with multiple 1GnB systems in parallel.

Due to the vast freedom in the choice of purification policy, there are multiple ways in which our analysis of purification strategies for entanglement buffers can be extended. Notably, determining the optimal ordering in which simple protocols should be applied to newly generated links (e.g., concatenated, nested [22], or banded [183]) is left as future work. Additionally, finding policies that optimize availability or average consumed fidelity remains an important open question.

7.6. [APPENDIX] A NOTE ON THE VIEWPOINT

In this appendix, we provide three further ways to compute the performance metrics A and $\overline{F}$. The initial (and most natural) definitions of the performance metrics (see Definitions 7.2 and 7.3) consist in averages from the viewpoint of the network user, who consumed the links. In Lemma 7.1, we show that the averaging may only be done over a single cycle of the renewal process. In Lemma 7.3, we show that the performance metrics can also be computed as limiting values when time goes to infinity. Lastly, in Lemma 7.2, it is shown that one may compute the metrics by averaging over time, regardless of consumption arrival times.

We denote the arrival time of the j-th consumption request as $T_{\text{con}}^{(j)}$. From now on, we write

$$\mathbb{1}_{\text{l.e.}}(F) \equiv \mathbb{1}_{\text{link exists}}(F) = \begin{cases} 1 \text{ if } F > 0, \\ 0 \text{ if } F = 0. \end{cases} \tag{7.17}$$

In the following, we let $\mathbb{N}_0$ denote the natural numbers containing zero, and $\mathbb{N} = \mathbb{N}_0 \setminus \{0\}$. Recall that $F = \{F(t), t \in \mathbb{N}_0\}$ is a discrete-time stochastic process. The value $F(t)$ is defined to be the fidelity at the *beginning* of the time step $[t, t+1)$. Then, since consumption removes the link from the G memory, at each consumption time we have $F(T_{\text{con}}^{(j)}) = 0$.

However, the consumed fidelity at this time depends on the value of the fidelity at time $T_{\text{con}}^{(j)} - 1$. We therefore introduce some new notation to more easily treat this issue.

In order to do this, we firstly note that associated with F there is an equivalent continuous-time stochastic process $\{F_{\text{cont}}(s) : s \geq 0\}$ that is obtained from F with the following procedure: given $t \in \mathbb{N}$,

(i) if $F(t) > 0$, then for $s \in [t, t+1)$, $F_{\text{cont}}(s)$ may be deduced by applying decoherence (7.1) to $F(t)$;

(ii) if $F(t) = 0$, then $F_{\text{cont}}(s) = 0$ for $s \in [t, t+1)$.

Conversely, F may be obtained from F_{cont} by taking its values at integer times.

From F_{cont}, for $t \in \mathbb{N}$, we define another discrete time process F^-,

$$F^- := \left\{F_{\text{cont}}(t^-) : t \in \mathbb{N}\right\}, \tag{7.18}$$

where t^- denotes taking the left-hand limit. In particular, the consumed fidelity $F^-(t)$ takes the value of the fidelity at the *end* of the time step $[t-1, t)$. The values of F^- may also be deduced directly from F as

$$F^-(t) = \begin{cases} e^{-\Gamma}\left(F(t-1) - \frac{1}{4}\right) + \frac{1}{4}, \text{ if } F(t-1) > 0, \\ 0, \text{ if } F(t-1) = 0. \end{cases} \tag{7.19}$$

We note that the evolution of $\{F^-(t), t \in \mathbb{N}\}$ may be deduced directly from $\{F(t), t \in \mathbb{N}\}$ via (), and vice-versa. The value $F^-(t)$ may be interpreted as the state of the system 'just before' time t, and $F(t)$ the state 'just after'. Each completely captures the behavior of the 1GnB system.

We then restate the original definitions 7.2 and 7.3 of availability, A, and average consumed fidelity, $\overline{F}$, below.

Definition 7.4 (Performance metrics, viewpoint of network user)**.** *We have*

$$A = \lim_{m\to\infty} \frac{1}{m} \sum_{j=1}^{m} \mathbb{1}_{\text{l.e.}}\left(F^-(T_{\text{con}}^{(j)})\right), \tag{7.20}$$

and

$$\overline{F} = \lim_{m\to\infty} \frac{\sum_{j=1}^{m} F^-(T_{\text{con}}^{(j)}) \cdot \mathbb{1}_{\text{l.e.}}\left(F^-(T_{\text{con}}^{(j)})\right)}{\sum_{j=1}^{m} \mathbb{1}_{\text{l.e.}}\left(F^-(T_{\text{con}}^{(j)})\right)}. \tag{7.21}$$

We now present a second way to compute the performance metrics, which is the form that is used to derive the solutions for A and $\overline{F}$ in Theorems 7.1 and 7.2 (see Appendix 7.7). To show this result, we use the fact that $F(t)$ is a regenerative process. Informally, every time the link in the G memory is removed from the system, the process 'starts again', in the sense that the stochastic properties from that point onwards are the same as when starting from any other time when the G memory is empty. This stems from the fact that entangled link generation and consumption request arrivals are assumed to be Markovian.

Definition 7.5 (Regenerative process, informal)**.** *A regenerative process* $\{X(t), t \geq 0\}$ *is a stochastic process with the following properties: there exists a random variable* $V_1 > 0$ *such that*

(i) $\{X(t+V_1), t \geq 0\}$ *is independent of* $\{X(t), t \leq V_1\}$ *and* V_1*;*

(ii) $\{X(t+V_1), t \geq 0\}$ *is stochastically equivalent to* $\{X(t), t \geq 0\}$ *(i.e., these two processes have the same joint distributions).*

For a formal definition of a regenerative process, see, e.g., [170]. If the process is regenerative, it may also be shown that there is a sequence of regeneration cycles $V_0 = 0$, $\{V_k\}$ such that the sequence regenerates at each cycle, i.e., $\{X(t), t \geq 0\}$ and $\{X(t+V_k), t \geq 0\}$ are stochastically equivalent.

We now show that our process F is regenerative. Let us assume the system starts when a new link is freshly generated and moved to the G memory, such that $F(0) = F_{\text{new}}$. The system then evolves as follows: the link in the G memory may undergo some purification rounds, between which it is subject to decoherence, and then is eventually removed from the G memory after time $T_{\text{occ}}^{(1)}$ due to either purification failure or consumption. The time $T_{\text{occ}}^{(1)}$ is the time during which the G memory is occupied. In particular,

$$T_{\text{occ}}^{(1)} := \min\{t : F(t) = 0\}. \tag{7.22}$$

After the link is removed, the system will then attempt entanglement generation until a successful generation. Let the time from which the G memory is emptied until a new link is produced be $T_{\text{gen}}^{(1)}$. By the assumption that entanglement generation attempts are independent and Bernoulli, $T_{\text{gen}}^{(1)} \sim \text{Geo}(1-(1-p_{\text{gen}})^n)$. When a fresh link is generated at time $t = T_{\text{occ}}^{(1)} + T_{\text{gen}}^{(1)}$, we have $F(T_{\text{occ}}^{(1)} + T_{\text{gen}}^{(1)}) = F_{\text{new}}$ and, from this time on, the process behaves equivalently to how it did from time $t = 0$. Letting $V_1 = T_{\text{occ}}^{(1)} + T_{\text{gen}}^{(1)}$, we see that $F(t)$ is regenerative. All regeneration cycles $\{V_k\}$ may each be split into two phases: we have $V_k = T_{\text{occ}}^{(k)} + T_{\text{gen}}^{(k)}$, where $T_{\text{occ}}^{(k)}$ is the time during which the memory is occupied, and $T_{\text{gen}}^{(k)}$ is the time during which the memory is empty and entanglement generation is being attempted. We note that since F^- is in one-to-one correspondence with F via $()$, then F^- is also regenerative with the same cycle lengths.

For the following results, we note two important properties of the process $\{V_k\}$. Firstly, the mean cycle length $\mathbb{E}[V_1] = \mathbb{E}[T_{\text{occ}}^{(1)}] + \mathbb{E}[T_{\text{gen}}^{(1)}]$ is finite: this may be seen by the fact that $T_{\text{gen}}^{(1)}$ is geometrically distributed (and therefore $\mathbb{E}[T_{\text{gen}}^{(1)}] < \infty$) and that $T_{\text{occ}}^{(1)}$ is bounded above by the time until the next consumption request, which is geometrically distributed, and so $\mathbb{E}[T_{\text{occ}}^{(1)}] \leq \mathbb{E}[T_{\text{con}}^{(1)}] < \infty$. The second important property is that the $\{V_k\}$ are *aperiodic*, which means that V_1 takes values in a set of integers that have greatest common denominator equal to one. Again, this may be seen by the fact that consumption and entanglement generation are assumed to be geometric. If $p_{\text{gen}} < 1$, the value of V_1 has a non-zero probability of taking any value in $\mathbb{N} \setminus \{1\}$ and therefore satisfies this property. The same holds if $p_{\text{gen}} = 1$, and there is a non-zero probability of either no purification or successful purification. The cases where the $\{V_k\}$ are periodic may be accounted for separately:

(A) If $p_{\text{gen}} = 1$ and $p_{\text{con}} = 1$, a link will deterministically be generated when in the empty state, and deterministically consumed in the following time step. The fidelity $F(t)$ then deterministically alternates between 0 and F_{new}, and the cycle length is always two. We therefore have

$$A = \frac{1}{2}, \quad \overline{F} = e^{-\Gamma}\left(F_{\text{new}} - \frac{1}{4}\right) + \frac{1}{4}. \tag{7.23}$$

(B) If $p_{\text{gen}} = 1$, $q = 1$ and $c_k = d_k = 0$, then we have deterministic link generation, and the system always decides to purify. However, purification always fails. The fidelity then again deterministically alternates between 0 and F_{new}, and the cycle length is two. We note that even if purification is always attempted and always fails, then if a consumption request arrives, this will take priority over purification and the link will be consumed with fidelity $e^{-\Gamma}\left(F_{\text{new}} - \frac{1}{4}\right) + \frac{1}{4}$. Then, $\overline{F}$ will also take this value. Moreover, by applying the PASTA property in discrete time [125], we have $A = 1/2$. Our metrics then take the values (7.23), as in case (A).

We note that our formulae, as given in Theorems 7.1 and 7.2, still hold for the above cases. The solutions for edge case (A) are obtained by inputting $p_{\text{gen}} = 1$ and $p_{\text{con}} = 1$. Edge case (B) can be dealt with in the same way: take $p_{\text{gen}} = 1$, $q = 1$ and the limit $c_k, d_k \to 0$. Note that the jump function (7.3) must still be well-defined, and so necessarily we must also take $a_k, b_k \to 0$. We then obtain (7.23). Although the proof in the general case may not be immediately applied in these cases, our formula still holds.

7

Lemma 7.1 (Performance metrics, single cycle)**.** *Suppose that the 1GnB system parameters are not in edge cases (A) or (B). The performance metrics in Definition 7.4 may be written in terms of the properties of a single cycle:*

$$A = \frac{\mathbb{E}[T^{(1)}_{\text{occ}}]}{\mathbb{E}[T^{(1)}_{\text{occ}}] + \mathbb{E}[T^{(1)}_{\text{gen}}]} \ \textit{a.s.} \tag{7.24}$$

and

$$\overline{F} = \mathbb{E}[F^{-}(T^{(1)}_{\text{occ}})|C_1] \ \textit{a.s.} \tag{7.25}$$

where C_1 is the event where the first link is removed due to consumption (and not failed purification), or equivalently $C_1 \equiv \{T^{(1)}_{\text{occ}} = T^{(1)}_{\text{con}}\}$.

Proof. Let F^{-}_{∞} be a random variable with distribution given by

$$\mathrm{P}\left(F^{-}_{\infty} \in B\right) = \lim_{t\to\infty} \frac{1}{t} \sum_{s=1}^{t} \mathbb{1}_B\left(F^{-}(s)\right). \tag{7.26}$$

Then, as F^{-} is a regenerative process with finite mean and aperiodic cycle length, by, e.g., part (a) of Theorem 1 from [192], the above quantity exists and may be computed in terms of the properties of a single cycle as

$$\mathrm{P}\left(F^{-}_{\infty} \in B\right) = \frac{1}{\mathrm{E}[V_1]} \mathbb{E}\left[\sum_{s=1}^{V_1} \mathbb{1}_B\left(F^{-}(s)\right)\right]. \tag{7.27}$$

Letting B be the event where a link is present in the G memory, we then see that

$$\mathrm{P}\left(F_\infty^- > 0\right) = \frac{1}{\mathrm{E}[V_1]}\mathbb{E}\left[\sum_{s=1}^{V_1} \mathbb{1}_{\text{l.e.}}\left(F^-(s)\right)\right] \tag{7.28}$$

$$= \frac{1}{\mathbb{E}[T_{\text{occ}}^{(1)}] + \mathbb{E}[T_{\text{gen}}^{(1)}]} \cdot \mathbb{E}\left[T_{\text{occ}}^{(1)}\right]. \tag{7.29}$$

We now show that the above expression is equal to A. Since the interarrival times of consumption requests are i.i.d. and follow a geometric distribution, we make use of the PASTA property in discrete time [125] to see that the availability from the point of view of the consumer in Definition 7.2 is equal to the time average as given above, i.e.,

$$A = \lim_{t\to\infty} \frac{1}{t}\sum_{s=1}^{t} \mathbb{1}_{\text{l.e.}}\left(F^-(s)\right) = \mathrm{P}(F_\infty^- > 0), \text{ a.s.} \tag{7.30}$$

Then, (7.24) is shown by combining (7.30) with (7.29).

We now show the identity for $\overline{F}$. For this, we also use the regerative property. We define $W_0 = 0$ and W_k to be the time at which the k-th cycle ends,

$$W_k := \sum_{j=1}^{k} V_j. \tag{7.31}$$

Then, the sequence of times at which the link is removed from the G memory is

$$\{W_{k-1} + T_{\text{occ}}^{(k)}\}_{k\geq 1}. \tag{7.32}$$

We then define the subsequence

$$\{W_{i_k-1} + T_{\text{occ}}^{(i_k)}\}_{k\geq 1} \tag{7.33}$$

to be the times at which link removal is due to consumption (and not purification failure). We recall that in our model, when a consumption request arrives, it immediately removes the link from the G memory. Then, (7.33) are precisely the times at which consumption requests arrive to find a link in the G memory. In particular,

$$\{W_{i_k-1} + T_{\text{occ}}^{(i_k)}\}_{k\geq 1} = \left\{T_{\text{con}}^{(k)} : F^-\left(T_{\text{con}}^{(k)}\right) > 0\right\}_{k\geq 1}, \tag{7.34}$$

recalling that $\{T_{\text{con}}^{(k)}\}$ is the sequence of arrival times for consumption requests. Recalling Definition (7.4) of $\overline{F}$, we then see that

$$\overline{F} = \lim_{m\to\infty} \frac{\sum_{k=1}^{m} F^-(T_{\text{con}}^{(k)}) \cdot \mathbb{1}_{\text{l.e.}}\left(F^-(T_{\text{con}}^{(k)})\right)}{\sum_{k=1}^{m} \mathbb{1}_{\text{l.e.}}\left(F^-(T_{\text{con}}^{(k)})\right)}$$

$$= \lim_{m\to\infty} \frac{\sum_{k=1}^{M(m)} F^-\left(W_{i_k-1} + T_{\text{occ}}^{(i_k)}\right)}{\sum_{k=1}^{M(m)} 1}, \tag{7.35}$$

7

where we have used the identity (7.34), and defined $M(m) \leq m$ as

$$M(m) = \left| \left\{ T_{\text{con}}^{(k)} : F^- \left(T_{\text{con}}^{(k)} \right) > 0,\ k \leq m \right\} \right|.$$

Then, $M(m)$ is the number of consumption requests up to time $T_{\text{con}}^{(m)}$ that arrive when a link is stored in memory. We now show that $\lim_{m\to\infty} M(m) = \infty$ a.s. so that we can apply SLLN to the above expression. To see this, recall that $\{V_k\}_{k\geq 1}$ are the i.i.d. interarrival times of a renewal process $N(t) = \sup\{k : W_k \leq t\}$. Since $|\mathbb{E}[V_1]| < \infty$, we have that $\lim_{t\to\infty} N(t) = \infty$ a.s. (see 10.1.2 of [76]). Within each of these cycles, the link is removed from memory exactly once. The probability that this is due to consumption is bounded below by $p_{\text{con}} > 0$, because for each cycle it is possible to consume directly after link generation, which occurs with probability p_{con}. Recalling the sequence of times when the link is removed due to consumption as given in (7.34), the number of these events may therefore be bounded below by a subsequence

$$\{W_{j_k-1} + T_{\text{occ}}^{(j_k)}\}_{k\geq 1} \subseteq \{W_{i_k-1} + T_{\text{occ}}^{(i_k)}\}_{k\geq 1} \tag{7.36}$$

such that the $j_k - j_{k-1}$ is geometrically distributed with parameter $\eta \geq p_{\text{con}}$. We therefore see that

$$\lim_{k\to\infty} |\{W_{j_k-1} + T_{\text{occ}}^{(j_k)}\}_{k\geq 1}| = \infty \text{ a.s.} \tag{7.37}$$

and therefore by (7.36), the total number of times when the link is consumed diverges to infinity almost surely. From (7.35), we then have

$$\begin{aligned} \overline{F} &\overset{\text{a.s.}}{=} \lim_{M\to\infty} \frac{1}{M} \sum_{k=1}^{M} F^- \left(W_{i_k-1} + T_{\text{occ}}^{(i_k)} \right) \\ &\overset{\text{a.s.}}{=} \mathbb{E}\left[F^-(T_{\text{occ}}^{(1)}) | C_1 \right], \end{aligned}$$

where we have used the fact that the sequence $\{F^-(W_{i_k-1} + T_{\text{occ}}^{(i_k)})\}_{k\geq 1}$ is i.i.d. since the process is regenerative, and the strong law of large numbers. □

In the final lemma of this section, we see that the above metrics are equal to the time averages over the whole process. This follows from a version of the well-known PASTA property (Poisson Arrivals See Time Averages) in queuing theory [125], which we can employ because the arrival of consumption requests in each time step is assumed to be a Bernoulli process.

Lemma 7.2 (Performance metrics, time average)**.** *Suppose that the 1GnB system parameters are not in edge cases (A) or (B). The performance metrics in Definition 7.4 may be computed using an average over time, i.e.,*

$$A = \lim_{t\to\infty} \frac{1}{t} \sum_{s=1}^{t} \mathbb{1}_{\text{l.e.}}(F^-(s)), \tag{7.38}$$

and

$$\overline{F} = \lim_{t\to\infty} \frac{\sum_{s=1}^{t} F^-(s) \cdot \mathbb{1}_{\text{l.e.}}(F^-(s))}{\sum_{s=1}^{t} \mathbb{1}_{\text{l.e.}}(F^-(s))}. \tag{7.39}$$

Proof. The identity for A is a direct application of the PASTA property in discrete time [125], which we also saw in the proof of Lemma 7.4.

For the second equality, from (7.21) we firstly rewrite $\overline{F}$ as

$$\overline{F} = \lim_{m\to\infty} \frac{\frac{1}{m}\sum_{j=1}^{m} F^{-}\left(T_{\text{con}}^{(j)}\right)}{\frac{1}{m}\sum_{j=1}^{m} \mathbb{1}_{\text{l.e.}}\left(F^{-}\left(T_{\text{con}}^{(j)}\right)\right)} = \frac{\overline{F}_{\text{tot}}}{A}, \tag{7.40}$$

where

$$\overline{F}_{\text{tot}} := \lim_{m\to\infty} \frac{1}{m}\sum_{j=1}^{m} F^{-}\left(T_{\text{con}}^{(j)}\right)$$

is the average fidelity seen by users, without conditioning on the fidelity being nonzero. In (7.40), we have removed the indicator function from the sum in the numerator by recalling that $F^{-}(T_{\text{con}}^{(j)}) = 0$ if the j-th consumption request does not find a link in memory. Then, since $\overline{F}_{\text{tot}} = \overline{F} \cdot A$ and by Lemma 7.4 both $\overline{F}$ and A converge, the PASTA property can be applied and we have that

$$\overline{F}_{\text{tot}} = \lim_{t\to\infty} \frac{1}{t}\sum_{s=1}^{t} F^{-}(s), \text{ a.s.} \tag{7.41}$$

Then,

$$\overline{F} = \frac{\overline{F}_{\text{tot}}}{A} = \frac{\lim_{t\to\infty}\frac{1}{t}\sum_{s=1}^{t} F^{-}(s)}{\lim_{t\to\infty}\frac{1}{t}\sum_{s=1}^{t} \mathbb{1}_{\text{l.e.}}\left(F^{-}(s)\right)} \tag{7.42}$$

$$= \lim_{t\to\infty} \frac{\sum_{s=1}^{t} F^{-}(s)}{\sum_{s=1}^{t} \mathbb{1}_{\text{l.e.}}\left(F^{-}(s)\right)} \tag{7.43}$$

$$= \lim_{t\to\infty} \frac{\sum_{s=1}^{t} F^{-}(s)\mathbb{1}_{\text{l.e.}}\left(F^{-}(s)\right)}{\sum_{s=1}^{t} \mathbb{1}_{\text{l.e.}}\left(F^{-}(s)\right)} \text{ a.s.} \tag{7.44}$$

□

In the following, we show that our performance metrics may be computed as limiting values of properties of $F(t)$. Note that this was the definition used in Chapter 6 and in ref. [50].

Lemma 7.3 (Performance metrics, limiting values)**.** *Suppose that the 1GnB system parameters are not in edge cases (A) or (B). Then, our performance metrics may be computed as*

$$A = \lim_{t\to\infty} \mathrm{P}(F^{-}(t) > 0) \ \textit{a.s.} \tag{7.45}$$

$$\overline{F} = \lim_{t\to\infty} \mathbb{E}\left[F^{-}(t) | F^{-}(t) > 0\right] \ \textit{a.s.} \tag{7.46}$$

Proof. Since $F^-(t)$ is a regenerative process with finite mean and an aperiodic cycle length, it follows that the limiting distribution is well-defined in the following sense. As in the proof of Lemma 7.4, we let F^-_∞ be a random variable with distribution given by (7.26). Then, by, e.g., parts (a) and (b) of Theorem 1 of [192], we have

$$\lim_{t\to\infty} \mathrm{P}\left(F^-(t) \in B\right) = \mathrm{P}(F^-_\infty \in B). \tag{7.47}$$

We therefore see that

$$\lim_{t\to\infty} \mathrm{P}\left(F^-(t) > 0\right) = \mathrm{P}(F^-_\infty > 0) = A, \tag{7.48}$$

where we have used the identity for A which we saw in (7.30) in the proof of Lemma 7.4. This shows (7.45).

To show the identity for $\overline{F}$, we make use of the renewal-reward theorem (see, e.g., 10.5.1 of [76]). From the previous discussion, associated with the regenerative process $\{F(t), t \in \mathbb{N}\}$ with cycle times $\{W_k\}$, there is a renewal process $N(t) = \sup\{k : W_k \le t\}$. We then define the reward $\tilde{R}_k$ as the sum of fidelity over the k-th cycle,

$$\tilde{R}_k = \sum_{t=W_{k-1}+1}^{W_k} F^-(t). \tag{7.49}$$

Then, the cumulative reward up to time t is given by

$$\tilde{C}(t) = \sum_{s=1}^{t} F^-(s) \tag{7.50}$$

$$= \sum_{k=1}^{N(t)} \tilde{R}_k + E(t), \tag{7.51}$$

where we have defined

$$E(t) = \sum_{s=W_{N(t)}+1}^{t} F^-(s) \tag{7.52}$$

to be the remainder of the reward that is not contained in a full cycle. Then, we see that

$$\begin{aligned} \frac{\tilde{C}(t)}{t} &\le \frac{1}{t} \sum_{k=1}^{N(t)+1} \tilde{R}_k \\ &= \frac{\sum_{k=1}^{N(t)+1} \tilde{R}_k}{N(t)+1} \cdot \frac{N(t)+1}{t}. \end{aligned} \tag{7.53}$$

We will now the strong law of large numbers (SLLN) for both terms in the above product. In particular, the convergence of $(N(t)+1)/t$ may be seen by noticing that

$$\frac{\sum_{k=1}^{N(t)} V_k}{N(t)} \cdot \frac{N(t)}{N(t)+1} < \frac{t}{N(t)+1} \le \frac{\sum_{k=1}^{N(t)+1} V_k}{N(t)+1} \tag{7.54}$$

and using SLLN shows that the upper and lower bound converge to $\mathbb{E}[V_1]$. From (7.53), we therefore see that

$$\lim_{t\to\infty}\frac{\tilde{C}(t)}{t} \leq \frac{\mathbb{E}\left[\tilde{R}_1\right]}{\mathbb{E}[V_1]} \tag{7.55}$$

$$= \frac{\mathbb{E}\left[\sum_{t=1}^{V_1} F^-(t)\right]}{\mathbb{E}[V_1]} \text{ a.s.} \tag{7.56}$$

Similarly,

$$\lim_{t\to\infty}\frac{\tilde{C}(t)}{t} \geq \lim_{t\to\infty}\frac{\sum_{k=1}^{N(t)}\tilde{R}_k}{N(t)+1}\cdot\frac{N(t)}{t} \tag{7.57}$$

$$= \frac{\mathbb{E}\left[\tilde{R}_1\right]}{\mathbb{E}[V_1]} \text{ a.s.} \tag{7.58}$$

Combining (7.41), (7.50), (7.56) and (7.58), we therefore see that

$$\overline{F}_{\text{tot}} = \lim_{t\to\infty}\frac{1}{t}\sum_{s=1}^{t} F^-(s) = \lim_{t\to\infty}\frac{\tilde{C}(t)}{t} = \frac{\mathbb{E}\left[\sum_{t=1}^{V_1} F^-(t)\right]}{\mathbb{E}[V_1]} \text{ a.s.} \tag{7.59}$$

Moreover, using part (b) of Theorem 1 from [192] , we see that

$$\lim_{t\to\infty}\mathbb{E}[F^-(t)] = \frac{\mathbb{E}\left[\sum_{t=1}^{V_1} F^-(t)\right]}{\mathbb{E}[V_1]}, \tag{7.60}$$

and therefore $\overline{F}_{\text{tot}} = \lim_{t\to\infty}\mathbb{E}[F^-(t)]$. Then, we have

$$\lim_{t\to\infty}\mathbb{E}[F^-(t)|F^-(t)>0] = \lim_{t\to\infty}\frac{\mathbb{E}[F^-(t)\mathbb{1}_{\text{l.e.}}(F^-(t))]}{\mathrm{P}(F^-(t)>0)} \tag{7.61}$$

$$= \lim_{t\to\infty}\frac{\mathbb{E}[F^-(t)]}{\mathrm{P}(F^-(t)>0)} \tag{7.62}$$

$$= \frac{\overline{F}_{\text{tot}}}{A} = \overline{F}. \tag{7.63}$$

□

7.7. [APPENDIX] DERIVATION OF FORMULAE FOR PERFORMANCE METRICS

In this appendix, we prove Theorems 7.1 and 7.2, which contain the formulae for the availability and the average consumed fidelity of the 1GnB system.

For these derivations, we work with the following change of variable.

Definition 7.6 (Shifted fidelity)**.** *The* shifted fidelity H *of the 1GnB system is given by*

$$H := F - \frac{1}{4}, \tag{7.64}$$

where F is the fidelity of the link in the G memory.

This will simplify our calculations because under decoherence, the shifted fidelity changes due to a multiplicative exponential factor. In particular, given an initial value h of the shifted fidelity, after t time steps this reduces to

$$h \to e^{-\Gamma t} h. \tag{7.65}$$

We see that the shifted fidelity does not inherit linear terms under decoherence, in contrast to the fidelity, which decays according to (7.1). This will simplify our derivations.

After successful $(k+1)$-to-1 purification, the value h of the shifted fidelity undergoes a jump given by

$$\tilde{J}_k(h) \coloneqq J_k\left(h + \frac{1}{4}\right) - \frac{1}{4} = \frac{a_k h + b_k}{c_k h + d_k} \tag{7.66}$$

where we have used (7.3). Similarly, the probability of successful purification is

$$\tilde{p}_k(h) \coloneqq p_k\left(h + \frac{1}{4}\right) = c_k h + d_k. \tag{7.67}$$

Therefore, $\tilde{J}_k$ and $\tilde{p}_k$ are the jump function and success probability of the corresponding purification events for the shifted fidelity.

Finally, we notice that the range for the fidelity $F \in [0,1]$ translates to $H \in \left[-\frac{1}{4}, \frac{3}{4}\right]$. In particular, we have $H < 0$ if and only if there is no link in the G memory.

We have fully characterized the dynamics of the shifted fidelity in 1GnB (decoherence, purification, and link removal). Our two key performance metrics may then be rewritten in terms of H. Recall that with the assumption $F_{\text{new}} > 1/4$, and the depolarizing decoherence model (7.1), a link exists at time t if and only if $F(t) > 1/4$, or equivalently $H(t) > 0$. Let us again denote the indicator function when acting on the shifted fidelity as

$$\mathbb{1}_{\text{link exists}}(H) \equiv \mathbb{1}_{\text{l.e.}}(H) = \begin{cases} 1 \text{ if } H \geq 0, \\ 0 \text{ if } H < 0. \end{cases}$$

Recalling Definition 7.2, the availability may then be written as

$$A = \lim_{m\to\infty} \frac{1}{m} \sum_{j=1}^{m} \mathbb{1}_{\text{l.e.}}\left(H(T_{\text{con}}^{(j)})\right). \tag{7.68}$$

Recalling Definition 7.3, the average consumed fidelity may be rewritten as

$$\begin{aligned} \overline{F} &= \lim_{m\to\infty} \frac{\sum_{j=1}^{m} F(T_{\text{con}}^{(j)}) \cdot \mathbb{1}_{\text{l.e.}}\left(F(T_{\text{con}}^{(j)})\right)}{\sum_{j=1}^{m} \mathbb{1}_{\text{l.e.}}\left(F(T_{\text{con}}^{(j)})\right)} \\ &= \lim_{m\to\infty} \frac{\sum_{j=1}^{m} \left(\frac{1}{4} + H(T_{\text{con}}^{(j)})\right) \cdot \mathbb{1}_{\text{l.e.}}\left(H(T_{\text{con}}^{(j)})\right)}{\sum_{j=1}^{m} \mathbb{1}_{\text{l.e.}}\left(H(T_{\text{con}}^{(j)})\right)} \\ &= \lim_{m\to\infty} \left[\frac{1}{4} + \frac{\sum_{j=1}^{m} H(T_{\text{con}}^{(j)}) \cdot \mathbb{1}_{\text{l.e.}}\left(H(T_{\text{con}}^{(j)})\right)}{\sum_{j=1}^{m} \mathbb{1}_{\text{l.e.}}\left(H(T_{\text{con}}^{(j)})\right)}\right] \\ &= \frac{1}{4} + \overline{H}. \end{aligned} \tag{7.69}$$

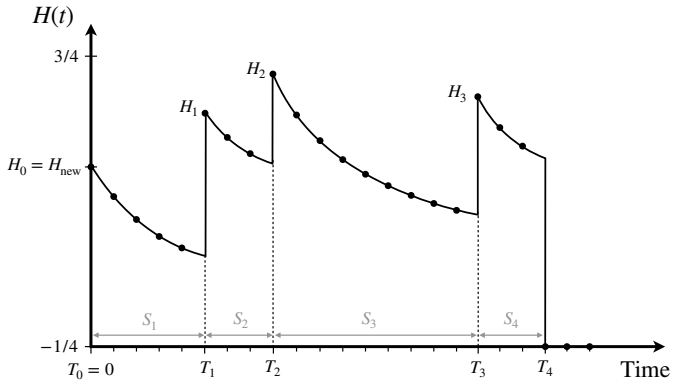

Figure 7.10: **Example dynamics of shifted fidelity in the first cycle of 1GnB.** We assume that $H(0) = H_{\text{new}}$, or equivalently that a freshly generated link is transferred to memory at time $t = 0$. The $\{T_i\}_{i\geq 0}$ are defined to be the times at which there are changes in the (shifted) fidelity that are not due to decoherence. We let T_{occ} be the first time at which the link is removed from the G memory. In the example, $T_{\text{occ}} = T_4$.

We have now written $\overline{F}$ in terms of $\overline{H}$, where

$$\overline{H} := \lim_{m\to\infty} \left[\frac{\sum_{j=1}^{m} H(T_{\text{con}}^{(j)}) \cdot \mathbb{1}_{\text{l.e.}}\left(H(T_{\text{con}}^{(j)})\right)}{\sum_{j=1}^{m} \mathbb{1}_{\text{l.e.}}\left(H(T_{\text{con}}^{(j)})\right)} \right] \tag{7.70}$$

is the average consumed *shifted* fidelity. Finding a formula for $\overline{F}$ then reduces to finding a formula for $\overline{H}$.

From now on, we will assume that the system starts with shifted fidelity $H(0) = H_{\text{new}}$, where

$$H_{\text{new}} := F_{\text{new}} - \frac{1}{4} \tag{7.71}$$

is the state of the G memory immediately after transferring a freshly generated link into memory. Note that H_{new} is a constant, as newly generated links are assumed to be identical. The subsequent dynamics of the system will then be as follows: the link may undergo decoherence followed by purification a number of times, until the link is removed. The removal is due to either consumption or purification failure. After the link is removed, entanglement generation will be attempted until success, at which point a link is transferred to the G memory with shifted fidelity H_{new}. See Figure 7.10 for an illustration of this.

Definition 7.7. *We define $T_0 = 0$, $\{T_i\}_{i=1}^{\infty}$ to be the times at which H (equivalently, F) experiences a change that is due to purification, consumption or entanglement generation (or alternatively, any change that is not due to decoherence). Let $S_i := T_i - T_{i-1}$ denote the times between each jump.*

We also refer to the $\{T_i\}$ as the *jump times*. See Figure 7.10 for a depiction.

Now, recall that both the time until entanglement generation and consumption are assumed to be geometrically distributed. Then, the distribution of S_i is then given by

$$S_i = \begin{cases} \min\left\{\tau_{\text{pur}}^{(i)}, \tau_{\text{con}}^{(i)}\right\} & \text{if } H(T_{i-1}) \geq 0 \\ T_{\text{gen}}^{(i)} & \text{if } H(T_{i-1}) < 0, \end{cases} \tag{7.72}$$

where $T_{\text{gen}}^{(i)}$, $\tau_{\text{pur}}^{(i)}$, and $\tau_{\text{con}}^{(i)}$ are independent random variables with the following distributions

$$\begin{aligned} T_{\text{gen}}^{(i)} &\sim \text{Geo}\left(1-(1-p_{\text{gen}})^n\right) \\ \tau_{\text{pur}}^{(i)} &\sim \text{Geo}\left(q(1-(1-p_{\text{gen}})^n)\right) \\ \tau_{\text{con}}^{(i)} &\sim \text{Geo}(p_{\text{con}}). \end{aligned} \tag{7.73}$$

Here, starting at jump time T_{i-1}, $T_{\text{gen}}^{(i)}$ is the time until a new link is generated and transferred to memory, $\tau_{\text{pur}}^{(i)}$ is the time until there is a successful generation and the system decides to attempt purification, and $\tau_{\text{con}}^{(i)}$ is the time until there is a consumption request.

Definition 7.8. *For $i \geq 0$, we define $H_i := H(T_i)$ to be the shifted fidelity at the jump times of the process. See Figure 7.10 for an illustration.*

Since we assume that the system starts with a freshly generated link in memory, we have $H_0 = H_{\text{new}}$. We note that $\{H_i\}_{i\geq 0}$ is a Markov chain.

Definition 7.9. *Let T_{occ} be the first time at which the link in the G memory is removed from the system. In particular, $T_{\text{occ}} = T_N$, where*

$$N = \min\left\{i : H_i < 0\right\}. \tag{7.74}$$

Note that N is finite a.s. since it is upper bounded by the time until the first consumption request arrives, which follows a geometric distribution.

In Appendix 7.6, we saw that $F(t)$ is a regenerative process, meaning that it can be broken down into i.i.d. cycles $V_k = T_{\text{occ}}^{(k)} + T_{\text{gen}}^{(k)}$, where $T_{\text{occ}}^{(k)}$ are the times during which the G memory is occupied and $T_{\text{gen}}^{(k)}$ are the times during which the G memory is empty. We note that in Definition 7.9, $T_{\text{occ}} = T_{\text{occ}}^{(1)}$. From now on, we also refer to $T_{\text{gen}} \equiv T_{\text{gen}}^{(1)}$.

It follows straightforwardly that $H(t) = F(t) - 1/4$ is a regenerative process with the same cycles as $F(t)$. We saw in Lemma 7.1 that the performance metrics may be rewritten in terms of the statistical properties of one cycle. This result also holds for $\overline{H}$, which we restate below. Recalling the notation introduced in Appendix 7.6 for F^-, we will also use the equivalent notation for H^-, i.e.,

$$H^-(t) = F^-(t) - \frac{1}{4}. \tag{7.75}$$

Lemma 7.4 (Performance metrics for H, single cycle)**.** *The availability is given by*

$$A = \frac{\mathbb{E}[T_{\text{occ}}]}{\mathbb{E}[T_{\text{occ}}] + \mathbb{E}[T_{\text{gen}}]} \text{ a.s.} \tag{7.76}$$

and the average consumed (shifted) fidelity is given by

$$\overline{H} = \mathbb{E}\left[e^{-\Gamma S_N} H_{N-1} | \tau_{\text{con}}^{(N)} \leq \tau_{\text{pur}}^{(N)} \right] \text{ a.s.} \tag{7.77}$$

where $C_1 \equiv \{\tau_{\text{con}}^{(N)} \leq \tau_{\text{pur}}^{(N)}\}$ *is the event that the link is consumed at time* T_{occ}.

Proof. The identity (7.76) follows directly from Lemma 7.1. In the same Lemma, we saw that

$$\overline{F} = \mathbb{E}[F(T_{\text{occ}}^{(1)-}) | C_1], \tag{7.78}$$

where C_1 is the event that the first link is removed due to consumption, and we recall the notation

$$F^-(t) = e^{-\Gamma}\left(F(t-1) - \frac{1}{4}\right) + \frac{1}{4},$$

which is necessary to capture the fidelity when *consumed* at time t, since the discrete-time stochastic process is defined such that $H(T_{\text{occ}}^{(1)}) = 0$. The value of $H^-(T_{\text{occ}}^{(1)})$ is given by $e^{-\Gamma S_N} H_{N-1}$, where H_{N-1} is the value of the shifted fidelity at the previous jump time (see Definition 7.7) and S_N is the time the link spends decohering in memory from that point until the link is removed from memory (see Definition 7.9). For the conditioning, we recall from (7.72) that $C_1 \equiv \{\tau_{\text{con}}^{(N)} \leq \tau_{\text{pur}}^{(N)}\}$. □

By properties of geometric random variables, we already know that

$$\mathbb{E}[T_{\text{gen}}^{(1)}] = \frac{1}{1-(1-p_{\text{gen}})^n}.$$

To solve for our two performance metrics, it is then sufficient to find formulae for $\mathbb{E}[T_{\text{occ}}]$ and $\mathbb{E}\left[e^{-\Gamma S_N} H_{N-1} | \tau_{\text{con}}^{(N)} \leq \tau_{\text{pur}}^{(N)} \right]$. This is what we accomplish with the following results.

Definition 7.10. *For* $i \leq N$, *let* U_i *denote the event that purification is attempted at the* i*th jump time, and* $R_i \subseteq U_i$ *denote the event that purification is attempted* and *succeeds at the* i*th jump time.*

Lemma 7.5. *Let* x *and* y *be given by*

$$x := \sum_{i=1}^{\infty} \mathbb{E}\left[H_i \prod_{j=1}^{i} \mathbb{1}_{R_j} \right], \quad y := \sum_{i=1}^{\infty} \mathrm{P}(N > i). \tag{7.79}$$

Then,

$$\mathbb{E}[T_{\text{occ}}] = \frac{1+y}{p_{\text{con}} + q\left(1-\left(1-p_{\text{gen}}\right)^n\right)(1-p_{\text{con}})} \tag{7.80}$$

and

$$\mathbb{E}[e^{-\Gamma S_N} H_{N-1} | \tau_{\text{con}}^{(N)} \leq \tau_{\text{pur}}^{(N)}] = \frac{(H_{\text{new}} + x)(p_{\text{con}} + q\left(1-\left(1-p_{\text{gen}}\right)^n\right)(1-p_{\text{con}}))}{(1+y)\left(e^{\Gamma} - 1 + p_{\text{con}} + q\left(1-\left(1-p_{\text{gen}}\right)^n\right)(1-p_{\text{con}})\right)}. \tag{7.81}$$

7

Proof. Denoting $U_N^{\mathrm{c}} = \left\{\tau_{\mathrm{con}}^{(N)} \leq \tau_{\mathrm{pur}}^{(N)}\right\}$ and using properties of the conditional expectation, we may write

$$\mathbb{E}[e^{-\Gamma S_N} H_{N-1} | \tau_{\mathrm{con}}^{(N)} \leq \tau_{\mathrm{pur}}^{(N)}] = \frac{\mathbb{E}[e^{-\Gamma S_N} H_{N-1} \mathbb{1}_{U_N^{\mathrm{c}}}]}{\mathrm{P}(U_N^{\mathrm{c}})}. \tag{7.82}$$

The denominator $\mathrm{P}(U_N^{\mathrm{c}})$ may be rewritten as

$$\begin{aligned}
\mathrm{P}(U_N^{\mathrm{c}}) &= \mathbb{E}[\mathbb{1}_{U_N^{\mathrm{c}}}] \\
&\overset{\mathrm{i}}{=} \mathbb{E}\left[\mathbb{1}_{U_1^{\mathrm{c}}} + \sum_{i=2}^{\infty} \mathbb{1}_{U_i^{\mathrm{c}}} \prod_{j=1}^{i-1} \mathbb{1}_{R_j}\right] \\
&\overset{\mathrm{ii}}{=} \mathbb{E}\left[\mathbb{1}_{U_1^{\mathrm{c}}}\right] + \sum_{i=2}^{\infty} \mathbb{E}\left[\mathbb{1}_{U_i^{\mathrm{c}}} | R_1, \ldots, R_{i-1}\right] \mathbb{E}\left[\prod_{j=1}^{i-1} \mathbb{1}_{R_j}\right] \\
&\overset{\mathrm{iii}}{=} \mathbb{E}\left[\mathbb{1}_{U_1^{\mathrm{c}}}\right] \left(1 + \sum_{i=1}^{\infty} \mathbb{E}\left[\prod_{j=1}^{i} \mathbb{1}_{R_j}\right]\right) \\
&\overset{\mathrm{iv}}{=} \mathrm{P}\left(U_1^{\mathrm{c}}\right) \left(1 + \sum_{i=1}^{\infty} \mathrm{P}\left(N > i\right)\right) \\
&= \mathrm{P}\left(U_1^{\mathrm{c}}\right) \left(1 + y\right).
\end{aligned} \tag{7.83}$$

In the above, we have used the following steps:

i. One may partition the event U_N^{c} by conditioning on the value of N as

$$U_N^{\mathrm{c}} = \bigcup_{i=1}^{\infty} \left(U_i^{\mathrm{c}} \cap \{N = i\}\right).$$

Now, notice that we have $U_i^{\mathrm{c}} \cap \{N = i\}$ exactly when successful purification occurs $i-1$ times, and the link is consumed. Therefore,

$$U_i^{\mathrm{c}} \cap \{N = i\} = U_i^{\mathrm{c}} \cap \left(\cap_{j=1}^{i-1} R_j\right). \tag{7.84}$$

Since the $U_i^{\mathrm{c}} \cap \{N = i\}$ are mutually exclusive, it follows from the above that

$$\begin{aligned}
\mathbb{1}_{U_N^{\mathrm{c}}} &= \sum_{i=1}^{\infty} \mathbb{1}_{U_i^{\mathrm{c}} \cap \{N=i\}} \\
&= \sum_{i=1}^{\infty} \mathbb{1}_{U_i^{\mathrm{c}} \cap \left(\cap_{j=1}^{i-1} R_j\right)} \\
&= \mathbb{1}_{U_1^{\mathrm{c}}} + \sum_{i=2}^{\infty} \mathbb{1}_{U_i^{\mathrm{c}}} \prod_{j=1}^{i-1} \mathbb{1}_{R_j}.
\end{aligned} \tag{7.85}$$

ii. We use linearity of taking the expectation and take the expectation inside the sum, which is possible by the monotone convergence theorem (see 5.6.12 of [76]). Then, we express the joint probability in terms of conditional probabilities.

iii. We have used the fact that

$$\mathbb{E}\left[\mathbb{1}_{U_i^c}|R_1,...,R_{i-1}\right] = \mathrm{P}(\tau_{\mathrm{con}}^{(i)} \leq \tau_{\mathrm{pur}}^{(i)}) = \mathrm{P}(\tau_{\mathrm{con}}^{(1)} \leq \tau_{\mathrm{pur}}^{(1)}) = \mathbb{E}\left[\mathbb{1}_{U_1^c}\right].$$

iv. Notice that $N > i$ if and only if the first i jump times are due to successful purification. Therefore,

$$\{N > i\} \equiv \cap_{j=1}^{i} R_j.$$

We therefore have

$$\mathbb{E}\left[\prod_{j=1}^{i} \mathbb{1}_{R_j}\right] = \mathbb{E}\left[\mathbb{1}_{\{N>i\}}\right] = P(N > i).$$

Secondly, we rewrite the numerator of (7.82) in a similar way:

$$\begin{aligned}
\mathbb{E}[e^{-\Gamma S_N} H_{N-1} \mathbb{1}_{U_N^c}] &\overset{\text{i}}{=} \mathbb{E}\left[e^{-\Gamma S_N} H_{N-1} \cdot \left(\mathbb{1}_{U_1^c} + \sum_{i=2}^{\infty} \mathbb{1}_{U_i^c} \prod_{j=1}^{i-1} \mathbb{1}_{R_j}\right)\right] \\
&\overset{\text{ii}}{=} \mathbb{E}\left[H_{\mathrm{new}} e^{-\Gamma S_1} \mathbb{1}_{U_1^c} + \sum_{i=1}^{\infty} e^{-\Gamma S_{i+1}} H_i \mathbb{1}_{U_{i+1}^c} \prod_{j=1}^{i} \mathbb{1}_{R_j}\right] \\
&\overset{\text{iii}}{=} H_{\mathrm{new}} \mathbb{E}\left[e^{-\Gamma S_1} \mathbb{1}_{U_1^c}\right] + \sum_{i=1}^{\infty} \mathbb{E}\left[e^{-\Gamma S_{i+1}} \mathbb{1}_{U_{i+1}^c}|R_1,...,R_i\right] \mathbb{E}\left[H_i \prod_{j=1}^{i} \mathbb{1}_{R_i}\right] \\
&\overset{\text{iv}}{=} \mathbb{E}\left[e^{-\Gamma S_1} \mathbb{1}_{U_1^c}\right] \left(H_{\mathrm{new}} + \sum_{i=1}^{\infty} \mathbb{E}\left[H_i \prod_{j=1}^{i} \mathbb{1}_{R_j}\right]\right) \\
&= \mathbb{E}\left[e^{-\Gamma S_1} \mathbb{1}_{U_1^c}\right] (H_{\mathrm{new}} + x). \qquad (7.86)
\end{aligned}$$

In the above, we have used the following steps:

i. We have again made use of (7.85), and $H_0 := H_{\mathrm{new}}$.

ii. Again making use of (7.84), we have noticed that the indicator function selects the value of N as

$$\begin{aligned}
e^{-\Gamma S_N} H_{N-1} \cdot \mathbb{1}_{U_i^c} \prod_{j=1}^{i-1} \mathbb{1}_{R_j} &= e^{-\Gamma S_N} H_{N-1} \cdot \mathbb{1}_{U_i^c \cap \{N=i\}} \\
&= e^{-\Gamma S_i} H_{i-1} \cdot \mathbb{1}_{U_i^c} \prod_{j=1}^{i-1} \mathbb{1}_{R_j}.
\end{aligned}$$

iii. We have used linearity of the expectation, and take the expectation inside the sum, which is possible by the monotone convergence theorem (see 5.6.12 of [76]). Then, we express the joint probability in terms of conditional probabilities.

iv. We have again used the fact that, conditioned on $R_1,...,R_{i-1}$, $e^{-\Gamma S_i} \mathbb{1}_{U_i^c}$ are identically distributed, for all $i \geq 1$.

We now directly evaluate the multiplying factor in the above expressions. Using the partition $U_1^c = \bigcup_{i=1}^{\infty} \{U_1^c, S_1 = i\}$,

$$\begin{aligned}\mathbb{E}\left[e^{-\Gamma S_1}\mathbb{1}_{U_1^c}\right] &= \mathbb{E}\left[e^{-\Gamma S_1}\mathbb{1}_{U_1^c}\middle|U_1\right]\mathrm{P}(U_1) + \sum_{i=1}^{\infty}\mathbb{E}\left[e^{-\Gamma S_1}\mathbb{1}_{U_1^c}\middle|U_1^c, S_1 = i\right]\mathrm{P}(U_1^c, S_1 = i)\\ &= 0 + \sum_{i=1}^{\infty} e^{-i\Gamma}\mathrm{P}(U_1^c, S_1 = i). \end{aligned} \tag{7.87}$$

Recalling that $U_1^c = \{\tau_{\mathrm{con}}^{(1)} \leq \tau_{\mathrm{pur}}^{(1)}\}$, we now evaluate

$$\begin{aligned}\mathrm{P}(U_1^c, S_1 = i) &= \mathrm{P}(i = \tau_{\mathrm{con}}^{(1)}, i \leq \tau_{\mathrm{pur}}^{(1)})\\ &= \mathrm{P}(i = \tau_{\mathrm{con}}^{(1)}) \cdot \mathrm{P}(i \leq \tau_{\mathrm{pur}}^{(1)})\\ &= (1-p_{\mathrm{con}})^{i-1} p_{\mathrm{con}} \cdot \left(1 - q(1-(1-p_{\mathrm{gen}})^n)\right)^{i-1}, \end{aligned} \tag{7.88}$$

where we have used the fact that $\tau_{\mathrm{con}}^{(1)}$ and $\tau_{\mathrm{pur}}^{(1)}$ are independent, and have distributions as given in (7.73). Therefore, combining (7.87) and (7.88), it follows that

$$\begin{aligned}\mathbb{E}\left[e^{-\Gamma S_1}\mathbb{1}_{U_1^c}\right] &= \sum_{i=1}^{\infty} e^{-\Gamma i}(1-p_{\mathrm{con}})^{i-1} p_{\mathrm{con}} \cdot \left(1 - q\left(1-\left(1-p_{\mathrm{gen}}\right)^n\right)\right)^{i-1}\\ &= p_{\mathrm{con}} e^{-\Gamma} \sum_{i=0}^{\infty} e^{-\Gamma i}\left(1 - q\left(1-\left(1-p_{\mathrm{gen}}\right)^n\right)\right)^{i}(1-p_{\mathrm{con}})^{i}\\ &= \frac{p_{\mathrm{con}} e^{-\Gamma}}{1 - e^{-\Gamma}\left(1 - q\left(1-\left(1-p_{\mathrm{gen}}\right)^n\right)\right)(1-p_{\mathrm{con}})}, \end{aligned} \tag{7.89}$$

where to obtain the first equality we have relabelled the summing index, and to obtain the second equality we have used the formula for a geometric series. By setting $\Gamma = 0$ in the above, we also obtain

$$\begin{aligned}\mathbb{E}\left[\mathbb{1}_{U_1^c}\right] = \mathrm{P}(U_1^c) &= \frac{p_{\mathrm{con}}}{1 - \left(1 - q\left(1-\left(1-p_{\mathrm{gen}}\right)^n\right)\right)(1-p_{\mathrm{con}})}\\ &= \frac{p_{\mathrm{con}}}{p_{\mathrm{con}} + q\left(1-\left(1-p_{\mathrm{gen}}\right)^n\right)(1-p_{\mathrm{con}})}. \end{aligned} \tag{7.90}$$

Then, combining (7.83), (7.86) (7.89), (7.90) allows us to rewrite (7.82) as

$$\mathbb{E}[e^{-\Gamma S_N} H_{N-1} | \tau_{\mathrm{con}}^{(N)} \leq \tau_{\mathrm{pur}}^{(N)}] = \frac{(H_{\mathrm{new}} + x)(p_{\mathrm{con}} + q\left(1-\left(1-p_{\mathrm{gen}}\right)^n\right)(1-p_{\mathrm{con}}))}{(1+y)\left(e^{\Gamma} - \left(1 - q\left(1-\left(1-p_{\mathrm{gen}}\right)^n\right)\right)(1-p_{\mathrm{con}})\right)}.$$

This shows (7.81). We now show (7.80) using a similar method: firstly, we again condition on the value of N. Recalling Definitions 7.7 and 7.9, we may rewrite T_N as

$$T_N = \sum_{i=1}^{\infty} S_i \cdot \mathbb{1}_{\{N \geq i\}} = S_1 + \sum_{i=2}^{\infty} S_i \prod_{j=1}^{i-1} \mathbb{1}_{R_j},$$

where we have again used $\{N \geq i\} \equiv \cap_{j=1}^{i-1} R_j$ to obtain the second equality. Taking expectations, it follows that

$$\begin{aligned}
\mathbb{E}[T_N] &= \mathbb{E}\left[S_1 + \sum_{i=2}^{\infty} S_i \prod_{j=1}^{i-1} \mathbb{1}_{R_j}\right] \\
&= \mathbb{E}[S_1] + \sum_{i=2}^{\infty} \mathbb{E}[S_i | R_1, ..., R_{i-1}] \mathbb{E}\left[\prod_{j=1}^{i-1} \mathbb{1}_{R_j}\right] \\
&= \mathbb{E}[S_1]\left(1 + \sum_{i=1}^{\infty} \mathbb{E}\left[\prod_{j=1}^{i} \mathbb{1}_{R_j}\right]\right) \\
&= \mathbb{E}[S_1]\big(1 + y\big), \qquad (7.91)
\end{aligned}$$

where we have used the same reasoning as was used to obtain (7.83) and (7.86). It now only remains to compute $\mathbb{E}[S_1]$. Recalling that $S_1 = \min\{\tau_{\text{con}}^{(1)}, \tau_{\text{pur}}^{(1)}\}$, we see that

$$\begin{aligned}
\mathrm{P}(S_1 > i) &= \mathrm{P}(\tau_{\text{con}}^{(1)} > i, \tau_{\text{pur}}^{(1)} > i) \\
&= \mathrm{P}(\tau_{\text{con}}^{(1)} > i)(\tau_{\text{pur}}^{(1)} > i) \\
&= (1 - p_{\text{con}})^i \cdot (1 - q(1 - (1 - p_{\text{gen}})^n))^i,
\end{aligned}$$

where we have used the fact that $\tau_{\text{con}}^{(1)}$ and $\tau_{\text{pur}}^{(1)}$ are independent random variables, and their distributions which are given in (7.73). Then, we may rewrite the expectation as

$$\begin{aligned}
\mathbb{E}[S_1] = \sum_{i=0}^{\infty} \mathrm{P}(S_1 > i) &= \sum_{i=0}^{\infty} (1 - p_{\text{con}})^i \cdot (1 - q(1 - (1 - p_{\text{gen}})^n))^i \\
&= \frac{1}{1 - (1 - p_{\text{con}})(1 - q(1 - (1 - p_{\text{gen}})^n))} \\
&= \frac{1}{p_{\text{con}} + q(1 - p_{\text{con}})(1 - (1 - p_{\text{gen}})^n)},
\end{aligned}$$

where we have used the formula for a geometric series to evaluate the sum. Rearranging terms and combining the above with (7.91), we may then write this as

$$\mathbb{E}[T_N] = \frac{1 + y}{p_{\text{con}} + q(1 - p_{\text{con}})(1 - (1 - p_{\text{gen}})^n)},$$

which shows (7.80). □

Lemma 7.6. *Let x and y be defined as in (7.79). Then,*

$$x = -H_{\text{new}} + \frac{\tilde{B} - \tilde{D}H_{\text{new}} + H_{\text{new}}}{(1 - \tilde{A})(1 - \tilde{D}) - \tilde{B}\tilde{C}}, \quad y = -1 + \frac{1 - \tilde{A} + \tilde{C}H_{\text{new}}}{(1 - \tilde{A})(1 - \tilde{D}) - \tilde{B}\tilde{C}}, \qquad (7.92)$$

where $\tilde{A}$, $\tilde{B}$, $\tilde{C}$, $\tilde{D}$ are defined in Theorem 7.1 in the main text.

Proof. We firstly define the quantities

$$x_i \coloneqq \mathbb{E}\left[H_i \prod_{j=1}^{i} \mathbb{1}_{R_j}\right], \quad y_i \coloneqq \mathrm{P}(N > i), \tag{7.93}$$

which means that, recalling (7.79), x and y may be rewritten as

$$x = \sum_{i=1}^{\infty} x_i, \quad y = \sum_{i=1}^{\infty} y_i. \tag{7.94}$$

We now show that there is a recursive relationship between the $\{x_i\}$ and the $\{y_i\}$. We firstly rewrite x_i by conditioning on the value of H_{i-1}. In particular, recalling that $\cap_{j=1}^{i} R_j = \{N > i\}$, we have

$$x_i = \mathbb{E}\left[H_i \prod_{j=1}^{i} \mathbb{1}_{R_j}\right] = \mathbb{E}\left[H_i \mathbb{1}_{\cap_{j=1}^{i} R_j}\right] = \mathbb{E}\left[H_i \mathbb{1}_{\{N>i\}}\right].$$

Then, one may partition by conditioning on the value of H_{i-1} in the following way:

$$\{N > i-1\} = \bigcup_h \{H_{i-1} = h, N > i-1\}.$$

We may then rewrite x_i as

$$\begin{aligned} x_i &= \mathbb{E}\left[H_i \mathbb{1}_{\{N>i\}} \middle| N \leq i-1\right] \mathrm{P}(N \leq i-1) \\ &\quad + \sum_h \mathbb{E}\left[H_i \mathbb{1}_{\{N>i\}} \middle| H_{i-1} = h, N > i-1\right] \cdot \mathrm{P}(H_{i-1} = h, N > i-1) \\ &= 0 + \sum_h \mathbb{E}\left[H_i \mathbb{1}_{\{N>i\}} \middle| H_{i-1} = h, N > i-1\right] \cdot \mathrm{P}(H_{i-1} = h, N > i-1). \end{aligned} \tag{7.95}$$

We now focus on evaluating $\mathbb{E}\left[H_i \mathbb{1}_{\{N>i\}} \middle| H_{i-1} = h, N > i-1\right]$. We do this for $h > 0$, as this is the only relevant range in the above formula. We firstly notice that this expression may be rewritten as

$$\begin{aligned} \mathbb{E}\left[H_i \mathbb{1}_{\{N>i\}} \middle| H_{i-1} = h, N > i-1\right] &= \mathbb{E}\left[H_i \prod_{j=1}^{i} \mathbb{1}_{R_j} \middle| H_{i-1} = h, \cap_{j=1}^{i-1} R_j\right] \\ &= \mathbb{E}\left[H_i \mathbb{1}_{R_i} \middle| H_{i-1} = h, \cap_{j=1}^{i-1} R_j\right] \\ &= \mathbb{E}\left[H_i \mathbb{1}_{R_i} \middle| H_{i-1} = h\right], \end{aligned} \tag{7.96}$$

where to obtain the final equality we have used the Markovian property of the system: given the information that $H_{i-1} = h > 0$, this is sufficient to understand the future behavior $\{H_k\}_{k \geq i}$. This follows from the fact that $\{H_i\}$ is a Markov chain.

Recall that R_i is the event where the ith jump time is due to a purification round succeeding. Given that in the above expression we are conditioning on the value H_{i-1}, the random variables on which H_i depends are therefore the time S_i until the next round of purification, and the number of links L_i that are used for this purification (recalling that this number determines which purification protocol is used). We must therefore take the expectation over these two random variables.

Definition 7.11. *For $i < N$ (the i-th successful purification round), let L_i be the number of links that were produced in the bad memories just before time T_i.*

We then expand the expectation (7.96) to condition on the values taken by S_i and L_i:

$$\begin{aligned}\mathbb{E}\left[H_i \mathbb{1}_{R_i} \middle| H_{i-1} = h\right] &= \sum_{t,k} \mathbb{E}\left[H_i \mathbb{1}_{R_i} \middle| H_{i-1} = h, S_i = t, L_i = k, R_i\right] \cdot \mathrm{P}(S_i = t, L_i = k, R_i | H_{i-1} = h) \\ &= \sum_{t,k} \tilde{J}_k\left(e^{-\Gamma t} h\right) \cdot \mathrm{P}(S_i = t, L_i = k, R_i | H_{i-1} = h), \end{aligned} \tag{7.97}$$

where, recalling (7.66), $\tilde{J}_k$ is the jump function corresponding to the $(k+1)$-to-1 purification protocol from our purification policy. To evaluate (7.97), it now remains to compute the probability distribution in the weighted sum. We again condition, to find

$$\begin{aligned}\mathrm{P}(S_i = t, L_i = k, R_i | H_{i-1} = h) =& \mathrm{P}(R_i | U_i, S_i = t, L_i = k, H_{i-1} = h) \\ & \cdot \mathrm{P}(U_i, S_i = t, L_i = k | H_{i-1} = h) \\ =& \tilde{p}_k(e^{-\Gamma t} h) \cdot \mathrm{P}(U_i, S_i = t, L_i = k | H_{i-1} = h), \end{aligned} \tag{7.98}$$

where $\tilde{p}_k$ determines the probability of successful purification when employing the $(k+1)$-to-1 protocol, recalling its definition in (7.67). Now, recalling the distribution of S_i from (7.72),

$$\begin{aligned}\mathrm{P}\left(U_i,\, S_i = t,\, L_i = k | H_{i-1} = h\right) =& \mathrm{P}(\tau_{\mathrm{con}}^{(i)} > t,\, \tau_{\mathrm{pur}}^{(i)} = t,\, L_i = k) \\ =& \mathrm{P}(\tau_{\mathrm{con}}^{(i)} > t) \cdot \mathrm{P}(\tau_{\mathrm{pur}}^{(i)} = t,\, L_i = k) \\ =& (1 - p_{\mathrm{con}})^t \cdot (1 - q(1 - (1 - p_{\mathrm{gen}})^n))^{t-1} \\ & \cdot q \binom{n}{k} p_{\mathrm{gen}}^k (1 - p_{\mathrm{gen}})^{n-k}, \end{aligned} \tag{7.99}$$

where we have used the fact that $\tau_{\mathrm{pur}}^{(i)}$ and L_i are independent of $\tau_{\mathrm{con}}^{(i)}$.

Combining (7.97), (7.98) and (7.99) yields

$$\begin{aligned}\mathbb{E}\left[H_i \mathbb{1}_{R_i} \middle| H_{i-1} = h\right] &= \sum_{t,k} \binom{n}{k} \tilde{J}_k\left(e^{-\Gamma t} h\right) \tilde{p}_k\left(e^{-\Gamma t} h\right) \cdot (1 - p_{\mathrm{con}})^t \\ &\qquad \cdot (1 - q(1 - (1 - p_{\mathrm{gen}})^n))^{t-1} \cdot q \cdot p_{\mathrm{gen}}^k (1 - p_{\mathrm{gen}})^{n-k} \\ &= \sum_{t,k} \binom{n}{k} \left(a_k e^{-\Gamma t} h + b_k\right) \cdot (1 - p_{\mathrm{con}})^t \cdot (1 - q(1 - (1 - p_{\mathrm{gen}})^n))^{t-1} \\ &\qquad \cdot q \cdot p_{\mathrm{gen}}^k (1 - p_{\mathrm{gen}})^{n-k} \\ &= \sum_t \left(\tilde{a} e^{-\Gamma t} h + \tilde{b}\right) \cdot (1 - p_{\mathrm{con}})^t \cdot (1 - q(1 - (1 - p_{\mathrm{gen}})^n))^{t-1} \cdot q, \end{aligned} \tag{7.100}$$

where in the first step, we have made use of the expressions (7.66) and (7.67) that define the purification jump function and success probability for the shifted fidelity. In the second step, we have defined

$$\tilde{a} = \sum_{k=1}^{n} a_k \cdot \binom{n}{k} (1 - p_{\mathrm{gen}})^{n-k} p_{\mathrm{gen}}^k, \quad \tilde{b} = \sum_{k=1}^{n} b_k \cdot \binom{n}{k} (1 - p_{\mathrm{gen}})^{n-k} p_{\mathrm{gen}}^k.$$

Now, using the fact that (7.100) is a geometric series (starting from $t = 1$), we obtain

$$\mathbb{E}\left[H_i \mathbb{1}_{R_i} | H_{i-1} = h\right] = \tilde{A}h + \tilde{B}, \tag{7.101}$$

where

$$\tilde{A} = \frac{q(1-p_{\text{con}})\tilde{a}}{e^{\Gamma} - \left(1 - q + q\left(1-p_{\text{gen}}\right)^n\right)(1-p_{\text{con}})}, \quad \tilde{B} = \frac{q(1-p_{\text{con}})\tilde{b}}{p_{\text{con}} + q\left(1-\left(1-p_{\text{gen}}\right)^n\right)(1-p_{\text{con}})}. \tag{7.102}$$

Combining (7.95) and (7.101), we may then write

$$\begin{aligned} x_i &= \sum_h (\tilde{A}h + \tilde{B}) \cdot \mathrm{P}(H_{i-1} = h,\, N > i-1) \\ &= \tilde{A} \cdot \mathbb{E}[H_{i-1} \mathbb{1}_{N>i-1}] + \tilde{B} \cdot \mathrm{P}(N > i-1) \\ &= \tilde{A} x_{i-1} + \tilde{B} y_{i-1}, \end{aligned}$$

which is our first recursion relation for $\{x_i\}$ and $\{y_i\}$. We now write down an analogous recursion relation for y_i. We use the same method as for the x_i. In particular, using

$$y_i = \mathrm{P}(N > i) = \mathbb{E}\left[\mathbb{1}_{N>i}\right],$$

we again expand the expectation while conditioning on the value of H_{i-1} in a step analogous to (7.95):

$$y_i = \sum_{h>0} \mathbb{E}\left[\mathbb{1}_{\{N>i\}} | H_{i-1} = h,\, N > i-1\right] \cdot \mathrm{P}(H_{i-1} = h,\, N > i-1). \tag{7.103}$$

In a step analogous to (7.96), we rewrite the conditional expectation as

$$\mathbb{E}\left[\mathbb{1}_{\{N>i\}} | H_{i-1} = h,\, N > i-1\right] = \mathbb{E}\left[\mathbb{1}_{R_i} | H_{i-1} = h\right]. \tag{7.104}$$

We then expand the above with the distributions of S_i and L_i to obtain

$$\begin{aligned} \mathbb{E}\left[\mathbb{1}_{R_i} | H_{i-1} = h\right] &= \sum_{t,k} 1 \cdot \mathrm{P}(S_i = t, L_i = k, R_i | H_{i-1} = h) \\ &= \sum_{t,k} 1 \cdot \tilde{p}_k(e^{-\Gamma t} h) \mathrm{P}(U_i, S_i = t, L_i = k | H_{i-1} = h) \\ &= \sum_{t,k} 1 \cdot \binom{n}{k} \cdot \tilde{p}_k(e^{-\Gamma t} h) \cdot (1-p_{\text{con}})^t \cdot (1 - q(1-(1-p_{\text{gen}})^n))^{t-1} \\ &\qquad \cdot q \cdot p_{\text{gen}}^k (1-p_{\text{gen}})^{n-k}, \end{aligned}$$

where in the second step we have used (7.98) and in the last step we have again used the conditional distribution in (7.99). Using again the definition for $\tilde{p}_k$ as given in (7.67), one may simplify the above to obtain

$$\mathbb{E}\left[\mathbb{1}_{R_i} | H_{i-1} = h\right] = \sum_{t>0} (\tilde{c} e^{-\Gamma t} h + \tilde{d}) \cdot (1-p_{\text{con}})^t \cdot (1 - q(1-(1-p_{\text{gen}})^n))^{t-1} \cdot q, \tag{7.105}$$

where we have defined

$$\tilde{c} = \sum_{k=1}^{n} c_k \cdot \binom{n}{k} (1-p_{\text{gen}})^{n-k} p_{\text{gen}}^k, \quad \tilde{d} = \sum_{k=1}^{n} d_k \cdot \binom{n}{k} (1-p_{\text{gen}})^{n-k} p_{\text{gen}}^k.$$

One may again use a geometric series to evaluate (7.105), to obtain

$$\mathbb{E}\left[\mathbb{1}_{R_i} | H_{i-1} = h\right] = \tilde{C}h + \tilde{D}, \tag{7.106}$$

where

$$\tilde{C} = \frac{q(1-p_{\text{con}})\tilde{c}}{e^{\Gamma} - \left(1-q+q\left(1-p_{\text{gen}}\right)^n\right)(1-p_{\text{con}})}, \quad \tilde{D} = \frac{q(1-p_{\text{con}})\tilde{d}}{p_{\text{con}} + q\left(1-\left(1-p_{\text{gen}}\right)^n\right)(1-p_{\text{con}})}. \tag{7.107}$$

Combining (7.103), (7.104) and (7.106) then yields

$$\begin{aligned} y_i &= \sum_{h>0} (\tilde{C}h + \tilde{D}) \cdot \mathrm{P}(H_{i-1} = h,\, N > i-1) \\ &= \tilde{C} \cdot \mathbb{E}\left[H_{i-1} \mathbb{1}_{\{N>i-1\}}\right] + \tilde{D} \cdot P(N > i-1) \\ &= \tilde{C} x_{i-1} + \tilde{D} y_{i-1}, \end{aligned}$$

which completes our second recursion relation for the $\{x_i\}$ and $\{y_i\}$. We now combine these to find expressions for x and y. Given the initial values

$$x_0 = \mathbb{E}\left[H_0 \mathbb{1}_{N>0}\right] = \mathbb{E}\left[H_{\text{new}} \cdot 1\right] = H_{\text{new}}$$

and

$$y_0 = \mathrm{P}(N > 0) = 1,$$

it follows that

$$\begin{aligned} x &= \sum_{i=1}^{\infty} (\tilde{A}x_{i-1} + \tilde{B}y_{i-1}) = \tilde{A}(x + H_{\text{new}}) + \tilde{B}(y+1) \\ y &= \sum_{i=1}^{\infty} (\tilde{C}x_{i-1} + \tilde{D}y_{i-1}) = \tilde{C}(x + H_{\text{new}}) + \tilde{D}(y+1). \end{aligned}$$

We therefore have a linear system of equations for x and y, which may be written as

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \tilde{A} & \tilde{B} \\ \tilde{C} & \tilde{D} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \tilde{A}H_{\text{new}} + \tilde{B} \\ \tilde{C}H_{\text{new}} + \tilde{D} \end{pmatrix}, \tag{7.108}$$

which has solution

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{(1-\tilde{A})(1-\tilde{D}) - \tilde{B}\tilde{C}} \begin{pmatrix} 1-\tilde{D} & \tilde{B} \\ \tilde{C} & 1-\tilde{A} \end{pmatrix} \begin{pmatrix} \tilde{A}H_{\text{new}} + \tilde{B} \\ \tilde{C}H_{\text{new}} + \tilde{D} \end{pmatrix}, \tag{7.109}$$

providing us with the formulae for x and y. These may be simplified in the following way:

$$\begin{aligned} x &= \frac{(1-\tilde{D})(\tilde{A}H_{\text{new}} + \tilde{B}) + \tilde{B}(\tilde{C}H_{\text{new}} + \tilde{D})}{(1-\tilde{A})(1-\tilde{D}) - \tilde{B}\tilde{C}} = -H_{\text{new}} + \frac{\tilde{B} - \tilde{D}H_{\text{new}} + H_{\text{new}}}{(1-\tilde{A})(1-\tilde{D}) - \tilde{B}\tilde{C}} \\ y &= \frac{\tilde{C}(\tilde{A}H_{\text{new}} + \tilde{B}) + (1-\tilde{A})(\tilde{C}H_{\text{new}} + \tilde{D})}{(1-\tilde{A})(1-\tilde{D}) - \tilde{B}\tilde{C}} = -1 + \frac{1 - \tilde{A} + \tilde{C}H_{\text{new}}}{(1-\tilde{A})(1-\tilde{D}) - \tilde{B}\tilde{C}}, \end{aligned}$$

which are in the final form for x and y, as given in (7.92). □

Proof of Theorems 7.1 and 7.2. We combine Lemmas 7.4, 7.5 and 7.6. From Lemma 7.4, we recall that our performance metrics may be written in terms of properties of the first cycle. From Lemma 7.5, we recall that these may be written in terms of x and y. Finally, in Lemma 7.6 we have found formulae for x and y. In order to write down the availability, we firstly combine (7.80) and (7.92), to find

$$\begin{aligned}\mathbb{E}[T_{\text{occ}}] = \mathbb{E}[T_N] &= \frac{1+y}{p_{\text{con}} + q(1-p_{\text{con}})(1-(1-p_{\text{gen}})^n)} \\ &= \frac{1-\tilde{A}+\tilde{C}H_{\text{new}}}{\left((1-\tilde{A})(1-\tilde{D})-\tilde{B}\tilde{C}\right)\tilde{P}} \\ &= \frac{1-\tilde{A}+\tilde{C}(F_{\text{new}}-\frac{1}{4})}{\left((1-\tilde{A})(1-\tilde{D})-\tilde{B}\tilde{C}\right)\tilde{P}}\end{aligned}$$

where $\tilde{P} := p_{\text{con}} + q(1-p_{\text{con}})(1-(1-p_{\text{gen}})^n)$. This suffices to show Theorem 7.1.

In order to write down the average consumed fidelity, we combine (7.81) and (7.92), to obtain

$$\begin{aligned}\overline{H} &\overset{\text{a.s.}}{=} \frac{\left[\tilde{B}-\tilde{D}H_{\text{new}}+H_{\text{new}}\right]\cdot\left[p_{\text{con}}+q\left(1-\left(1-p_{\text{gen}}\right)^n\right)(1-p_{\text{con}})\right]}{\left[1-\tilde{A}+\tilde{C}H_{\text{new}}\right]\cdot\left[e^{\Gamma}-\left(1-q\left(1-\left(1-p_{\text{gen}}\right)^n\right)\right)(1-p_{\text{con}})\right]} \\ &= \frac{q(1-p_{\text{con}})(\tilde{b}-\tilde{d}H_{\text{new}})+H_{\text{new}}\left(p_{\text{con}}+q\left(1-\left(1-p_{\text{gen}}\right)^n\right)(1-p_{\text{con}})\right)}{q(1-p_{\text{con}})(\tilde{c}H_{\text{new}}-\tilde{a})+e^{\Gamma}-\left(1-q\left(1-\left(1-p_{\text{gen}}\right)^n\right)\right)(1-p_{\text{con}})}\end{aligned}$$

where we have used the formulae (7.102) and (7.107) for $\tilde{A}$, $\tilde{B}$, $\tilde{C}$, and $\tilde{D}$. The above may be rewritten as

$$\overline{H} \overset{\text{a.s.}}{=} \frac{q(1-p_{\text{con}})(\tilde{b}-\tilde{d}H_{\text{new}})+H_{\text{new}}\left(p_{\text{con}}+qp^*_{\text{gen}}(1-p_{\text{con}})\right)}{q(1-p_{\text{con}})(\tilde{c}H_{\text{new}}-\tilde{a})+e^{\Gamma}-\left(1-qp^*_{\text{gen}}\right)(1-p_{\text{con}})} \tag{7.110}$$

$$= \frac{\left[p_{\text{con}}+q(1-p_{\text{con}})\left(p^*_{\text{gen}}-\tilde{d}\right)\right]\cdot H_{\text{new}}+q(1-p_{\text{con}})\tilde{b}}{\left[q(1-p_{\text{con}})\tilde{c}\right]\cdot H_{\text{new}}+e^{\Gamma}-1+p_{\text{con}}+q(1-p_{\text{con}})\left(p^*_{\text{gen}}-\tilde{a}\right)}, \tag{7.111}$$

where we have let $p^*_{\text{gen}} = 1-(1-p_{\text{gen}})^n$ be the effective probability of link generation. We now convert the above to $\overline{F}$. Recalling that $H_{\text{new}} = F_{\text{new}} - 1/4$, it follows that

$$\begin{aligned}\overline{F} &= \overline{H} + \frac{1}{4} \\ &\overset{\text{a.s.}}{=} \frac{\left[p_{\text{con}}+q(1-p_{\text{con}})\left(p^*_{\text{gen}}-\tilde{d}\right)\right]\cdot(F_{\text{new}}-\frac{1}{4})+q(1-p_{\text{con}})\tilde{b}}{\left[q(1-p_{\text{con}})\tilde{c}\right]\cdot(F_{\text{new}}-\frac{1}{4})+e^{\Gamma}-1+p_{\text{con}}+q(1-p_{\text{con}})\left(p^*_{\text{gen}}-\tilde{a}\right)}+\frac{1}{4} \\ &= \frac{\left[p_{\text{con}}+q(1-p_{\text{con}})\left(p^*_{\text{gen}}+\frac{\tilde{c}}{4}-\tilde{d}\right)\right]\cdot F_{\text{new}}+\frac{1}{4}\left[e^{\Gamma}-1+q(1-p_{\text{con}})\left(-\tilde{a}+4\tilde{b}-\frac{\tilde{c}}{4}+\tilde{d}\right)\right]}{\left[q(1-p_{\text{con}})\tilde{c}\right]\cdot F_{\text{new}}+e^{\Gamma}-1+p_{\text{con}}+q(1-p_{\text{con}})\left(p^*_{\text{gen}}-\tilde{a}-\frac{\tilde{c}}{4}\right)},\end{aligned}$$

which is our formula for the average consumed fidelity in terms of the system parameters, as is given in Theorem 7.2. □

7.8. [APPENDIX] PURIFICATION COEFFICIENTS

Here, we discuss the values that the coefficients a_k, b_k, c_k, and d_k of a purification protocol k are allowed to take. Note that these coefficients are in general functions of the newly generated state, ρ_{new}, although here we do not write this dependence explicitly for brevity. Then, in Subsection 7.8.1, we provide explicit expressions for the coefficients of the DEJMPS policy discussed in the main text.

The probability of success of the protocol is given by

$$p_k(F) = c_k\left(F - \frac{1}{4}\right) + d_k, \tag{7.112}$$

where the fidelity of the buffered state F can take values between $1/4$ (fully depolarized state) and 1 (perfect Bell pair). Since p_k is a probability, we must enforce $0 \leq p_k \leq 1$. At $F = 1/4$, this yields

$$0 \leq d_k \leq 1. \tag{7.113}$$

At $F = 1$, it yields

$$-\frac{4}{3}d_k \leq c_k \leq \frac{4}{3}(1 - d_k). \tag{7.114}$$

Combining (7.113) and (7.114) yields

$$-\frac{4}{3} \leq c_k \leq \frac{4}{3}. \tag{7.115}$$

The jump function (output fidelity) of the protocol is given by

$$J_k(F) = \frac{1}{4} + \frac{a_k\left(F - \frac{1}{4}\right) + b_k}{c_k\left(F - \frac{1}{4}\right) + d_k}. \tag{7.116}$$

This output fidelity must also be between $1/4$ (fully depolarized state) and 1 (perfect Bell pair). This condition can be written as $0 \leq a_k\left(F - \frac{1}{4}\right) + b_k \leq \frac{3}{4}p_k$. At $F = 1/4$, this yields

$$0 \leq b_k \leq \frac{3}{4}d_k. \tag{7.117}$$

Combining (7.117) and (7.113) yields

$$0 \leq b_k \leq \frac{3}{4}. \tag{7.118}$$

Similarly, the condition on the jump function at $F = 1$ can be written as

$$-\frac{4}{3}b_k \leq a_k \leq -\frac{4}{3}b_k + \frac{3}{4}c_k + d_k. \tag{7.119}$$

Combining (7.119) with (7.113), (7.114), and (7.117), we find

$$-1 \leq a_k \leq 1. \tag{7.120}$$

7.8.1. DEJMPS AND CONCATENATED DEJMPS POLICIES

As explained in the main text, the DEJMPS policy applies the well-known 2-to-1 DEJMPS purification protocol [53] to the buffered link and one of the newly generated links (and discarding the rest). This policy is given by the following purification coefficients:

$$\begin{aligned} a_k &= \frac{1}{6}\left(5\rho_{00}+\rho_{11}+\rho_{22}-3\rho_{33}\right), \\ b_k &= \frac{1}{24}\left(3\rho_{00}-3\rho_{11}-3\rho_{22}+5\rho_{33}\right), \\ c_k &= \frac{2}{3}\left(\rho_{00}-\rho_{11}-\rho_{22}+\rho_{33}\right), \\ d_k &= \frac{1}{2}\left(\rho_{00}+\rho_{11}+\rho_{22}+\rho_{33}\right), \end{aligned} \tag{7.121}$$

$\forall k \in \{1,...,n\}$, where ρ_{ii} are the diagonal elements of ρ_{new} in the Bell basis $\{|\phi^+\rangle, |\phi^-\rangle, |\psi^+\rangle, |\psi^-\rangle\}$. Note that we define the Bell states as follows:

$$|\phi^+\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}},\ |\phi^-\rangle = \frac{|00\rangle-|11\rangle}{\sqrt{2}},\ |\psi^+\rangle = \frac{|01\rangle+|10\rangle}{\sqrt{2}},\ |\psi^-\rangle = \frac{|01\rangle-|10\rangle}{\sqrt{2}}. \tag{7.122}$$

Regarding a concatenated or a nested DEJMPS policy, one can find the purification coefficients by applying (7.121) recursively. For each round of DEJMPS, the coefficients ρ_{ii} in (7.121) are the diagonal elements of the output state from the previous application of DEJMPS. These diagonal elements are given by [53]

$$\begin{aligned} \rho_{00} &= \frac{\sigma_{00}\sigma'_{00}+\sigma_{33}\sigma'_{33}}{P}, \\ \rho_{11} &= \frac{\sigma_{00}\sigma'_{33}+\sigma_{33}\sigma'_{00}}{P}, \\ \rho_{22} &= \frac{\sigma_{11}\sigma'_{11}+\sigma_{22}\sigma'_{22}}{P}, \\ \rho_{33} &= \frac{\sigma_{11}\sigma'_{22}+\sigma_{11}\sigma'_{22}}{P}, \end{aligned} \tag{7.123}$$

with $P = (\sigma_{00}+\sigma_{33})(\sigma'_{00}+\sigma'_{33})+(\sigma_{11}+\sigma_{22})(\sigma'_{11}+\sigma'_{22})$, where σ_{ii} and σ'_{ii} are the Bell diagonal elements of the two input states, σ and σ'.

7.8.2. OPTIMAL BILOCAL CLIFFORD POLICY

In the main text, we compare the concatenated versions of the DEJMPS policy to the optimal bilocal Clifford (optimal-bC) policy. When there is a buffered link in memory and k new links are generated, the optimal-bC policy operates as follows:

- When $k = 1$, DEJMPS is applied, using the buffered link and the newly generated link as inputs.
- When $k > 1$, the optimal k-to-1 purification protocol from ref. [95] is applied to all k new links. Then, the resulting state is used for DEJMPS, together with the buffered link. This is illustrated in Fig. 7.11b.

The reason why we apply an optimal bilocal Clifford protocol followed by DEJMPS is because these bilocal Clifford protocols have been shown to be optimal when the input states are identical. Hence, they ensure that the second link used in the final DEJMPS subroutine has maximum fidelity (see ref. [95] for a comparison of the output fidelity using the optimal protocol versus concatenated DEJMPS). This combined protocol (optimal k-to-1 followed by DEJMPS, Fig. 7.11b) is not necessarily the $(k+1)$-to-1 protocol that yields the largest output fidelity. However, one would expect it to provide better buffering performance than a simple concatenation of DEJMPS subroutines (Fig. 7.11a) – nevertheless, in the main text we show that this intuition is incorrect.

Let us now show how to compute the purification coefficients a_k, b_k, c_k, and d_k of the optimal-bC policy:

- When $k = 1$ new links are generated, the purification coefficients a_1, b_1, c_1, and d_1 are given by (7.121), as in the DEJMPS policy.

- When $k > 1$, we first apply the optimal bilocal Clifford protocol, which outputs a state σ_k, with diagonal elements in the Bell basis $\sigma_{k,ii}$. The probability of success of this subroutine is θ_k. Then, the state σ_k is used as input for a final DEJMPS subroutine. Using (7.121), we obtain

$$\begin{aligned} a_k &= \frac{1}{6\theta_k}\left(5\sigma_{k,00}+\sigma_{k,11}+\sigma_{k,22}-3\sigma_{k,33}\right), \\ b_k &= \frac{1}{24\theta_k}\left(3\sigma_{k,00}-3\sigma_{k,11}-3\sigma_{k,22}+5\sigma_{k,33}\right), \\ c_k &= \frac{2}{3\theta_k}\left(\sigma_{k,00}-\sigma_{k,11}-\sigma_{k,22}+\sigma_{k,33}\right), \\ d_k &= \frac{1}{2\theta_k}\left(\sigma_{k,00}+\sigma_{k,11}+\sigma_{k,22}+\sigma_{k,33}\right). \end{aligned} \tag{7.124}$$

In the example discussed in the main text, we consider $n = 4$. We also consider the newly generated links to be Werner states [197] with fidelity F_{new}, i.e.,

$$\rho_{\text{new}} = F_{\text{new}}\left|\phi^+\right\rangle\!\left\langle\phi^+\right| + \frac{1-F_{\text{new}}}{3}\left|\phi^-\right\rangle\!\left\langle\phi^-\right| + \frac{1-F_{\text{new}}}{3}\left|\psi^+\right\rangle\!\left\langle\psi^+\right| + \frac{1-F_{\text{new}}}{3}\left|\psi^-\right\rangle\!\left\langle\psi^-\right|. \tag{7.125}$$

Under these assumptions, the values of $\sigma_{k,00}$ (fidelity of the output state from the optimal bilocal Clifford protocol) and θ_k are given explicitly in ref. [95]:

$$\begin{aligned} \theta_2 &= \frac{8}{9}F_{\text{new}}^2 - \frac{4}{9}F_{\text{new}} + \frac{5}{9}, \\ \theta_3 &= \frac{32}{27}F_{\text{new}}^3 - \frac{4}{9}F_{\text{new}}^2 + \frac{7}{27}, \\ \theta_4 &= \frac{32}{27}F_{\text{new}}^4 - \frac{4}{9}F_{\text{new}}^2 + \frac{4}{27}F_{\text{new}} + \frac{1}{9}, \end{aligned} \tag{7.126}$$

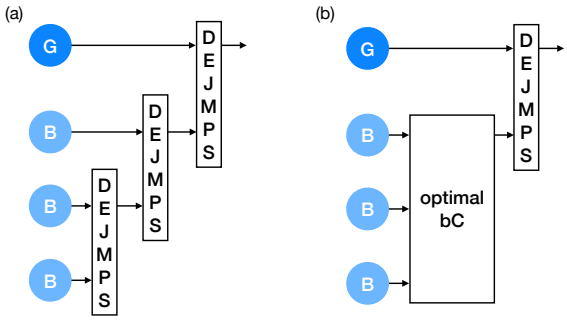

Figure 7.11: **The optimal bilocal Clifford policy applies an optimal protocol followed by DEJMPS.** Illustration of two purification policies: (a) concatenated DEJMPS and (b) optimal bilocal Clifford.

$$\begin{aligned}
\sigma_{2,00} &= \frac{1}{\theta_2}\cdot\left(\frac{10}{9}F_{\text{new}}^2 - \frac{2}{9}F_{\text{new}} + \frac{1}{9}\right),\\
\sigma_{3,00} &= \frac{1}{\theta_3}\cdot\left(\frac{28}{27}F_{\text{new}}^3 - \frac{1}{9}F_{\text{new}} + \frac{2}{27}\right),\\
\sigma_{4,00} &= \frac{1}{\theta_4}\cdot\left(\frac{8}{9}F_{\text{new}}^4 + \frac{8}{27}F_{\text{new}}^3 - \frac{2}{9}F_{\text{new}}^2 + \frac{1}{27}\right),
\end{aligned} \tag{7.127}$$

where F_{new} is the fidelity of the newly generated Werner states. The rest of the diagonal elements of σ_k can be found using the code provided in our repository[1]. For $F_{\text{new}} = 0.7$, which we use in the example from the main text, we have

$$\begin{cases} \sigma_{2,11} &= 0.20589\\ \sigma_{2,22} &= 0.02941\\ \sigma_{2,33} &= 0.02941 \end{cases}, \quad \begin{cases} \sigma_{3,11} &= 0.14287\\ \sigma_{3,22} &= 0.03571\\ \sigma_{3,33} &= 0.03571 \end{cases} \text{ and } \begin{cases} \sigma_{4,11} &= 0.04545\\ \sigma_{4,22} &= 0.04545\\ \sigma_{4,33} &= 0.04545 \end{cases}. \tag{7.128}$$

The calculations from ref. [95] can also be used to obtain θ_k and σ_k for $k > 4$, although their methods become infeasible for $k > 8$ due to the large computational cost, as discussed in their paper.

7.9. [APPENDIX] BUFFERING WITH THE 513 EC POLICY

In this appendix, we compare the performance of a 1GnB system that uses a concatenated DEJMPS policy to a system that uses the *513 EC policy*. When there is a buffered link in memory and k new links are generated, the 513 EC policy operates as follows:

- When $k = 1$, DEJMPS is applied, using the buffered link and the newly generated link as inputs.

[1] https://github.com/AlvaroGI/buffering-1GnB; the code we provide to find the diagonal elements of σ_k is based on the methods from ref. [95].

- When $k = 5$, the purification protocol based on the $[[5,1,3]]$ quantum error-correcting code [112] from ref. [173] is applied to all k new links. Then, the output state is twirled into a Werner state (that is, it is transformed into Werner form while preserving the fidelity) and used for DEJMPS, together with the buffered link.

- Otherwise, twice-concatenated DEJMPS is applied.

This policy is heavily based on twice-concatenated DEJMPS, with the main difference being that, when $k = 5$, a different protocol is applied. Note that, when $k = 5$, we apply some twirling after the purification step to be able to use the results reported in ref. [173], where they provide the output fidelity and the success probability of the protocol but not the full density matrix of the output state.

The purification coefficients of the 513 EC policy can be computed as follows:

- When $k \neq 5$, this policy applies DEJMPS or concatenated DEJMPS. Hence, a_k, b_k, c_k, and d_k can be found as explained in Appendix 7.8.1.

- When $k = 5$, the purification coefficients are given by the output fidelity and probability of success of the 513 EC protocol reported in Figure 3 from ref. [173]. Since we apply this protocol followed by twirling and DEJMPS, we can use (7.121) to compute the purification coefficients of the whole protocol:

$$\begin{aligned} a_5 &= \frac{1}{6\theta}\left(5\sigma_{00} + \sigma_{11} + \sigma_{22} - 3\sigma_{33}\right), \\ b_5 &= \frac{1}{24\theta}\left(3\sigma_{00} - 3\sigma_{11} - 3\sigma_{22} + 5\sigma_{33}\right), \\ c_5 &= \frac{2}{3\theta}\left(\sigma_{00} - \sigma_{11} - \sigma_{22} + \sigma_{33}\right), \\ d_5 &= \frac{1}{2\theta}\left(\sigma_{00} + \sigma_{11} + \sigma_{22} + \sigma_{33}\right), \end{aligned} \tag{7.129}$$

  where σ is the output state of the 513 EC protocol after twirling: σ_{00} is the output fidelity from the 513 protocol (reported in Figure 3 from ref. [173]), and $\sigma_{11} = \sigma_{22} = \sigma_{33} = (1-\sigma_{00})/3$ (since we twirl the output state); and θ is the probability of success of the 513 EC protocol (reported in Figure 3 from ref. [173]).

Figure 7.12 shows the performance of the 513 EC policy versus DEJMPS and twice-concatenated DEJMPS. In this example, twice-concatenated DEJMPS also includes twirling before the final round of DEJMPS, to make the comparison with the 513 EC policy fairer. In Fig. 7.12a, we assume $F_{\text{new}} = 0.86$ (according to Figure 3 from ref. [173], this corresponds to $\theta = 0.869$ and $\sigma_{00} = 0.864$), and in Fig. 7.12a, we assume $F_{\text{new}} = 0.95$ (according to Figure 3 from ref. [173], this corresponds to $\theta = 0.981$ and $\sigma_{00} = 0.978$). Similar to the optimal-bC policies discussed in the main text, we observe that the 513 EC policy can be outperformed by DEJMPS, twice-concatenated DEJMPS, and replacement (Figure 7.12a). In some parameter regions, the 513 EC may provide better performance (Figure 7.12b), although this behavior may not be achievable experimentally, as it requires both large p_{gen} and large F_{new} – in commonly used entanglement generation protocols, there is a tradeoff between these two parameters, see, e.g., ref. [84].

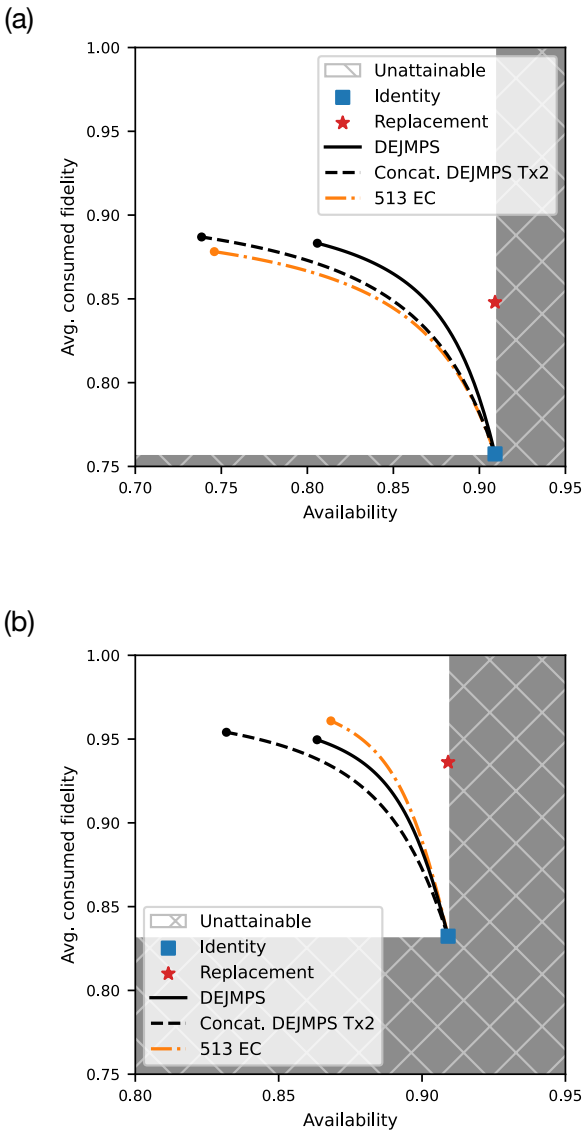

Figure 7.12: **The 513 EC policy may perform better than DEJMPS when new links are generated with a very large fidelity.** Performance of 1GnB systems with different purification policies, in terms of availability A and average consumed fidelity $\overline{F}$. In (a), newly generated links are Werner states with fidelity $F_{\text{new}} = 0.86$, while in (b) we assume $F_{\text{new}} = 0.95$. The shaded area corresponds to unattainable values of A and $\overline{F}$ (see (7.13) and (7.16)). Replacement (star marker) and identity (square marker) policies provide maximum availability. Lines represent the achievable values when using one of the following policies: DEJMPS (solid line), twice-concatenated DEJMPS with twirling (dashed line), and 513 EC (dotted line). See main text for a detail explanation of these policies. Parameter values used in this example: $n = 5$, $p_{\text{gen}} = 1$, $p_{\text{con}} = 0.1$, and $\Gamma = 0.02$.

7.10. [APPENDIX] MONOTONICITY OF THE AVAILABILITY

In this appendix, we show that the availability of the 1GnB system (given in Theorem 7.1) is monotonically decreasing with increasing probability of purification q (Proposition 7.1). This means that the availability is maximized when no purification is performed. If any purification is performed, the availability can only decrease, until reaching its minimum value at $q = 1$. Using these ideas, we compute upper and lower bounds for the availability in Section 7.10.1 below.

Proof of Proposition 7.1. We start by taking the partial derivative of the availability:

$$\frac{\partial A}{\partial q} = \frac{\mathbb{E}[T_{\text{gen}}]}{\left(\mathbb{E}[T_{\text{gen}}] + \mathbb{E}[T_{\text{occ}}]\right)^2} \frac{\partial \mathbb{E}[T_{\text{occ}}]}{\partial q}, \tag{7.130}$$

where we have used (7.8), (7.9), and (7.10). Since the first term in 7.130 is always positive, the sign of $\partial A/\partial q$ is the same as the sign of $\partial \mathbb{E}[T_{\text{occ}}]/\partial q$. Hence, we only need to show that $\partial \mathbb{E}[T_{\text{occ}}]/\partial q \leq 0$. Next, we write $\mathbb{E}[T_{\text{occ}}]$ explicitly in terms of q:

$$\mathbb{E}[T_{\text{occ}}] = \frac{\varepsilon + \varepsilon' q}{\delta + \delta' q + \delta'' q^2}, \tag{7.131}$$

where

$$\begin{aligned}
\varepsilon &:= \gamma + p_{\text{con}}, \\
\varepsilon' &:= (1 - p_{\text{con}}) \left(p^*_{\text{gen}} - \tilde{a} + H_{\text{new}} \tilde{c} \right), \\
\delta &:= \varepsilon p_{\text{con}}, \\
\delta' &:= (1 - p_{\text{con}}) \left(\gamma p^*_{\text{gen}} + 2 p_{\text{con}} p^*_{\text{gen}} - p_{\text{con}} \tilde{a} - (\gamma + p_{\text{con}}) \tilde{d} \right), \\
\delta'' &:= (1 - p_{\text{con}})^2 \left((p^*_{\text{gen}})^2 - p^*_{\text{gen}} \tilde{a} - p^*_{\text{gen}} \tilde{d} + \tilde{a}\tilde{d} - \tilde{b}\tilde{c} \right),
\end{aligned} \tag{7.132}$$

with $\gamma := e^{\Gamma} - 1$, $p^*_{\text{gen}} := 1 - (1 - p_{\text{gen}})^n$ and $H_{\text{new}} := F_{\text{new}} - \frac{1}{4}$. The derivative of $\mathbb{E}[T_{\text{occ}}]$ can be written as

$$\frac{\partial \mathbb{E}[T_{\text{occ}}]}{\partial q} = \frac{\varepsilon \left(\varepsilon' p_{\text{con}} - \delta' \right) - 2\varepsilon \delta'' q - \varepsilon' \delta'' q^2}{\left(\delta + \delta' q + \delta'' q^2 \right)^2}. \tag{7.133}$$

To prove that $\partial \mathbb{E}[T_{\text{occ}}]/\partial q \leq 0$, we will now show that all three terms in the numerator are negative.

FIRST TERM FROM (7.133) - The first term can be expanded as follows:

$$\varepsilon \left(\varepsilon' p_{\text{con}} - \delta' \right) = -\varepsilon (1 - p_{\text{con}}) \left(\gamma (p^*_{\text{gen}} - \tilde{d}) + p_{\text{con}} (p^*_{\text{gen}} - \tilde{d} - H_{\text{new}} \tilde{c}) \right) \geq 0, \tag{7.134}$$

where, in the last step, we have used the following: (i) $0 \leq p_{\text{con}} \leq 1$, (ii) $\gamma := e^{\Gamma} - 1 \geq 0$,

(iii) $\tilde{d} + H_{\text{new}}\tilde{c} \leq p^*_{\text{gen}}$, and (iv) $\tilde{d} \leq p^*_{\text{gen}}$. Inequality (iii) can be shown as follows:

$$\begin{aligned}\tilde{d} + H_{\text{new}}\tilde{c} &= \sum_{k=1}^{n} (d_k + H_{\text{new}} c_k) \binom{n}{k} (1-p_{\text{gen}})^{n-k} p_{\text{gen}}^k \\ &\leq \sum_{k=1}^{n} \binom{n}{k} (1-p_{\text{gen}})^{n-k} p_{\text{gen}}^k \\ &= \sum_{k=0}^{n} \binom{n}{k} (1-p_{\text{gen}})^{n-k} p_{\text{gen}}^k - (1-p_{\text{gen}})^n \\ &= 1 - (1-p_{\text{gen}})^n = p^*_{\text{gen}},\end{aligned} \tag{7.135}$$

where we have used the definition of $\tilde{c}$ and $\tilde{d}$ from Theorem 7.1 and the fact that $d_k + H_{\text{new}} c_k \leq 1$ (this is the success probability of purification protocol k when the link in memory has fidelity F_{new}). Inequality (iv) can be shown in a similar way:

$$\begin{aligned}\tilde{d} &= \sum_{k=1}^{n} d_k \binom{n}{k} (1-p_{\text{gen}})^{n-k} p_{\text{gen}}^k \\ &\leq \sum_{k=1}^{n} \binom{n}{k} (1-p_{\text{gen}})^{n-k} p_{\text{gen}}^k \\ &= \sum_{k=0}^{n} \binom{n}{k} (1-p_{\text{gen}})^{n-k} p_{\text{gen}}^k - (1-p_{\text{gen}})^n \\ &= 1 - (1-p_{\text{gen}})^n = p^*_{\text{gen}},\end{aligned} \tag{7.136}$$

where we have used $d_k \leq 1$ (upper bound from (7.113)).

SECOND TERM FROM (7.133) - Regarding the second term in the numerator of (7.133), we first note that, since $p_{\text{con}} \geq 0$ and $\gamma \geq 0$, then $\varepsilon \geq 0$. Moreover, $q \geq 0$ by definition. Consequently, the second term in the numerator of (7.133) is negative if and only if $\delta'' \geq 0$, which in turn is equivalent to $(p^*_{\text{gen}})^2 - p^*_{\text{gen}}\tilde{a} - p^*_{\text{gen}}\tilde{d} + \tilde{a}\tilde{d} - \tilde{b}\tilde{c} \geq 0$. This can be shown as follows:

$$\begin{aligned}(p^*_{\text{gen}})^2 - p^*_{\text{gen}}\tilde{a} - p^*_{\text{gen}}\tilde{d} + \tilde{a}\tilde{d} - \tilde{b}\tilde{c} &\overset{\text{i}}{\geq} (p^*_{\text{gen}})^2 - p^*_{\text{gen}}\tilde{a} - p^*_{\text{gen}}\tilde{d} + \tilde{a}\tilde{d} - \tilde{b}\frac{4}{3}(p^*_{\text{gen}} - \tilde{d}) \\ &= \left(p^*_{\text{gen}} - \frac{4}{3}\tilde{b} - \tilde{a}\right)(p^*_{\text{gen}} - \tilde{d}) \\ &\overset{\text{ii}}{\geq} 0,\end{aligned} \tag{7.137}$$

with these steps:

i. We use $\tilde{b} \geq 0$ (which follows from the lower bound in (7.118)) and $\tilde{c} \leq (p^*_{\text{gen}} - \tilde{d})/H_{\text{new}}$ (shown in (7.135)). This last inequality must hold for any $H_{\text{new}} \in [0, 3/4]$, and therefore $\tilde{c} \leq 4(p^*_{\text{gen}} - \tilde{d})/3$.

ii. To show that the first factor is non-negative, we use $\tilde{a} + 4\tilde{b}/3 \leq \tilde{d} + H_{\text{new}}\tilde{c} \leq p^*_{\text{gen}}$. The first inequality can be shown using the definitions of $\tilde{a}$, $\tilde{b}$, $\tilde{c}$, and $\tilde{d}$ from Theorem 7.1

and the upper bound from (7.119); while the second inequality was shown in (7.135). The second factor $(p^*_{\text{gen}} - \tilde{d})$ is also non-negative, as shown in (7.136).

THIRD TERM FROM (7.133) - Lastly, the third term in the numerator of (7.133) is negative if and only if $\varepsilon' \geq 0$, since we just showed that $\delta'' \geq 0$. Moreover, $\varepsilon' \geq 0 \Leftrightarrow p^*_{\text{gen}} - \tilde{a} + H_{\text{new}}\tilde{c} \geq 0$. The latter can be shown as follows:

$$\begin{aligned}
p^*_{\text{gen}} - \tilde{a} + H_{\text{new}}\tilde{c} &\overset{\text{i}}{=} p^*_{\text{gen}} + \sum_{k=1}^{n} (-a_k + H_{\text{new}} c_k) \binom{n}{k} (1-p_{\text{gen}})^{n-k} p_{\text{gen}}^k \\
&\overset{\text{ii}}{\geq} p^*_{\text{gen}} + \sum_{k=1}^{n} \left(-\left(\frac{3}{4} - H_{\text{new}}\right) c_k - d_k \right) \binom{n}{k} (1-p_{\text{gen}})^{n-k} p_{\text{gen}}^k \\
&\overset{\text{iii}}{\geq} p^*_{\text{gen}} + \sum_{k=1}^{n} \left(\frac{4}{3} H_{\text{new}}(1-d_k) - 1 \right) \binom{n}{k} (1-p_{\text{gen}})^{n-k} p_{\text{gen}}^k \\
&\overset{\text{iv}}{\geq} p^*_{\text{gen}} - \sum_{k=1}^{n} \binom{n}{k} (1-p_{\text{gen}})^{n-k} p_{\text{gen}}^k \\
&\overset{\text{v}}{=} 0,
\end{aligned} \tag{7.138}$$

with these steps:

i. We use the definitions of $\tilde{a}$ and $\tilde{c}$ from Theorem 7.1.

ii. We use $a_k \leq 3c_k/4 + d_k$, which can be shown using the upper bound from (7.119) in combination with the lower bound from (7.118).

iii. We use $c_k \leq 4(1-d_k)/3$ (upper bound from (7.114)).

iv. We note that $H_{\text{new}}(1-d_k) \geq 0$, since $H_{\text{new}} \geq 0$ (by definition) and $d_k \leq 1$ (as shown in (7.113)).

v. We recall the definition $p^*_{\text{gen}} := 1 - (1-p_{\text{gen}})^n = \sum_{k=0}^{n} \binom{n}{k} (1-p_{\text{gen}})^{n-k} p_{\text{gen}}^k - (1 - p_{\text{gen}})^n$.

We have now shown that all three terms in the numerator of (7.133) are negative. Therefore, $\partial \mathbb{E}[T_{\text{occ}}]/\partial q \leq 0$ and, consequently, $\partial A/\partial q \leq 0$. □

7.10.1. UPPER AND LOWER BOUNDING THE AVAILABILITY

Since $\partial A/\partial q \leq 0$, the availability is upper bounded by the value it takes when $q = 0$. From (7.131), we have

$$\mathbb{E}[T_{\text{occ}}]\big|_{q=0} = \frac{1}{p_{\text{con}}}. \tag{7.139}$$

Combining this with (7.8), we obtain

$$A \leq A\big|_{q=0} = \frac{p^*_{\text{gen}}}{p^*_{\text{gen}} + p_{\text{con}}}, \tag{7.140}$$

with $p^*_{\text{gen}} := 1 - (1-p_{\text{gen}})^n$.

To evaluate A at $q=1$, we first use (7.8) and (7.131) to write it as follows:

$$A \geq A\big|_{q=1} = \frac{p^*_{\text{gen}}\eta}{p^*_{\text{gen}}\eta + \Delta}, \tag{7.141}$$

with $\eta := \varepsilon + \varepsilon'$, $\Delta := \delta + \delta' + \delta''$, with $\varepsilon, \varepsilon', \delta, \delta', \delta''$ defined in (7.132).

The solution from (7.141) constitutes a lower bound for the availability. However, η and Δ implicitly depend on the parameters of the purification policy, a_k, b_k, c_k, and d_k, $k \in \{0, \ldots, n\}$. Next, we find a more general and meaningful lower bound that applies to any purification policy.

We start by noting that

- $\varepsilon \geq 0$ (since $p_{\text{con}} \geq 0$ and $\gamma \geq 0$),
- $\varepsilon' \geq 0$ (as shown in (7.138)),
- $\delta \geq 0$ (since $\varepsilon \geq 0$),
- $\delta' \geq 0$ (this can be shown using the fact that $\tilde{d} \leq p^*_{\text{gen}}$, shown in (7.136), and $\tilde{a} \leq p^*_{\text{gen}}$, which can be shown in a similar way as (7.136) and using (7.120)),
- and $\delta'' \geq 0$ (as shown in (7.137)).

As a consequence, none of the factors in (7.141) can be negative: $p^*_{\text{gen}} \geq 0$ (by definition), $\eta \geq 0$, and $\Delta \geq 0$. This means that we can find a lower bound for $A\big|_{q=1}$ by lower bounding η and upper bounding Δ. We first lower bound η:

$$\eta = \gamma + p_{\text{con}} + (1 - p_{\text{con}})\left(p^*_{\text{gen}} - \tilde{a} + H_{\text{new}}\tilde{c}\right) \geq \gamma + p_{\text{con}}, \tag{7.142}$$

where we have used $p^*_{\text{gen}} - \tilde{a} + H_{\text{new}}\tilde{c} \geq 0$, which was shown in (7.138).

Regarding Δ, we proceed as follows:

$$\begin{aligned}
\Delta =& \delta + \delta' + \delta'' \\
=& (\gamma + p_{\text{con}})p_{\text{con}} + (1 - p_{\text{con}})\left((\gamma + 2p_{\text{con}})p^*_{\text{gen}} - p_{\text{con}}(\tilde{a} + \tilde{d}) - \gamma\tilde{d}\right) \\
&+ (1 - p_{\text{con}})^2\left((p^*_{\text{gen}})^2 - p^*_{\text{gen}}(\tilde{a} + \tilde{d}) + \tilde{a}\tilde{d} - \tilde{b}\tilde{c}\right) \\
\overset{\text{i}}{\leq}& (\gamma + p_{\text{con}})p_{\text{con}} + (1 - p_{\text{con}})\left(\gamma p^*_{\text{gen}} + 2p_{\text{con}}p^*_{\text{gen}} - p_{\text{con}}\tilde{a}\right) \\
&+ (1 - p_{\text{con}})^2\left((p^*_{\text{gen}})^2 - p^*_{\text{gen}}\tilde{a} + \tilde{a}\tilde{d} - \tilde{b}\tilde{c}\right) \\
\overset{\text{ii}}{\leq}& (\gamma + p_{\text{con}})p_{\text{con}} + (1 - p_{\text{con}})\left(\gamma p^*_{\text{gen}} + 2p_{\text{con}}p^*_{\text{gen}} + p_{\text{con}}p^*_{\text{gen}}\right) \\
&+ (1 - p_{\text{con}})^2\left((p^*_{\text{gen}})^2 + (p^*_{\text{gen}})^2 + \tilde{a}\tilde{d} - \tilde{b}\tilde{c}\right) \\
=& (\gamma + p_{\text{con}})p_{\text{con}} + (1 - p_{\text{con}})\left(\gamma + 3p_{\text{con}}\right)p^*_{\text{gen}} + (1 - p_{\text{con}})^2\left(2(p^*_{\text{gen}})^2 + \tilde{a}\tilde{d} - \tilde{b}\tilde{c}\right) \\
\overset{\text{iii}}{\leq}& (\gamma + p_{\text{con}})p_{\text{con}} + (1 - p_{\text{con}})\left(\gamma + 3p_{\text{con}}\right)p^*_{\text{gen}} + (1 - p_{\text{con}})^2\left(2(p^*_{\text{gen}})^2 + p^*_{\text{gen}}\right) \\
=& (\gamma + p_{\text{con}})p_{\text{con}} + (1 - p_{\text{con}})\left(1 + \gamma + 2p_{\text{con}}\right)p^*_{\text{gen}} + 2(1 - p_{\text{con}})^2(p^*_{\text{gen}})^2
\end{aligned} \tag{7.143}$$

with these steps:

i. We use $-(\gamma + p_{\text{con}})\tilde{d} \leq 0$ and $-p^*_{\text{gen}}\tilde{d} \leq 0$, which follows from $\tilde{d} \geq 0$ (shown in (7.113)).

ii. Using the lower bound from (7.120) and following a similar derivation as in (7.136), one can show that $\tilde{a} \geq -p^*_{\text{gen}}$. This implies that $-p_{\text{con}}\tilde{a} \leq p_{\text{con}}p^*_{\text{gen}}$ and $-p^*_{\text{gen}}\tilde{a} \leq (p^*_{\text{gen}})^2$.

iii. We use $\tilde{a}\tilde{d} - \tilde{b}\tilde{c} \leq p^*_{\text{gen}}$. This can be shown as follows:

$$\tilde{a}\tilde{d} - \tilde{b}\tilde{c} \leq -\frac{4}{3}\tilde{b}\tilde{d} + \frac{3}{4}\tilde{c}\tilde{d} + \tilde{d}^2 - \tilde{b}\tilde{c} \leq -\frac{4}{3}\tilde{b}\tilde{d} + (1-\tilde{d})\tilde{d} + \tilde{d}^2 - \tilde{b}\tilde{c} \leq \tilde{d} \leq p^*_{\text{gen}}, \quad (7.144)$$

where we have used the upper bound from (7.119) in the first step; $\tilde{d} \geq 0$ (see (7.113)) and the upper bound from (7.114) in the second step; $\tilde{b} \geq 0$ (see (7.118)) and the lower bound from (7.114) in the third step; and (7.136) in the last step.

Lastly, combining (7.141) with the bounds from (7.142) and (7.143), we obtain

$$A \geq A\big|_{q=1} = \frac{p^*_{\text{gen}}\eta}{p^*_{\text{gen}}\eta + \Delta} \geq \frac{p^*_{\text{gen}}\left(\gamma + p_{\text{con}}\right)}{\xi + \xi' p^*_{\text{gen}} + \xi''(p^*_{\text{gen}})^2}, \quad (7.145)$$

with $\xi := \gamma p_{\text{con}} + p^2_{\text{con}}$, $\xi' := 1 + 2\gamma + (2-\gamma)p_{\text{con}} - 2p^2_{\text{con}}$, and $\xi'' := 2(1-p_{\text{con}})^2$. This lower bound is general and applies to every 1GnB system, no matter which purification policy it employs.

7.11. [APPENDIX] MONOTONICITY OF THE AVERAGE CONSUMED FIDELITY

In this appendix, we show that the average consumed fidelity of the 1GnB system (given in Theorem 7.2) is monotonically increasing with increasing probability of purification q (Proposition 7.2), as long as the purification policy is made of protocols that can increase the fidelity of newly generated links (i.e., $J_k(F_{\text{new}}) \geq F_{\text{new}}$, $\forall k \in \{1, ..., n\}$). This means that the average consumed fidelity is maximized when purification is performed every time a new link is generated ($q = 1$). Using these ideas, we compute upper and lower bounds for the average consumed fidelity in Section 7.11.1.

Proof of Proposition 7.2. Recalling from (7.69) that $\overline{F} = \overline{H} + 1/4$, showing the monotonicity of $\overline{H}$ is equivalent to showing the monotonicity of $\overline{F}$. We firstly rewrite the formula for $\overline{H}$ as given in (7.110),

$$\overline{H} = \frac{q(1-p_{\text{con}})\left[\tilde{b} - \tilde{d}H_{\text{new}} + H_{\text{new}}p^*_{\text{gen}}\right] + H_{\text{new}}p_{\text{con}}}{q(1-p_{\text{con}})\left[\tilde{c}H_{\text{new}} - \tilde{a} + p^*_{\text{gen}}\right] + e^{\Gamma} - 1 + p_{\text{con}}},$$

where $p^*_{\text{gen}} = 1 - (1-p_{\text{gen}})^n$. Now consider functions of the form $g(x) = \frac{\alpha x + \beta}{\gamma x + \delta}$. This is

non-decreasing if and only if

$$\frac{\mathrm{d}g}{\mathrm{d}x} = \frac{\alpha\delta - \beta\gamma}{(\gamma x + \delta)^2} \geq 0$$
$$\Leftrightarrow \alpha\delta - \beta\gamma \geq 0.$$

We therefore see that $\overline{H}$ is non-decreasing in q if and only if

$$(1-p_{\mathrm{con}})\left(\left[\tilde{b} - \tilde{d}H_{\mathrm{new}} + H_{\mathrm{new}}p^*_{\mathrm{gen}}\right](e^{\Gamma} - 1 + p_{\mathrm{con}}) - H_{\mathrm{new}}p_{\mathrm{con}}\left[\tilde{c}H_{\mathrm{new}} - \tilde{a} + p^*_{\mathrm{gen}}\right]\right) \geq 0,$$

or equivalently

$$(e^{\Gamma} - 1)\left(\tilde{b} - \tilde{d}H_{\mathrm{new}} + H_{\mathrm{new}}p^*_{\mathrm{gen}}\right) + p_{\mathrm{con}}\left(\tilde{b} - \tilde{d}H_{\mathrm{new}} - \tilde{c}H^2_{\mathrm{new}} + \tilde{a}H_{\mathrm{new}}\right) \geq 0 \tag{7.146}$$

We now show this by considering the two parts of the expression:

(a) $\tilde{b} - \tilde{d}H_{\mathrm{new}} + H_{\mathrm{new}}p^*_{\mathrm{gen}} \geq 0$

Recall that the jump functions $\tilde{J}_k$ map the shifted fidelity h as

$$\tilde{J}_k(h) = \frac{a_k h + b_k}{c_k h + d_k}. \tag{7.147}$$

When the input state is completely mixed ($h = 0$), the probability of successful purification is

$$\tilde{p}_k(0) = d_k,$$

and so we must have $0 \leq d_k \leq 1$. If $d_k > 0$, the output fidelity when inputting a completely mixed state then satisfies

$$\tilde{J}_k(0) = \frac{b_k}{d_k} \geq 0$$

which implies $b \geq 0$. If $d = 0$, the output fidelity as the input state approaches the completely mixed state is

$$\lim_{h\to 0} \frac{a_k h + b_k}{c_k h},$$

and since this is bounded, it must be the case that $b = 0$. Therefore,

$$\tilde{b} = \sum_{k=1}^{n} b_k \cdot \binom{n}{k} (1-p_{\mathrm{gen}})^{n-k} p^k_{\mathrm{gen}} \geq 0$$

and

$$\tilde{d} = \sum_{k=1}^{n} d_k \cdot \binom{n}{k} (1-p_{\mathrm{gen}})^{n-k} p^k_{\mathrm{gen}}$$
$$\leq \sum_{k=1}^{n} \binom{n}{k} (1-p_{\mathrm{gen}})^{n-k} p^k_{\mathrm{gen}} = 1 - (1-p_{\mathrm{gen}})^n = p^*_{\mathrm{gen}}.$$

Combining the above, we obtain

$$\tilde{b} - \tilde{d}H_{\mathrm{new}} + H_{\mathrm{new}}p^*_{\mathrm{gen}} \geq H_{\mathrm{new}}(p^*_{\mathrm{gen}} - \tilde{d}) \geq 0.$$

(b) $\tilde{b} - \tilde{d}H_{\text{new}} - \tilde{c}H_{\text{new}}^2 + \tilde{a}H_{\text{new}} \geq 0$
We have that

$$\begin{aligned}\tilde{b} - \tilde{d}H_{\text{new}} - \tilde{c}H_{\text{new}}^2 + \tilde{a}H_{\text{new}} &= \sum_{k=1}^{n} \binom{n}{k} (1-p_{\text{gen}})^{n-k} p_{\text{gen}}^k \\ &\qquad \cdot (b_k - d_k H_{\text{new}} - c_k H_{\text{new}}^2 + a_k H_{\text{new}}) \\ &= \sum_{k=1}^{n} \binom{n}{k} (1-p_{\text{gen}})^{n-k} p_{\text{gen}}^k \\ &\qquad \cdot \left(\frac{a_k H_{\text{new}} + b_k}{c_k H_{\text{new}} + d_k} - H_{\text{new}} \right) (c_k H_{\text{new}} + d_k) \\ &= \sum_{k=1}^{n} \binom{n}{k} (1-p_{\text{gen}})^{n-k} p_{\text{gen}}^k \\ &\qquad \cdot \left(\tilde{J}_k(H_{\text{new}}) - H_{\text{new}} \right) \cdot \tilde{p}_k(H_{\text{new}}),\end{aligned}$$

which is non-negative if all jump functions $\tilde{J}_k$ satisfy

$$\tilde{J}_k(H_{\text{new}}) \geq H_{\text{new}},$$

or equivalently $J_k(F_{\text{new}}) \geq F_{\text{new}}$ Since $\Gamma \geq 0$, we therefore see that (7.146) holds. □

7.11.1. UPPER AND LOWER BOUNDING THE AVERAGE CONSUMED FIDELITY

Here, we only consider purification policies made of protocols that can increase the fidelity of newly generated links (i.e., $J_k(F_{\text{new}}) \geq F_{\text{new}}$, $\forall k \in \{1, ..., n\}$). For these policies, $\partial \overline{F}/\partial q \geq 0$. A tight lower bound can be found by setting $q = 0$ in (7.11):

$$\overline{F} \geq \overline{F}\big|_{q=0} = \frac{\gamma/4 + F_{\text{new}} p_{\text{con}}}{\gamma + p_{\text{con}}}, \tag{7.148}$$

where $\gamma := e^{\Gamma} - 1$.

An upper bound for $\overline{F}$ can be found by upper bounding it maximum value, which occurs at $q = 1$. Using (7.110), we can write the maximum value as

$$\overline{F}\big|_{q=1} = \frac{1}{4} + \frac{(1-p_{\text{con}})(\tilde{b} - \tilde{d}H_{\text{new}}) + H_{\text{new}}\left(p_{\text{con}} + p_{\text{gen}}^*(1-p_{\text{con}})\right)}{(1-p_{\text{con}})(\tilde{c}H_{\text{new}} - \tilde{a}) + \gamma + p_{\text{con}} + (1-p_{\text{con}})p_{\text{gen}}^*}, \tag{7.149}$$

where $p_{\text{gen}}^* = 1 - (1-p_{\text{gen}})^n$. Using (7.117) and (7.136), it can be shown that $\tilde{b} - \tilde{d}H_{\text{new}} \leq p_{\text{gen}}^*(3/4 - H_{\text{new}})$. Moreover, from (7.138), we know that $H_{\text{new}}\tilde{c} - \tilde{a} \geq -p_{\text{gen}}^*$. Applying these two inequalities to (7.149), we find the upper bound:

$$\overline{F} \leq \overline{F}\big|_{q=1} \leq \frac{1}{4} + \frac{H_{\text{new}} p_{\text{con}} + (3/4)(1-p_{\text{con}})p_{\text{gen}}^*}{\gamma + p_{\text{con}}} = \frac{\gamma/4 + F_{\text{new}} p_{\text{con}}}{\gamma + p_{\text{con}}} + \frac{3}{4}\frac{(1-p_{\text{con}})p_{\text{gen}}^*}{\gamma + p_{\text{con}}}. \tag{7.150}$$

7.12. [APPENDIX] CONCATENATED PURIFICATION

In this appendix, we discuss further features of 1GnB buffers that use concatenated purification policies. In 7.12.1, we consider different orderings for the purification subroutines that are being concatenated. In 7.12.2, we show that increasing the number of concatenations is beneficial when noise in memory is very strong.

7.12.1. DIFFERENT CONCATENATION ORDERINGS

As stated in the main text, we tested different orderings of the concatenated purification subroutines. In Figure 7.5, we showed two different orderings for a concatenated DEJMPS policy: sequentially concatenated DEJMPS and nested DEJMPS. Here, we consider a policy that applies a nested DEJMPS protocol to all the newly generated links, and then uses the output state to purify the link in memory with a final round of DEJMPS. This policy is only defined when the number of links generated is a power of 2. Hence, we assume $n = 4$ bad memories and deterministic entanglement generation ($p_{\text{gen}} = 1$) in the following example. Figure 7.13 shows the performance of this policy compared to concatenated versions of DEJMPS (in which DEJMPS is applied sequentially to all links, as shown in Figure 7.5a). The performance of all policies shown is qualitatively similar. We also observe that, in this case, nesting is better than concatenating as much as possible, but it is worse than concatenating twice.

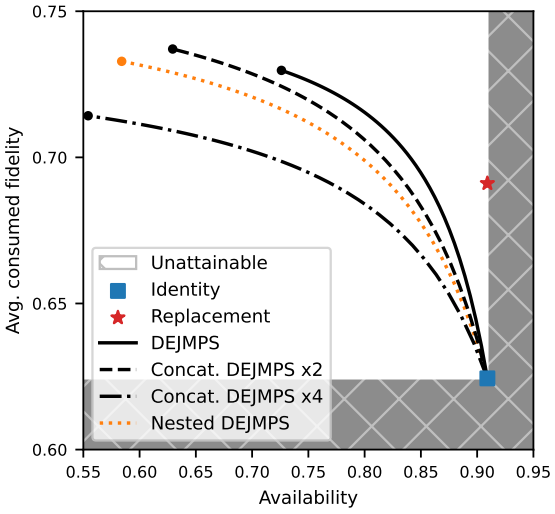

Figure 7.13: **Different concatenation orderings seem to yield qualitatively similar performance.** Performance of 1GnB systems with different purification policies, in terms of availability A and average consumed fidelity $\overline{F}$. The shaded area corresponds to unattainable values of A and $\overline{F}$ (see (7.13) and (7.16)). Lines and markers show the combinations of A and $\overline{F}$ achievable by different purification policies: identity (square marker), replacement (star marker), DEJMPS (solid line), twice-concatenated DEJMPS (dashed line), thrice-concatenated DEJMPS (dotted-dashed line), and nested DEJMPS (orange dotted line). Parameter values used in this example: $n = 4$, $p_{\text{gen}} = 1$, $F_{\text{new}} = 0.7$ (ρ_{new} is a Werner state), $p_{\text{con}} = 0.1$, and $\Gamma = 0.02$.

7.12.2. INCREASING NUMBER OF CONCATENATIONS

In the main text, we showed that using some newly generated entangled links in the purification protocol and discarding the rest may provide a better buffering performance than implementing a more complex protocol that uses all the newly generated links. In particular, we showed that increasing the maximum number of concatenations in a concatenated DEJMPS policy does not necessarily lead to better performance. The reason was that, as we increase the number of concatenations, the overall probability of success of the protocol decreases. Nevertheless, this effect is irrelevant when noise is strong: the quality of the buffered entanglement decays so rapidly that we need a protocol that can compensate noise with large boosts in fidelity, even if the probability of failure is large. This is shown in Figure 7.14, where we display the maximum average consumed fidelity (i.e., assuming purification probability $q = 1$, see Proposition 7.2) versus the number of concatenations. When no purification is applied (zero concatenations), $\overline{F}$ is below 0.5, meaning that the good memory stores no entanglement, on average (see Section 6.10.3). As we increase the number of concatenations in the purification protocol, $\overline{F}$ increases, although the increase is marginal. Note that this is a consequence of the strong noise experienced by the buffered entanglement – in Figure 7.7 we showed the same plot but considering a lower noise level and the conclusions were different: increasing the number of concatenations eventually led to a decrease in average consumed fidelity.

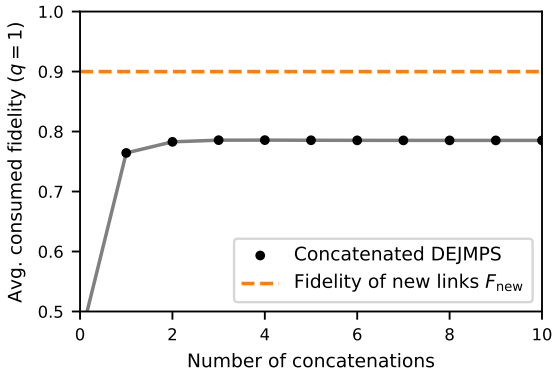

Figure 7.14: **Additional concatenation may improve the performance when noise is strong.** Maximum average consumed fidelity $\overline{F}$ achieved by a purification policy that concatenates DEJMPS a limited number of times. Zero concatenations corresponds to an identity policy (no purification is performed). One concatenation corresponds to the DEJMPS policy. Parameter values used in this example: $n = 10$, $p_{\text{gen}} = 0.5$, $F_{\text{new}} = 0.9$ (ρ_{new} is a Werner state), $p_{\text{con}} = 0.1$, and $\Gamma = 0.2$.

CODE AVAILABILITY

The code used to perform the analysis and generate all the plots shown in this chapter can be found in the following GitHub repository: https://github.com/AlvaroGI/buffering-1GnB. This repository also includes a discrete-event simulator of a 1GnB system that we

used to validate our analytical results.

AUTHOR CONTRIBUTIONS

ÁGI and BD conceived and defined the project. BD and SK proved Theorems 7.1 and 7.2. ÁGI and BD proved Propositions 7.1 and 7.2. ÁGI carried out the analysis from Sections 7.3 and 7.4, and coded the discrete-event simulation (used to validate analytical results). ÁGI and BD wrote the manuscript. SW provided active feedback at every stage of the project.

8

CONCLUSIONS

In this dissertation, we aimed to guide engineering efforts, through theoretical insights, towards practical entanglement distribution in quantum networks. Over the course of our research, we learned important lessons. The following results are of particular relevance, as they offer crucial guidance for quantum network design and naturally give rise to research directions that could play a fundamental role in advancing the field in the near future.

- Optimal strategies for entanglement distribution in simplified settings serve as valuable benchmarks for assessing more realistic strategies constrained by practical limitations. The methods and results from Chapter 2 are particularly useful for benchmarking entanglement distribution policies in quantum repeater chains. However, the computational cost of algorithms for finding optimal policies can be prohibitive. For example, the complexity of the algorithms used in Chapter 2 grows exponentially with the number of repeaters in the chain. Consequently, **finding optimal policies in complex quantum networks may require alternative methods, such as reinforcement learning [175]**, which, despite not guaranteeing optimality, can yield sufficiently good solutions. This approach has gained increasing attention in recent years (see, e.g., refs. [78, 117, 156]).

- In Chapter 3, we discuss how to employ one-way quantum repeaters and a quantum circuit switching approach to meet entanglement needs over metropolitan distances. While the alternative strategy – quantum packet switching – has been explored in the literature [54], an **in-depth quantitative comparison between circuit switching and packet switching in quantum networks** remains an open question, to the best of our knowledge. This is a necessary step to determine the most effective way of operating a network of one-way quantum repeaters.

- In some parameter regimes, protocols for continuous entanglement distribution are able to meet entanglement requests at a much faster rate than on-demand strategies, as discussed in Chapter 5. However, this type of protocol may also lead

to a waste of entangled states: if entanglement is distributed much earlier than needed, it will degrade due to decoherence and may need to be discarded. Consequently, continuous distribution is often overlooked in favor of less wasteful on-demand schemes. This concern is particularly valid given that state-of-the-art quantum technologies are costly and shared entangled states are highly valuable. Looking ahead, we hope that the cost of entanglement distribution will decrease as technology matures, and therefore **we encourage the community to consider continuous-distribution protocols** as a viable alternative that can provide enhanced performance at the expense of wasting some entanglement.

- Lastly, we expect our work on entanglement buffers to open new avenues towards efficient entanglement-distribution in larger networks, as they allow users to distribute entanglement in advance and keep it protected from time-dependent noise by continuously applying purification subroutines. The closed-form solutions found in Chapter 7 for the performance of purification-based buffers are particularly relevant, as they can be evaluated with negligible computational cost across all parameter regimes. This enables seamless **integration of entanglement buffers into the study of complex systems**, from quantum computing clusters to continental-scale quantum networks.

We would like to conclude this dissertation with a short reflection on the societal relevance of our work. Large part of our contributions aims to enable networking applications that require entanglement shared among remote parties, such as cryptographic primitives [12, 59], protocols for distributed quantum computing [26, 46], and distributed quantum sensing experiments [74, 152, 202]. Consequently, the long-term impact of our results will only become clear as quantum technologies mature and we agree on the relevance of such applications. This process will not only depend on scientific and technological considerations but also on the needs and values of a rapidly evolving society.

REFERENCES

1. Abobeih, M. H. *et al.* One-second coherence for a single electron spin coupled to a multi-qubit nuclear-spin environment. *Nat. Commun.* **9,** 2552 (2018).
2. Acín, A., Cirac, J. I. & Lewenstein, M. Entanglement percolation in quantum networks. *Nat. Phys.* **3,** 256–259 (2007).
3. Adan, I. & Resing, J. Queueing theory. *Eindhoven University of Technology* **180** (2002).
4. Aparicio, L. & Van Meter, R. Multiplexing schemes for quantum repeater networks. in *Quantum Communications and Quantum Imaging IX* **8163** (2011), 816308.
5. Askarani, M. F., Chakraborty, K. & Do Amaral, G. C. Entanglement distribution in multi-platform buffered-router-assisted frequency-multiplexed automated repeater chains. *New J. Phys.* **23,** 063078 (2021).
6. Avis, G., Rozpędek, F. & Wehner, S. Analysis of multipartite entanglement distribution using a central quantum-network node. *Phys. Rev. A* **107,** 012609 (2023).
7. Azuma, K., Tamaki, K. & Lo, H.-K. All-photonic quantum repeaters. *Nat. Commun.* **6,** 1–7 (2015).
8. Baran, P. On distributed communications networks. *IEEE Trans. on Commun. Syst.* **12,** 1–9 (1964).
9. Barrett, S. D. & Kok, P. Efficient high-fidelity quantum computation using matter qubits and linear optics. *Phys. Rev. A* **71,** 060310 (2005).
10. Ben-Or, M., Crépeau, C., Gottesman, D., Hassidim, A. & Smith, A. Secure multiparty quantum computation with (only) a strict honest majority. in *47th Annual IEEE Symposium on Foundations of Computer Science* (2006), 249–260.
11. Benjamin, S. C., Browne, D. E., Fitzsimons, J. & Morton, J. J. Brokered graph-state quantum computation. *New J. Phys.* **8,** 141 (2006).
12. Bennett, C. H., Brassard, G. & Mermin, N. D. Quantum cryptography without Bell's theorem. *Phys. Rev. Lett.* **68,** 557 (1992).
13. Bennett, C. H., DiVincenzo, D. P., Smolin, J. A. & Wootters, W. K. Mixed-state entanglement and quantum error correction. *Phys. Rev. A* **54,** 3824 (1996).
14. Bennett, C. H. *et al.* Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Phys. Rev. Lett.* **70,** 1895 (1993).
15. Bennett, C. H. *et al.* Purification of noisy entanglement and faithful teleportation via noisy channels. *Phys. Rev. Lett.* **76,** 722 (1996).
16. Bernien, H. *et al.* Heralded entanglement between solid-state qubits separated by three metres. *Nature* **497,** 86–90 (2013).

17. Bhatia, R. & Davis, C. A better bound on the variance. *The American Mathematical Monthly* **107,** 353–357 (2000).

18. Borregaard, J. *et al.* One-way quantum repeater based on near-deterministic photon-emitter interfaces. *Phys. Rev. X* **10,** 021071 (2020).

19. Bradley, C. E. *et al.* A ten-qubit solid-state spin register with quantum memory up to one minute. *Phys. Rev. X* **9,** 031045 (2019).

20. Brassard, G., Broadbent, A. & Tapp, A. Quantum pseudo-telepathy. *Found. Phys.* **35,** 1877–1907 (2005).

21. Bratzik, S., Abruzzo, S., Kampermann, H. & Bruß, D. Quantum repeaters and quantum key distribution: The impact of entanglement distillation on the secret key rate. *Phys. Rev. A* **87,** 062335 (2013).

22. Briegel, H.-J., Dür, W., Cirac, J. I. & Zoller, P. Quantum repeaters: the role of imperfect local operations in quantum communication. *Phys. Rev. Lett.* **81,** 5932 (1998).

23. Briegel, H. J., Browne, D. E., Dür, W., Raussendorf, R. & Van den Nest, M. Measurement-based quantum computation. *Nat. Phys.* **5,** 19–26 (2009).

24. Brito, S., Canabarro, A., Cavalcanti, D. & Chaves, R. Satellite-based photonic quantum networks are small-world. *PRX Quantum* **2,** 010304 (2021).

25. Brito, S., Canabarro, A., Chaves, R. & Cavalcanti, D. Statistical properties of the quantum internet. *Phys. Rev. Lett.* **124,** 210501 (2020).

26. Broadbent, A., Fitzsimons, J. & Kashefi, E. Universal blind quantum computation. in *50th Annual IEEE Symposium on Foundations of Computer Science* (2009), 517–526.

27. Broadbent, A. & Tapp, A. Can quantum mechanics help distributed computing? *ACM SIGACT News* **39,** 67–76 (2008).

28. Broomell, G. & Heath, J. R. Classification categories and historical development of circuit switching topologies. *ACM Comput. Surv.* **15,** 95–133 (1983).

29. Bugalho, L., Coutinho, B. C., Monteiro, F. A. & Omar, Y. Distributing multipartite entanglement over noisy quantum networks. *Quantum* **7,** 920 (2023).

30. Cai, Z. & Benjamin, S. C. Constructing smaller pauli twirling sets for arbitrary error channels. *Sci. Rep.* **9,** 11281 (2019).

31. Calero Mas, H. Scalable and Implementable Entanglement Distribution Policies in Homogeneous Repeater Chains with Cutoffs [Master's thesis, Delft University of Technology] (2024).

32. Calsamiglia, J. & Lütkenhaus, N. Maximum efficiency of a linear-optical Bell-state analyzer. *Appl. Phys. B* **72,** 67–71 (2001).

33. Campbell, E. T. & Benjamin, S. C. Measurement-based entanglement under conditions of extreme photon loss. *Phys. Rev. Lett.* **101,** 130502 (2008).

34. Cao, Y. *et al.* The evolution of quantum key distribution networks: On the road to the qinternet. *IEEE Commun. Surv. Tutor.* **24,** 839–894 (2022).

35. Chakraborty, K., Rozpedek, F., Dahlberg, A. & Wehner, S. Distributed Routing in a Quantum internet. *arXiv preprint arXiv:1907.11630* (2019).

36. Chandra, A., Dai, W. & Towsley, D. Scheduling Quantum Teleportation with Noisy Memories. in *Proc. IEEE Int. Conf. Quantum Comput. Eng. (QCE)* (2022), 437–446.

37. Chen, K. C. *et al.* Zero-added-loss entangled-photon multiplexing for ground-and space-based quantum networks. *Phys. Rev. Appl.* **19,** 054029 (2023).

38. Chirolli, L. & Burkard, G. Decoherence in solid-state qubits. *Adv. Phys.* **57,** 225–285 (2008).

39. Chitambar, E. & Gour, G. Quantum resource theories. *Rev. Mod. Phys.* **91,** 025001 (2019).

40. Choi, H., Davis, M. G., Iñesta, Á. G. & Englund, D. R. Scalable Quantum Networks: Congestion-Free Hierarchical Entanglement Routing with Error Correction. *arXiv preprint arXiv:2306.09216* (2023).

41. Choi, H., Pant, M., Guha, S. & Englund, D. Percolation-based architecture for cluster state creation using photon-mediated entanglement between atomic memories. *npj Quantum Inf.* **5,** 104 (2019).

42. Cicconetti, C., Conti, M. & Passarella, A. Request scheduling in quantum networks. *IEEE Trans. Quantum Eng.* **2,** 2–17 (2021).

43. Collins, O., Jenkins, S., Kuzmich, A. & Kennedy, T. Multiplexed memory-insensitive quantum repeaters. *Phys. Rev. Lett.* **98,** 060502 (2007).

44. Coopmans, T., Brand, S. & Elkouss, D. Improved analytical bounds on delivery times of long-distance entanglement. *Phys. Rev. A* **105,** 012608 (2022).

45. Coopmans, T. *et al.* Netsquid, a network simulator for quantum information using discrete events. *Commun. Phys.* **4,** 164 (2021).

46. Crépeau, C., Gottesman, D. & Smith, A. Secure multi-party quantum computation. in *Proc. 34th Annual ACM Symposium on Theory of Computing* (2002), 643–652.

47. Cuquet, M. & Calsamiglia, J. Entanglement percolation in quantum complex networks. *Phys. Rev. Lett.* **103,** 240503 (2009).

48. Cuquet, M. & Calsamiglia, J. Limited-path-length entanglement percolation in quantum complex networks. *Phys. Rev. A* **83,** 032319 (2011).

49. Davies, B., Beauchamp, T., Vardoyan, G. & Wehner, S. Tools for the analysis of quantum protocols requiring state generation within a time window. *IEEE Trans. Quantum Eng.* **5,** 1–20 (2024).

50. Davies, B., Iñesta, Á. G. & Wehner, S. Entanglement buffering with two quantum memories. *Quantum* **8,** 1458 (2024).

51. Davies, D. W., Price, W., Barber, D. & Solomonides, C. *Computer networks and their protocols* (John Wiley & Sons, Inc., 1979).

52. Dehaene, J., Van den Nest, M., De Moor, B. & Verstraete, F. Local permutations of products of Bell states and entanglement distillation. *Phys. Rev. A* **67,** 022310 (2003).

53. Deutsch, D. *et al.* Quantum privacy amplification and the security of quantum cryptography over noisy channels. *Phys. Rev. Lett.* **77,** 2818 (1996).

54. DiAdamo, S., Qi, B., Miller, G., Kompella, R. & Shabani, A. Packet switching in quantum networks: A path to the quantum Internet. *Phys. Rev. Res.* **4,** 043064 (2022).

55. Duan, L.-M., Lukin, M. D., Cirac, J. I. & Zoller, P. Long-distance quantum communication with atomic ensembles and linear optics. *Nature* **414,** 413–418 (2001).

56. Dür, W., Briegel, H.-J., Cirac, J. I. & Zoller, P. Quantum repeaters based on entanglement purification. *Phys. Rev. A* **59,** 169 (1999).

57. Dür, W. & Briegel, H. J. Entanglement purification and quantum error correction. *Rep. Prog. Phys.* **70,** 1381 (2007).

58. Dür, W., Hein, M., Cirac, J. I. & Briegel, H.-J. Standard forms of noisy quantum operations via depolarization. *Phys. Rev. A* **72,** 052326 (2005).

59. Ekert, A. K. Quantum cryptography based on Bell's theorem. *Phys. Rev. Lett.* **67,** 661 (1991).

60. Elsayed, K., KhudaBukhsh, W. R. & Rizk, A. On the Fidelity Distribution of Link-level Entanglements under Purification. *arXiv preprint arXiv:2310.18198* (2023).

61. England, D. G., Balaji, B. & Sussman, B. J. Quantum-enhanced standoff detection using correlated photon pairs. *Phys. Rev. A* **99,** 023828 (2019).

62. Ewert, F. & van Loock, P. 3/4-efficient Bell measurement with passive linear optics and unentangled ancillae. *Phys. Rev. Lett.* **113,** 140403 (2014).

63. Freund, J., Pirker, A. & Dür, W. A flexible quantum data bus. *Phys. Rev. Res.* **6,** 033267 (2024).

64. Gauthier, S., Vardoyan, G. & Wehner, S. A control architecture for entanglement generation switches in quantum networks. in *Proc. 1st Workshop on Quantum Networks and Distributed Quantum Computing* (2023), 38–44.

65. Gauthier, S., Vasantam, T. & Vardoyan, G. An on-demand resource allocation algorithm for a quantum network hub and its performance analysis. in *Proc. IEEE Int. Conf. Quantum Comput. Eng. (QCE)* **1** (2024), 1748–1759.

66. Ghaderibaneh, M., Gupta, H., Ramakrishnan, C. & Luo, E. Pre-distribution of entanglements in quantum networks. in *Proc. IEEE Int. Conf. Quantum Comput. Eng. (QCE)* (2022), 426–436.

67. Gold, M., Lin, J., Chitambar, E. & Goldschmidt, E. A. Heralded arbitrary graph states with inefficient quantum emitters. *arXiv preprint arXiv:2405.13263* (2024).

68. Goodenough, K. *et al.* Near-term n to k distillation protocols using graph codes. *arXiv preprint arXiv:2303.11465* (2023).

69. Gottesman, D. *Stabilizer codes and quantum error correction* (California Institute of Technology, 1997).

70. Gottesman, D. The Heisenberg representation of quantum computers. *arXiv preprint quant-ph/9807006* (1998).

71. Gottesman, D. Theory of fault-tolerant quantum computation. *Phys. Rev. A* **57,** 127 (1998).

72. Gottesman, D. & Chuang, I. L. Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. *Nature* **402,** 390–393 (1999).

73. Gottesman, D., Lo, H.-K., Lutkenhaus, N. & Preskill, J. Security of quantum key distribution with imperfect devices. in *IEEE Int. Symp. Inf. Theory (ISIT)* (2004), 136.

74. Grace, M. R., Gagatsos, C. N. & Guha, S. Entanglement-enhanced estimation of a parameter embedded in multiple phases. *Phys. Rev. Res.* **3,** 033114 (2021).

75. Grimbergen, J., Haldar, S., Iñesta, Á. G. & Wehner, S. Probabilistic Cutoffs in Homogeneous Quantum Repeater Chains. *In preparation.*

76. Grimmett, G. & Stirzaker, D. *Probability and random processes* (Oxford university press, 2020).

77. Gupta, V., Harchol-Balter, M., Dai, J. G. & Zwart, B. On the inapproximability of M/G/K: why two moments of job size distribution are not enough. *Queueing Systems* **64,** 5–48 (2010).

78. Haldar, S., Barge, P. J., Khatri, S. & Lee, H. Fast and reliable entanglement distribution with quantum repeaters: Principles for improving protocols using reinforcement learning. *Phys. Rev. Appl.* **21,** 024041 (2024).

79. Haldar, S. *et al.* Policies for multiplexed quantum repeaters: theory and practical performance analysis. *arXiv preprint arXiv:2401.13168* (2024).

80. Harney, C. & Pirandola, S. Analytical methods for high-rate global quantum networks. *PRX Quantum* **3,** 010349 (2022).

81. Hartmann, L., Kraus, B., Briegel, H.-J. & Dür, W. Role of memory errors in quantum repeaters. *Phys. Rev. A* **75,** 032310 (2007).

82. Harty, T. *et al.* High-fidelity preparation, gates, memory, and readout of a trapped-ion quantum bit. *Phys. Rev. Lett.* **113,** 220501 (2014).

83. Hensen, B. *et al.* Loophole-free Bell inequality violation using electron spins separated by 1.3 kilometres. *Nature* **526,** 682–686 (2015).

84. Hermans, S. *et al.* Entangling remote qubits using the single-photon protocol: an in-depth theoretical and experimental study. *New J. Phys.,* 013011 (2023).

85. Horodecki, M., Horodecki, P. & Horodecki, R. Separability of mixed states: necessary and sufficient conditions. *Phys. Lett. A* **223,** 1–8. ISSN: 0375-9601 (1996).

86. Horodecki, M., Horodecki, P. & Horodecki, R. General teleportation channel, singlet fraction, and quasidistillation. *Phys. Rev. A* **60,** 1888 (1999).

87. Humphreys, P. C. *et al.* Deterministic delivery of remote entanglement on a quantum network. *Nature* **558,** 268–273 (2018).

88. Illiano, J., Caleffi, M., Manzalini, A. & Cacciapuoti, A. S. Quantum internet protocol stack: A comprehensive survey. *Computer Networks* **213,** 109092 (2022).

89. Iñesta, Á. G., Choi, H., Englund, D. & Wehner, S. Quantum circuit switching with one-way repeaters in star networks. in *Proc. IEEE Int. Conf. Quantum Comput. Eng. (QCE)* **1** (2024), 1857–1867.

90. Iñesta, Á. G., Davies, B., Kar, S. & Wehner, S. Entanglement buffering with multiple quantum memories. *arXiv preprint arXiv:2502.20240* (2025).

91. Iñesta, Á. G., Vardoyan, G., Scavuzzo, L. & Wehner, S. Optimal entanglement distribution policies in homogeneous repeater chains with cutoffs. *npj Quantum Inf.* **9,** 46 (2023).

92. Iñesta, Á. G., Vardoyan, G., Scavuzzo, L. & Wehner, S. *Data for "Optimal entanglement distribution policies in homogeneous repeater chains with cutoffs" [doi: 10.4121/20402037.v1],* 2022.

93. Iñesta, Á. G. & Wehner, S. *Data for "Performance metrics for the continuous distribution of entanglement in multi-user quantum networks" [doi: 10.4121/75ccbe86-76dd-4188-8c34-f6e012b1373a],* 2023.

94. Iñesta, Á. G. & Wehner, S. Performance metrics for the continuous distribution of entanglement in multiuser quantum networks. *Phys. Rev. A* **108,** 052615 (5 2023).

95. Jansen, S., Goodenough, K., de Bone, S., Gijswijt, D. & Elkouss, D. Enumerating all bilocal Clifford distillation protocols through symmetry reduction. *Quantum* **6,** 715 (2022).

96. Jiang, L., Taylor, J. M., Khaneja, N. & Lukin, M. D. Optimal approach to quantum communication using dynamic programming. *Proc. Natl. Acad. Sci.* **104,** 17291–17296 (2007).

97. Jones, C., Kim, D., Rakher, M. T., Kwiat, P. G. & Ladd, T. D. Design and analysis of communication protocols for quantum repeater networks. *New J. Phys.* **18,** 083015 (2016).

98. Jones, D. R., Schonlau, M. & Welch, W. J. Efficient global optimization of expensive black-box functions. *J. Glob. Optim.* **13,** 455–492 (1998).

99. Kaelbling, L. P., Littman, M. L. & Moore, A. W. Reinforcement learning: A survey. *J. Artif. Intell. Res.* **4,** 237–285 (1996).

100. Kalb, N. *et al.* Entanglement distillation between solid-state quantum network nodes. *Science* **356,** 928–932 (2017).

101. Khatri, S. Policies for elementary link generation in quantum networks. *arXiv preprint arXiv:2007.03193* **5** (2020).

102. Khatri, S. On the design and analysis of near-term quantum network protocols using Markov decision processes. *AVS Quantum Sci.* **4** (2022).

103. Khatri, S., Matyas, C. T., Siddiqui, A. U. & Dowling, J. P. Practical figures of merit and thresholds for entanglement distribution in quantum networks. *Phys. Rev. Res.* **1,** 023032 (2019).

104. Kirkpatrick, S., Gelatt Jr, C. D. & Vecchi, M. P. Optimization by simulated annealing. *Science* **220,** 671–680 (1983).

105. Kolar, A., Zang, A., Chung, J., Suchara, M. & Kettimuthu, R. Adaptive, Continuous Entanglement Generation for Quantum Networks. in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM)* (2022), 1–6.

106. Krastanov, S., Albert, V. V. & Jiang, L. Optimized entanglement purification. *Quantum* **3,** 123 (2019).

107. Kruszynska, C., Anders, S., Dür, W. & Briegel, H. J. Quantum communication cost of preparing multipartite entanglement. *Phys. Rev. A* **73,** 062328 (2006).

108. Krutyanskiy, V., Canteri, M., Meraner, M., Krcmarsky, V. & Lanyon, B. Multimode ion-photon entanglement over 101 kilometers. *PRX Quantum* **5,** 020308 (2024).

109. Krutyanskiy, V. *et al.* Entanglement of trapped-ion qubits separated by 230 meters. *Phys. Rev. Lett.* **130,** 050803 (2023).

110. Krutyanskiy, V. *et al.* Light-matter entanglement over 50 km of optical fibre. *npj Quantum Inf.* **5,** 72 (2019).

111. Kumar, V., Cicconetti, C., Conti, M. & Passarella, A. Routing in Quantum Repeater Networks with Mixed Noise Figures. *arXiv preprint arXiv:2310.08990* (2023).

112. Laflamme, R., Miquel, C., Paz, J. P. & Zurek, W. H. Perfect quantum error correcting code. *Phys. Rev. Lett.* **77,** 198 (1996).

113. Lee, A. & Longton, P. Queueing processes associated with airline passenger check-in. *J. Oper. Res. Soc.* **10,** 56–71 (1959).

114. Lee, Y., Bersin, E., Dahlberg, A., Wehner, S. & Englund, D. A quantum router architecture for high-fidelity entanglement flows in quantum networks. *npj Quantum Inf.* **8,** 75 (2022).

115. Leichtle, D., Music, L., Kashefi, E. & Ollivier, H. Verifying BQP Computations on Noisy Devices with Minimal Overhead. *PRX Quantum* **2,** 040302 (2021).

116. Li, B., Coopmans, T. & Elkouss, D. Efficient optimization of cut-offs in quantum repeater chains. in *Proc. IEEE Int. Conf. Quantum Comput. Eng. (QCE)* (2020), 158–168.

117. Li, L., Anand, P., He, K. & Englund, D. Dynamic Inhomogeneous Quantum Resource Scheduling with Reinforcement Learning. *arXiv preprint arXiv:2405.16380* (2024).

118. Li, L. *et al.* Coherent spin control of a nanocavity-enhanced qubit in diamond. *Nat. Commun.* **6,** 6173 (2015).

119. Liew, S. C. & Lee, T. T. *Principles of Broadband Switching and Networking* (John Wiley & Sons, 2010).

120. Lipinska, V., Murta, G., Ribeiro, J. & Wehner, S. Verifiable hybrid secret sharing with few qubits. *Phys. Rev. A* **101,** 032332 (2020).

121. Liu, J.-L. *et al.* A multinode quantum network over a metropolitan area. *arXiv preprint arXiv:2309.00221* (2023).

122. Liu, J.-L. *et al.* Creation of memory–memory entanglement in a metropolitan quantum network. *Nature* **629,** 579–585 (2024).

123. Lloyd, S. Enhanced sensitivity of photodetection via quantum illumination. *Science* **321,** 1463–1465 (2008).

124. Lo, H.-K., Curty, M. & Tamaki, K. Secure quantum key distribution. *Nat. Photonics* **8,** 595–604 (2014).

125. Makowski, A., Melamed, B. & Whitt, W. On averages seen by arrivals in discrete time. in *Proc. 28th IEEE Conf. Decis. Control* (1989), 1084–1086.

126. Marler, R. T. & Arora, J. S. Survey of multi-objective optimization methods for engineering. *Struct. Multidiscip. Optim.* **26,** 369–395 (2004).

127. Meignant, C., Markham, D. & Grosshans, F. Distributing graph states over arbitrary quantum networks. *Phys. Rev. A* **100,** 052333 (2019).

128. Meng, X., Gao, J. & Havlin, S. Concurrence Percolation in Quantum Networks. *arXiv preprint arXiv:2103.13985* (2021).

129. Mitov, K. V., Omey, E., Mitov, K. V. & Omey, E. *Renewal processes* (Springer, 2014).

130. Moehring, D. L. *et al.* Entanglement of single-atom quantum bits at a distance. *Nature* **449,** 68–71 (2007).

131. Mower, J. & Englund, D. Efficient generation of single and entangled photons on a silicon photonic integrated chip. *Phys. Rev. A* **84,** 052326 (2011).

132. Munro, W. J., Azuma, K., Tamaki, K. & Nemoto, K. Inside quantum repeaters. *IEEE J. Sel. Top. Quantum Electron.* **21,** 78–90 (2015).

133. Munro, W. J., Stephens, A. M., Devitt, S. J., Harrison, K. A. & Nemoto, K. Quantum communication without the necessity of quantum memories. *Nat. Photonics* **6,** 777–781 (2012).

134. Munro, W., Harrison, K., Stephens, A., Devitt, S. & Nemoto, K. From quantum multiplexing to high-performance quantum networking. *Nat. Photonics* **4,** 792–796 (2010).

135. Muralidharan, S., Kim, J., Lütkenhaus, N., Lukin, M. D. & Jiang, L. Ultrafast and fault-tolerant quantum communication across long distances. *Phys. Rev. Lett.* **112,** 250501 (2014).

136. Muralidharan, S. *et al.* Optimal architectures for long distance quantum communication. *Sci. Rep.* **6,** 1–10 (2016).

137. Nain, P., Vardoyan, G., Guha, S. & Towsley, D. On the analysis of a multipartite entanglement distribution switch. *Proc. ACM Meas. Analysis Comput. Syst.* **4,** 1–39 (2020).

138. Nielsen, M. A. & Chuang, I. *Quantum computation and quantum information* (American Association of Physics Teachers, 2002).

139. Niu, D., Zhang, Y., Shabani, A. & Shapourian, H. All-photonic one-way quantum repeaters with measurement-based error correction. *npj Quantum Inf.* **9,** 106 (2023).

140. Pant, M. *et al.* Routing entanglement in the quantum internet. *npj Quantum Inf.* **5,** 25 (2019).

141. Panteleev, P. & Kalachev, G. Degenerate quantum LDPC codes with good finite length performance. *Quantum* **5,** 585 (2021).

142. Patil, A., Pant, M., Englund, D., Towsley, D. & Guha, S. Entanglement generation in a quantum network at distance-independent rate. *npj Quantum Inf.* **8,** 51 (2022).

143. Peres, A. Separability criterion for density matrices. *Phys. Rev. Lett.* **77,** 1413 (1996).

144. Perseguers, S., Lewenstein, M., Acín, A. & Cirac, J. I. Quantum random networks. *Nat. Phys.* **6,** 539–543 (2010).

145. Pirker, A., Wallnöfer, J. & Dür, W. Modular architectures for quantum networks. *New J. Phys.* **20,** 053054 (2018).

146. Pirker, A. & Dür, W. A quantum network stack and protocols for reliable entanglement-based networks. *New J. Phys.* **21,** 033003 (2019).

147. Pompili, M. *et al.* Realization of a multinode quantum network of remote solid-state qubits. *Science* **372,** 259–264 (2021).

148. Pompili, M. *et al.* Experimental demonstration of entanglement delivery using a quantum network stack. *npj Quantum Inf.* **8,** 121 (2022).

149. Pouryousef, S., Panigrahy, N. K. & Towsley, D. A quantum overlay network for efficient entanglement distribution. in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM)* (2023), 1–10.

150. Praxmeyer, L. Reposition time in probabilistic imperfect memories. *arXiv preprint arXiv:1309.3407* (2013).

151. Prielinger, L., Iñesta, Á. G. & Vardoyan, G. Surrogate-guided optimization in quantum networks. *arXiv preprint arXiv:2407.17195* (2024).

152. Qian, K. *et al.* Heisenberg-scaling measurement protocol for analytic functions with quantum sensor networks. *Phys. Rev. A* **100,** 042304 (2019).

153. Rains, E. M. Rigorous treatment of distillable entanglement. *Phys. Rev. A* **60,** 173 (1999).

154. Rains, E. M. A semidefinite program for distillable entanglement. *IEEE Trans. Inf. Theory* **47,** 2921–2933 (2001).

155. Raussendorf, R. & Briegel, H. J. A one-way quantum computer. *Phys. Rev. Lett.* **86,** 5188 (2001).

156. Reiß, S. D. & van Loock, P. Deep reinforcement learning for key distribution based on quantum repeaters. *Phys. Rev. A* **108,** 012406 (2023).

157. Riera-Sàbat, F. & Dür, W. A modular entanglement-based quantum computer architecture. *arXiv preprint arXiv:2406.05735* (2024).

158. Rohatgi, A. *Webplotdigitizer 4.6 [https://automeris.io/WebPlotDigitizer],* 2022.

159. Rozpędek, F., Schiet, T., Elkouss, D., Doherty, A. C., Wehner, S., *et al.* Optimizing practical entanglement distillation. *Phys. Rev. A* **97,** 062333 (2018).

160. Rozpędek, F. *et al.* Parameter regimes for a single sequential quantum repeater. *Quantum Sci. Technol.* **3,** 034002 (2018).

161. Rozpędek, F. *et al.* Near-term quantum-repeater experiments with nitrogen-vacancy centers: Overcoming the limitations of direct transmission. *Phys. Rev. A* **99,** 052330 (2019).

162. Ruan, L., Kirby, B. T., Brodsky, M. & Win, M. Z. Efficient entanglement distillation for quantum channels with polarization mode dispersion. *Phys. Rev. A* **103,** 032425 (2021).

163. Ruf, M., Wan, N. H., Choi, H., Englund, D. & Hanson, R. Quantum networks based on color centers in diamond. *J. Appl. Phys.* **130,** 070901 (2021).

164. Sangouard, N., Simon, C., De Riedmatten, H. & Gisin, N. Quantum repeaters based on atomic ensembles and linear optics. *Rev. Mod. Phys.* **83,** 33 (2011).

165. Scarani, V. *et al.* The security of practical quantum key distribution. *Rev. Mod. Phys.* **81,** 1301–1350 (2009).

166. Schoute, E., Mancinska, L., Islam, T., Kerenidis, I. & Wehner, S. Shortcuts to quantum network routing. *arXiv preprint arXiv:1610.05238* (2016).

167. Shchukin, E., Schmidt, F. & van Loock, P. Waiting time in quantum repeaters with probabilistic entanglement swapping. *Phys. Rev. A* **100,** 032322 (2019).

168. Shchukin, E. & van Loock, P. Optimal entanglement swapping in quantum repeaters. *Phys. Rev. Lett.* **128,** 150502 (2022).

169. Sidhu, J. S. *et al.* Advances in space quantum communications. *IET Quantum Comm.* **2,** 182–217 (2021).

170. Sigman, K. & Wolff, R. W. A review of regenerative processes. *SIAM Rev.* **35,** 269–288 (1993).

171. Skrzypczyk, M. & Wehner, S. An Architecture for Meeting Quality-of-Service Requirements in Multi-User Quantum Networks. *arXiv preprint arXiv:2111.13124* (2021).

172. Slodička, L. *et al.* Atom-atom entanglement by single-photon detection. *Phys. Rev. Lett.* **110,** 083603 (2013).

173. Stephens, A. M., Huang, J., Nemoto, K. & Munro, W. J. Hybrid-system approach to fault-tolerant quantum communication. *Phys. Rev. A* **87,** 052333 (2013).

174. Stephenson, L. *et al.* High-rate, high-fidelity entanglement of qubits across an elementary quantum network. *Phys. Rev. Lett.* **124,** 110501 (2020).

175. Sutton, R. S. & Barto, A. G. *Reinforcement learning: An introduction* (MIT press, 2018).

176. Szepesvári, C. Algorithms for reinforcement learning. *Synth. Lect. Artif. Intell. Mach. Learn.* **4,** 1–103 (2010).

177. Talsma, L., Iñesta, Á. G. & Wehner, S. Continuously distributing entanglement in quantum networks with regular topologies. *Phys. Rev. A* **110,** 022429 (2024).

178. Thijssen, J. *Computational physics* (Cambridge university press, 2007).

179. Tillman, I., Vasantam, T., Towsley, D. & Seshadreesan, K. P. Calculating the capacity region of a quantum switch. in *Proc. IEEE Int. Conf. Quantum Comput. Eng. (QCE)* **1** (2024), 1868–1878.

180. Togan, E. *et al.* Quantum entanglement between an optical photon and a solid-state spin qubit. *Nature* **466,** 730–734 (2010).

181. Ursin, R. *et al.* Entanglement-based quantum communication over 144 km. *Nat. Phys.* **3,** 481–486 (2007).

182. Van Dam, S. B., Humphreys, P. C., Rozpędek, F., Wehner, S. & Hanson, R. Multiplexed entanglement generation over quantum networks using multi-qubit nodes. *Quantum Sci. Technol.* **2,** 034002 (2017).

183. Van Meter, R., Ladd, T. D., Munro, W. J. & Nemoto, K. System design for a long-line quantum repeater. *IEEE/ACM Trans. Netw.* **17,** 1002–1013 (2008).

184. Van Mieghem, P. *Performance analysis of complex networks and systems* (Cambridge University Press, 2014).

185. Vardoyan, G., Guha, S., Nain, P. & Towsley, D. On the capacity region of bipartite and tripartite entanglement switching. *ACM SIGMETRICS Perform. Eval. Rev.* **48,** 45–50 (2021).

186. Vardoyan, G., Guha, S., Nain, P. & Towsley, D. On the stochastic analysis of a quantum entanglement distribution switch. *IEEE Trans. Quantum Eng.* **2,** 1–16 (2021).

187. Vardoyan, G., Nain, P., Guha, S. & Towsley, D. On the capacity region of bipartite and tripartite entanglement switching. *ACM Trans. Model. Perform. Eval. Comput. Syst.* **8,** 1–18 (2023).

188. Victora, M., Krastanov, S., de la Cerda, A. S., Willis, S. & Narang, P. Purification and Entanglement Routing on Quantum Networks. *arXiv preprint arXiv:2011.11644* (2020).

189. Victora, M. *et al.* Entanglement purification on quantum networks. *Phys. Rev. Res.* **5,** 033171 (2023).

190. Vidal, G. & Werner, R. F. Computable measure of entanglement. *Phys. Rev. A* **65,** 032314 (2002).

191. Vinay, S. E. & Kok, P. Statistical analysis of quantum-entangled-network generation. *Phys. Rev. A* **99,** 042313 (2019).

192. Vlasiou, M. in *Wiley Encyclopedia of Operations Research and Management Science* (John Wiley & Sons, Ltd, 2011).

193. Wallnöfer, J., Hahn, F., Wiesner, F., Walk, N. & Eisert, J. Faithfully Simulating Near-Term Quantum Repeaters. *PRX Quantum* **5,** 010351 (2024).

194. Wehner, S., Elkouss, D. & Hanson, R. Quantum internet: A vision for the road ahead. *Science* **362,** eaam9288 (2018).

195. Welte, S., Hacker, B., Daiss, S., Ritter, S. & Rempe, G. Photon-mediated quantum gate between two neutral atoms in an optical cavity. *Phys. Rev. X* **8,** 011018 (2018).

196. Wengerowsky, S., Joshi, S. K., Steinlechner, F., Hübel, H. & Ursin, R. An entanglement-based wavelength-multiplexed quantum communication network. *Nature* **564,** 225–228 (2018).

197. Werner, R. F. Quantum states with Einstein-Podolsky-Rosen correlations admitting a hidden-variable model. *Phys. Rev. A* **40,** 4277 (1989).

198. Whitt, W. Approximations for the GI/G/m queue. *Prod. and Oper. Manage.* **2,** 114–161 (1993).

199. Wu, B.-H., Guha, S. & Zhuang, Q. Entanglement-assisted multi-aperture pulse-compression radar for angle resolving detection. *Quantum Sci. Technol.* **8,** 035016 (2023).

200. Wu, L. & Zhu, S. Entanglement percolation on a quantum internet with scale-free and clustering characters. *Phys. Rev. A* **84,** 052304 (2011).

201. Wu, X. *et al.* SeQUeNCe: a customizable discrete-event simulator of quantum networks. *Quantum Sci. Technol.* **6,** 045027 (2021).

202. Xia, Y. *et al.* Demonstration of a reconfigurable entangled radio-frequency photonic sensor network. *Phys. Rev. Lett.* **124,** 150502 (2020).

203. Xu, P., Wu, T. & Ye, L. The quantum correlations of the werner state under quantum decoherence. *Int. J. Theor. Phys.* **54,** 1958–1967 (2015).

204. Yan, H. *et al.* Entanglement purification and protection in a superconducting quantum network. *Phys. Rev. Lett.* **128,** 080504 (2022).

205. Yan, P.-S., Zhou, L., Zhong, W. & Sheng, Y.-B. Advances in quantum entanglement purification. *Sci. China Phys. Mech. Astron.* **66,** 250301 (2023).

206. Yoshino, K.-i., Ochi, T., Fujiwara, M., Sasaki, M. & Tajima, A. Maintenance-free operation of WDM quantum key distribution system through a field fiber over 30 days. *Opt. Express* **21,** 31395–31401 (2013).

207. Zhou, Y. *et al.* Long-lived quantum memory enabling atom-photon entanglement over 101 km of telecom fiber. *PRX Quantum* **5,** 020307 (2024).

208. Zhuang, Q. & Shapiro, J. H. Ultimate accuracy limit of quantum pulse-compression ranging. *Phys. Rev. Lett.* **128,** 010501 (2022).

209. Żukowski, M., Zeilinger, A., Horne, M. A. & Ekert, A. K. "Event-ready-detectors" Bell experiment via entanglement swapping. *Phys. Rev. Lett.* **71,** 4287 (1993).

ACKNOWLEDGEMENTS

This dissertation is the result of four years of work. During this time, I have been surrounded and supported by many wonderful people, and I would like to express my gratitude to all of them.

First, I would like to thank my supervisor, Prof. Stephanie Wehner, for giving me the amazing opportunity to conduct my thesis in her group. I am deeply thankful for your unwavering support, especially every time I decided to explore new research directions, as well as for all the resources you provided, from your encyclopedic knowledge and scientific vision to invaluable professional connections. Thanks for countless discussions, support, and critical feedback to all my collaborators and colleagues, as well as my master students – I hope you learned from me as much as I learned from you! Special thanks to my co-promotor, Prof. Ronald Hanson, and to Profs. Miriam Blaauboer, Eleni Diamanti, Rihan Hai, Lieven Vandersypen, Fernando Kuipers, and Gary Steele for kindly accepting the invitation to join the thesis committee and taking the time to provide critical feedback on this dissertation. And thanks to my wonderful paranymphs, Bethany and Yujia, for supporting me during my PhD journey, particularly on the week of my defense.

Lastly, my deepest gratitude and all my love to my friends and family, who gave me the strength to carry on. As I reflect on the incredible people who accompanied me throughout these years, I find myself revisiting the joyful moments they have so generously shared with me. So many amazing and ridículas atenvuras con los semolinos. El Café Latte. The board game nights with Celsi. Free coffees with birdwatching lessons. All the badminton competitions. The stories about fixing shuttles while growing apricots. Endless gossips at De Botanie. Los días de fiaca. Las noches de freefire. The game jams. The bubble teas. Evenings with Azul and *What we do in the shadows*. Deep talks over a scenic drive to Boston. Cozy visits to ceramics markets. Los treat-yo-self's. Las gambas nagasaki y demás tratos. Los almuerzos con aguacate y mango. Los Zeldas. Los sargos. Las cenas en Steki. NoveldaTech. And that contemporary art painting. Plus countless other memories destined to accompany me forever.

This PhD work has been financially supported by the Netherlands Organisation for Scientific Research (NWO/OCW), as part of the Frontiers of Nanoscience program. I would like to conclude with my gratitude towards the Casimir Research School for providing funding via the Frontiers of Nanoscience program.

Curriculum Vitæ

Álvaro G. Iñesta

2020-2025 **PhD in Quantum Networks**
Technische Universiteit Delft (Delft, the Netherlands)
Thesis: *Sharing Entanglement Efficiently: Protocols and Architectures for Quantum Networks*
Supervisors: Prof. dr. S.D.C. Wehner
Prof. dr. ir. R. Hanson

2018-2020 **MSc in Applied Physics (*cum laude*)**
Technische Universiteit Delft (Delft, the Netherlands)
Thesis: *Verifiable Hybrid Secret Sharing in the Presence of Noise*
Supervisor: Prof. dr. S.D.C. Wehner

2013-2018 **BSc in Engineering Physics & BSc in Aerospace Engineering**
CFIS, Universitat Politècnica de Catalunya (Barcelona, Spain)
Thesis: *Quantum Corrections in Nanoplasmonics and Deep Neural Network Training with Reverse Supervision*
Supervisors: Prof. dr. M. Soljačić (MIT, USA)
Prof. dr. J. Trull (UPC, Spain)

List of Publications

Included in this thesis

6. **Iñesta, Á. G.**, & Wehner, S. (2025). *The Viability of Preemptive Delivery of Quantum Resources.* In preparation.
   This manuscript is included in Chapter 5.

5. **Iñesta, Á. G.***, Davies, B.*, Kar, S., & Wehner, S. (2025). *Entanglement Buffering with Multiple Quantum Memories.* arXiv preprint arXiv:2502.20240.
   This manuscript is included in Chapter 7.

4. **Iñesta, Á. G.**, Choi, H., Englund, D., & Wehner, S. (2024). *Quantum Circuit Switching with One-Way Repeaters in Star Networks.* IEEE Int. Conf. Quantum Comput. Eng. (QCE24), p. 1857-1867. Full version available at arXiv:2405.19049.
   This article is included in Chapter 3.

3. Davies, B.*, **Iñesta, Á. G.*** & Wehner, S. (2023). *Entanglement Buffering with Two Quantum Memories.* Quantum, 8, 1458.
   This article is included in Chapter 6.

2. **Iñesta, Á. G.** & Wehner, S. (2023). *Performance Metrics for the Continuous Distribution of Entanglement in Multiuser Quantum Networks.* Phys. Rev. A, 108(5), 052615.
   This article is included in Chapter 4.

1. **Iñesta, Á. G.**, Vardoyan, G., Scavuzzo, L., & Wehner, S. (2023). *Optimal Entanglement Distribution Policies in Homogeneous Repeater Chains with Cutoffs.* npj Quantum Inf., 9(1), 46.
   This article is included in Chapter 2.

Not included in this thesis

4. Grimbergen, J., Haldar, S., **Iñesta, Á. G.**, & Wehner, S. (2024). *Probabilistic Cutoffs in Homogeneous Quantum Repeater Chains.* In preparation.

3. Prielinger, L., **Iñesta, Á. G.** & Vardoyan, G. (2024). *Surrogate-guided Optimization in Quantum Networks.* arXiv preprint arXiv:2407.17195.

2. Talsma, L., **Iñesta, Á. G.** & Wehner, S. (2024). *Continuously Distributing Entanglement in Quantum Networks with Regular Topologies.* Phys. Rev. A, 110(2), 022429.

1. Choi, H.*, Davis, M. G.*, **Iñesta, Á. G.**, & Englund, D. R. (2023). *Scalable Quantum Networks: Congestion-free Hierarchical Entanglement Routing with Error Correction.* arXiv preprint arXiv:2306.09216.

* These authors contributed equally.

TU Delft