Tools for the design of quantum repeater networks

Coopmans, T.J.

DOI
10.4233/uuid:90d06f1d-4f23-48cc-8f96-51500258020f

Publication date
2021

Document Version
Final published version

Citation (APA)
Coopmans, T. J. (2021). *Tools for the design of quantum repeater networks*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:90d06f1d-4f23-48cc-8f96-51500258020f

TOOLS FOR THE DESIGN OF QUANTUM REPEATER NETWORKS

TOOLS FOR THE DESIGN OF QUANTUM REPEATER 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 19 november 2021 om 12:30 uur

door

Tim COOPMANS

Master of Science in Logic,
Universiteit van Amsterdam, Nederland

Dit proefschrift is goedgekeurd door de promotoren.

Promotor: prof. dr. S.D.C. Wehner
Copromotor: dr. D. Elkouss Coronas

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof. dr. S.D.C. Wehner	Technische Universiteit Delft, promotor
Dr. D. Elkouss Coronas	Technische Universiteit Delft, copromotor

Onafhankelijke leden:

Prof. dr. ir. R. Hanson	Technische Universiteit Delft
Prof. dr. P. van Loock	Johannes Gutenberg Universität, Mainz
Prof. dr. N. Sangouard	CEA/Université Paris-Saclay
Prof. dr. B.M. Terhal	Technische Universiteit Delft
Dr. J. Borregaard	Technische Universiteit Delft

Footer images: a run of the symmetric NESTED-SWAP-ONLY scheme on 9 nodes (see sec. 3.3 and sec. 5.1.2). End nodes hold a single quantum memory (dot), intermediate nodes hold two. Connections between nodes either hold entanglement (solid line) or not (dashed line). A single page corresponds to a single timestep. Used parameters: $p_{\text{gen}} = 0.1$, $p_{\text{swap}} = 0.5$; an entanglement swap takes a single timestep.

ISBN 978-94-6384-269-3

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

If you want to go fast, go alone. If you want to go far, go together.

Old proverb

CONTENTS

SUMMARY

The Internet, a global network of communicating computers, has profoundly changed our lives, both in the way we work and relax. Quantum computers are a fundamentally new type of computer which brings certain computational tasks within reach, for example chemistry simulations for a reduction in global energy consumption. The Quantum Internet, the vision of a global network of quantum computers, combines these two, with applications such as secure quantum computing in the cloud.

A barrier to the realisation of a Quantum Internet is the loss of transmitted quantum information, usually encoded in particles of light. This fundamental limit can be overcome by splitting up the distance into segments and positioning so-called quantum repeaters in between. In principle, chains of quantum repeaters can extend the transmission range of quantum information to an arbitrarily long distance.

In this thesis, we consider the type of quantum repeater closest to experimental realisation, which is based on quantum memories for storing quantum information and probabilistically succeeding operations on them. Researchers have proposed a multitude of such quantum repeater schemes on the drawing board. We develop tools to analyse how these quantum repeater schemes will perform when implemented on real hardware suffering from time-dependent noise, in particular imperfect quantum memories for storing quantum information. Such time-dependent noise is often hard to capture, due to its complex interplay with the random time that these quantum repeater schemes need to finish. Our tools thus help to bridge the gap between theoretical proposals for quantum repeaters and the hardware components that are currently experimentally available. On the one hand, they enable optimisation over the design of quantum repeaters, while on the other hand they provide us with an indication of the hardware components whose improvement will pay off most to bring quantum repeaters to realisation.

This thesis consists of two parts. In the first part, we abstract away from many of the details of the hardware that quantum repeaters can be built of. In particular, we assume that a repeater has an unlimited number of memories for storing quantum information, and can perform any operation on the memories in parallel. We develop fast algorithms for characterising the time that a large class of quantum repeater schemes need to finish, as well as the quality of the quantum states they produce. We use one of the algorithms to investigate how much quantum repeater schemes benefit from discarding of quantum information after a maximum storage time. We optimise the storage time and find that the use of the optimal storage time lowers the hardware quality threshold necessary for quantum secure communication. Furthermore, we provide analytical bounds on the completion time of quantum repeater chains which in some cases improve exponentially upon existing work. We also prove that a commonly used approximation to the average completion time is in essence an upper bound, which renders existing feasibility analyses of quantum repeater schemes pessimistic.

In the second part, we introduce the quantum network simulator NetSquid to investigate more detailed hardware models. We simulate a quantum repeater chain of nitrogen vacancy centres (NV) in diamond, a promising hardware platform for building quantum networks. Since an NV centre cannot perform multiple operations in parallel, this required us to adapt the existing protocols. We use our simulation to show how much better the various parts of the NV setup should become to meet various performance targets. We also simulate a quantum switch, which can be thought of as a quantum repeater serving many users, with an abstract hardware model with a limited number of memories.

SAMENVATTING

Het Internet, een wereldwijd netwerk van communicerende computers, heeft ons leven grondig veranderd, zowel qua werk als ontspanning. Kwantumcomputers zijn een fundamenteel nieuw type computer die bepaalde rekentaken binnen handbereik brengen, zoals bijvoorbeeld simulaties van scheikundige processen die kunnen leiden tot een vermindering van het wereldwijd energieverbruik. Het Kwantuminternet, de visie van een wereldwijd netwerk van kwantumcomputers, combineert deze twee, met toepassingen zoals veilige kwantumberekeningen *in the cloud*.

Een hindernis voor de verwezenlijking van het Kwantuminternet is het verlies van kwantuminformatie wanneer deze verstuurd wordt, meestal opgeslagen in lichtdeeltjes. Deze fundamentele barrière kan overkomen worden door de afstand op te breken in segmenten en zogenaamde *kwantum repeaters* ertussen te plaatsen. In principe kunnen ketens van kwantum repeaters de afstand waarover kwantuminformatie verstuurd kan worden willekeurig ver verlengen.

In dit proefschrift bestuderen we de kwantum repeater typen die het dichtst bij verwezenlijking zijn. Deze zijn gebaseerd op het gebruik van kwantumgeheugen voor de opslag van kwantuminformatie en aanpassingen (operaties) van dit geheugen die met een zekere kans wél of niet slagen. Onderzoekers hebben een breed scala van zulke kwantum repeater ontwerpen op de figuurlijke tekentafel bedacht. In dit proefschrift ontwikkelen we theoretische instrumenten (*tools*) om te analyseren hoe goed zulke kwantum repeater ontwerpen gaan werken als ze geïmplementeerd zijn op echte hardware die, in het bijzonder, onderhevig is aan tijdsafhankelijke ruis. Onze instrumenten helpen ons zo om het gat te dichten tussen theoretische kwantum repeater ontwerpen en de hardware componenten die op dit moment beschikbaar zijn. Aan de ene kant staan de instrumenten ons toe om kwantum repeater ontwerpen te optimaliseren, terwijl ze ons aan de andere kant een indicatie geven welke hardware componenten we zouden moeten verbeteren om kwantum repeaters zo snel mogelijk te verwezenlijken.

Dit proefschrift bestaat uit twee delen. In het eerste deel abstraheren we veel details van de kwantum repeater hardware weg. In het bijzonder nemen we aan dat een repeater een onbeperkt aantal kwantumgeheugenplaatsen heeft, en bovendien operaties op verschillende geheugenplaatsen tegelijkertijd uit kan voeren. We ontwikkelingen snelle computeralgoritmen (voor een gewone computer, niet een kwantumcomputer) om de tijd die een kwantum repeater nodig heeft om een verbinding tot stand te brengen (de voltooiingstijd), te karakteriseren. De algoritmen berekenen ook de kwaliteit van de verbinding. We gebruiken één van de algoritmen om uit te zoeken hoeveel kwantum repeater ontwerpen profiteren van het weggooien van kwantuminformatie na een gegeven opslagtijd. We optimaliseren de opslagtijd en zien dat het gebruik van de optimale opslagtijd de vereiste hardware kwaliteit om kwantumcommunicatie te laten werken, naar beneden brengt. Daarnaast ontdekken we formules die de voltooiingstijd begrenzen; in sommige gevallen zijn deze formules een exponentiële verbetering ten opzichte

van bestaand werk. We bewijzen bovendien dat een vaak gebruikte benadering van de gemiddelde voltooiingstijd in essentie een bovengrens is, wat bestaande haalbaarheidsstudies pessimistisch maakt.

In het twee gedeelte introduceren we de kwantumnetwerksimulator NetSquid (een software pakket) voor het bestuderen van gedetailleerdere hardware modellen. We simuleren een keten van kwantum repeaters gebaseerd op het stikstofgatdefect (*nitrogen vacancy*) in diamant, een veelbelovend hardware platform voor de verwezenlijking van kwantumnetwerken. Omdat een stikstofgatdefect het niet mogelijk maakt om meerdere operaties tegelijkertijd uit te voeren, hebben we de bestaande kwantum repeater protocollen aangepast. We gebruiken onze computersimulaties om te laten zien hoeveel minder ruizig de verscheidene onderdelen van het stikstofgatdefect moeten worden om gegeven repeater doelstellingen te halen. We simuleren ook een kwantumschakelaar, die gezien kan worden als kwantum repeater die meer dan twee gebruikers dient, met een abstract hardware model en een beperkt kwantumgeheugen.

List of Publications

10. L. Vinkhuijzen*, **T. Coopmans***, D. Elkouss, V. Dunjko and A. Laarman, *LIMDD: a decision diagram for simulation of quantum computing including stabilizer states.*
Preprint: arXiv:2108.00931 (2021)

9. **T. Coopmans**, S. Brand and D. Elkouss, *Improved analytical bounds on delivery times of long-distance entanglement,*
Accepted for publication in *Physical Review A.* Preprint: arXiv:2103.11454 (2021)

8. F. da Silva, A. Torres-Knoop, **T. Coopmans**, D. Maier, S. Wehner, *Optimizing Entanglement Generation and Distribution Using Genetic Algorithms,*
Quantum Science and Technology (2021)

7. **T.Coopmans***, R. Knegjens*, A. Dahlberg, D. Maier, L. Nijsten, J. de Oliveira Filho, M. Papendrecht, J. Rabbie, F. Rozpędek, M. Skrzypczyk, L. Wubben, W. de Jong, D. Podareanu, Ariana Torres-Knoop, D. Elkouss†, S. Wehner†, *NetSquid, a NETwork Simulator for QUantum Information using Discrete events,*
Nature Communications Physics (2021)

6. K. Azuma, S. Bäuml, **T. Coopmans**, D. Elkouss and B. Li, *Tools for quantum network design,*
AVS Quantum Science 3, 014101 (2021)

5. B. Li, **T. Coopmans** and D. Elkouss, *Efficient optimization of cut-offs in quantum repeater chains,*
IEEE Transactions on Quantum Engineering (2021)

4. S. Brand*, **T. Coopmans*** and D. Elkouss, *Efficient computation of the waiting time and fidelity in quantum repeater chains,*
IEEE Journal on Selected Areas in Communications 38, 619 (2020)

3. A. Dahlberg, M. Skrzypczyk, **T. Coopmans**, L. Wubben, F. Rozpędek, M. Pompili, A. Stolk, P. Pawełczak, R. Knegjens, J. de Oliveira Filho, R. Hanson, S. Wehner, *A Link Layer Protocol for Quantum Networks,*
Proceedings of the ACM Special Interest Group on Data Communication, SIGCOMM '19 (Association for Computing Machinery, New York, NY, USA, 2019), pp. 159–173

2. **T. Coopmans**, J. Kaniewski and C. Schaffner, *Robust self-testing of two-qubit states,*
Physical Review A 99, 052123 (2019)

1. B. Kwaadgras, T. Besseling, **T. Coopmans**, A. Kuijk, A. Imhof, A. van Blaaderen, M. Dijkstra and R. van Roij, *Orientation of a dielectric rod near a planar electrode,*
Physical Chemistry Chemical Physics 16, no. 41 (2014): 22575-22582

* These authors contributed equally.

† These authors jointly supervised this work.

PREFACE

Dear reader,

This thesis is the result of four years of PhD work. During that time, I have been immersed in an environment of highly talented people, who taught me the language of quantum networks. After so many hours in this community, I have become accustomed to a particular jargon and way of writing – maybe accustomed too much. For that reason, let me give a brief 'reading guide'.

First, most researchers do not read scientific articles like a novel: from front to back. Instead, they read the title and summary, then skim through the paper and look at the figures. If the article still looks interesting, then they read the introduction, and potentially also the conclusion. By doing so, the reader establishes a 'frame' in which the rest of the information of the article can be put (You could compare it to a wardrobe that contains many drawers: it is neater to first have the wardrobe and then fill it with clothes, then to first have a pile of clothes on the ground, after which you will build the wardrobe). With this frame, it is much easier to read the rest of the paper – this time from front to back.

I propose that you, as reader of this dissertation, do exactly the same, at least at the start: first read the title and the summary, then proceed to the introduction. At the end of the introduction I will give an overview of the different chapters, which hopefully gives you an idea of what you would like to read next. Like most theses in the hard sciences, the core of this dissertation consists of chapters which are modified versions of (pre)published articles. Although these were written for an expert audience, please do not feel held back to read them by reading title and summary, skim the figures, etc.

I believe the use of jargon can hardly ever be completely avoided. However, I have noticed that many around me are increasingly often using field-specific terminology in everyday conversations, often without noticing, and it would be naive to think I am any different. For that reason, I have tried to compile a list of words that might be useful to know when reading this thesis – already starting with the words in the title. If you find that the dissertation is completely incomprehensible after having read this list and the introductory chapters, then I welcome you to come over for a cup of tea (or something stronger) and a good chat. I am more than willing to explain this dissertation's content.

So here comes the list of words.

First of all, you will find that this dissertation is written in 'we' form (first person plural). Partially I decided to do so because this is common in the field. More importantly, a lot of the work in the core chapters of this dissertation was done in collaboration with others. For those chapters, it would therefore be not only impolite to say that 'I did such and such', but simply incorrect.

Next, the phrase *design of quantum networks* refers to the fact that there are many different ways of building a network of quantum computers, and we are trying to find

the best one. In particular, it does not mean that we are designing in the way that artists do.

A *model* is a representation of reality. In this thesis, we will treat models which are more or less idealised. That is, we abstract away from many details of reality; in this case, from many details of the physical hardware that networks of quantum computers are built of.

The word *analytical* in the context of this thesis refers to the use of formulas and mathematics, as opposed to using computer code to arrive at a *numerical* analysis. By *semi-analytical*, we mean a combined approach, where we use mathematics to arrive at a formula that we then further investigate using numerics.

The abbreviations *i.e.* and *e.g.*, or *id est* and *exempli gratia*, mean 'that is' and 'for example', respectively.

In the current Internet, many computers are connected to each other. In this context, we refer to a computer as a *node* in the network. We could also say that the Internet consists of many *remote parties*.

A *protocol* is, loosely speaking, a computer program which runs at a node and decides what information to send to another node, and to react to information the node receives.

By *experimental*, we refer to the meaning of the word in the phrase 'experimental physicist'. That is, a physicist which performs scientific experiments. If a quantum network has been 'experimentally realised', it has been actually built in real life.

The most important word is maybe *classical*, which we use to indicate the conventional counterparts of quantum technology: classical computers are the computers that are currently available that do not specifically make use of particular features of quantum physics. Similarly, we talk about classical networks, of which the current Internet is an example.

Although this list is far from complete, I hope it helps in reading this thesis a bit more easily. Let me repeat that I am more than willing to come talk to you to explain what I have been doing for the past four years.

Tim Coopmans
Delft, May 2021

1

INTRODUCTION

Today, many of us use communication technology on a daily basis. We call by phone, send text messages by chat or e-mail, read news websites, watch videos or digital television, navigate over satellite-based GPS, connect to audio devices through bluetooth, and so on. It is an understatement to say that the development of communication technology has had a profound impact on they way we live.

All these applications are about communication between electronic devices which are, effectively, computers. In the realm of computers, the world has recently seen a surge in development and interest in quantum computers: a fundamentally new type of computer, explicitly taking advantage of the laws of quantum physics. Quantum physics describes the behaviour of physical objects which are at least a billion times smaller than the thickness of a human hair, such as atoms and electrons. Quantum computers [1] can perform some computational tasks astronomically faster than their conventional counterparts [2]. In particular, they bring tasks into reach for which today's computers would need ten thousands years [3] or longer, with promising applications such as chemistry simulations for a reduction in energy consumption [4, 5] or fast algorithms for linear-algebra tasks [6].

Given, on the one hand, the benefits of communication between computing devices, and on the other hand the advantages of quantum computers, it is a natural question whether communication technology could also benefit from quantum physics.

You will not be surprised that the answer is 'yes'. The first example was shown in 1984, when two researchers in North America found that the use of quantum physics enables two people at a distance to communicate in such a secure way, that no-one could read their messages, not even in principle [7][1] (Most of current secure data transfer is based on the idea that a hacker should solve a mathematical problem which is *expected* to be hard to solve, but not *proven* to be so. For quantum communication, it is known that it

[1] Specifically, this *quantum key distribution* allows two remote parties to generate a random password that no-one else knows. With such a password, they can encode their messages so that no-one can read them if they were intercepted.

is not only hard, but really *impossible* to break the security if the hacker wanted to read the message[2]).

Since then, more potential applications of so-called *quantum communication* have been found, such as performing a secret computation on a quantum computer in the cloud [8] or very precise synchronisation of clocks [9]. Given the fact that many applications for present-day computers have emerged relatively recently, it is more than likely that there are many applications for quantum communication that we have not found yet.

1.1. THIS THESIS: QUANTUM REPEATERS

In this thesis, we are concerned with the fact that communication signals *weaken* over a distance. This happens when you are talking to someone who is physically close (you will have to shout beyond 50 meters, or beyond 10 centimetres in a dance club), but also when your mobile device is too far from another bluetooth-enabled device or WiFi hotspot to connect. For internet data transfer beyond tens of kilometres, which often occurs through sending light pulses through glass fibre, the signal also weakens. In all these cases, a solution to bridge the distance is to make use of a *repeater*, which is a device that reads the weakened signal, amplifies it, and sends it on.

The same signal weakening happens for quantum communication, where often light is transferred through glass fibre or free space. In the quantum case, however, reading and amplifying the information is not possible in general[3]. For this reason, the way a repeater works cannot be straightforwardly translated to the quantum case. Fortunately, throughout the years researchers have come up with various alternative proposals for a *quantum repeater*, as well as chains of quantum repeaters for covering large distances. Although experimental progress towards building quantum repeaters has been enormous, a setup of one or more quantum repeaters, bridging a distance that could not have been bridged without, has not been realised yet[4].

This thesis is about the type of quantum repeater chain that is, arguably, closest to experimental reach (see Chapter 3 for a description). Central to this thesis is the question: *How close?*

In the next chapters, we provide tools to analyse how the quantum repeater schemes would perform when implemented using hardware that can currently be built in experimentalists' laboratories. We will see that in many cases the "most naive" implementation does not work. In other words: there is a *gap* between the repeater design on paper and what can be built in real life. We try to bridge this gap by changing the repeater design on the one hand, while on the other hand identifying the key parts of the physical hardware that need to be improved to most easily experimentally realise a quantum repeater.

We are not the first to analyse repeater schemes in this way. The novelty of this thesis is our progress in capturing *how fast* quantum repeaters would be and how their delays are affected by *time-dependent* effects.

[2] Provided quantum physics is correct.

[3] This is a famous result, which states that quantum information cannot be 'cloned'.

[4] There is no crystal clear concensus in the quantum repeater research community on what kind of experiment would count as a 'realisation of a quantum repeater'. Similarly, on how close start-of-the-art experiments are to such a realisation. I would like to refrain from such discussions here.

Let us explain this in a bit more detail. The goal of a quantum repeater is to establish a 'connection' between two quantum devices which are separated by a distance. (This sentence is deliberately very vague. For now, you may think of a quantum repeater as a magical means to transfer information, although that is not an entirely correct description.) The quantum repeaters we investigate are not equally fast every time they are used: sometimes they establish a connection quickly, sometimes not, and our goal is to exactly characterise this behaviour. That is, determine the probability that it takes longer than a given time to complete. The delay of quantum repeaters influences how well it allows us to transmit information. Particularly so because quantum information cannot be stored for a long time, in the same way that fresh fruit goes bad after a while[5]. There are also effects that work in the opposite direction (the worse the repeater, the longer it takes to finish) and it is this interplay that we can capture using the tools that we present in this thesis.

Another novelty of this thesis is the fact that we capture more details of the hardware than before, and adjust the quantum repeater schemes to that. Arguably, this makes our predictions of how well quantum repeaters will perform closer to reality than before.

1.2. NOT INCLUDED IN THIS THESIS

In this thesis, we will present various algorithms (computer programs, in this case for conventional computers, not quantum computers) to predict how quantum repeaters will perform on real hardware, at varying levels of detail. In the scenarios we consider in this thesis, the bottleneck in the time that such computations take is the number of quantum repeaters, but not the mimicking (simulation) of the quantum operations that the quantum repeaters perform.

In contrast, simulating a quantum computation on a regular computer is generally not feasible for any practical application that would run on a single quantum computer[6]. For most quantum *communication* applications, however, the devices need only perform a limited set of operations on the quantum particles known as *Clifford operations*. Fortunately, Clifford operations *can* be simulated quickly [12, 13], which allows us to investigate large networks of quantum computers[7]. Part of the PhD project leading to this thesis was devoted to thinking about further speedups for Clifford-based simulation and about the possible states of quantum particles after performing Clifford operations on them, known as *stabiliser states*. This work has not been included in this thesis; we very briefly mention the results here.

First, we found new concise expressions of stabiliser states (namely, by noting that

[5] The timescales at which quantum information decays in memory are much shorter than for rotting fruit: on the order of a second for nitrogen-vacancy centres in diamond, a hardware type which we will study in the second part of this thesis [10, 11].

[6] By 'not feasible', we mean that the computer would take at least thousands of years to finish. The fact that conventional computers cannot perform tasks that are relevant for quantum computers is true by definition of 'practical': if we could perform them on a conventional computer within reasonable time, we would not need a quantum computer for them.

[7] We emphasise that this fast simulation only holds for a quantum computer positioned at *a single location*. In particular, the existence of fast simulation of a network of quantum computers on a (single) conventional computer does not disprove the advantage of quantum computers for communication, for which quantum computers are spatially separated.

their 'substates' are local-Pauli-equivalent). We have combined this idea with decision diagrams, which is an existing framework for reasoning about computer programs (specifically: for capturing boolean functions and operations on them). The result is a new type of decision diagram, which is strictly more powerful than the union of both stabiliser states and existing decision diagrams [14].

In addition, we have attempted to find fast algorithms for computing the overlap between two quantum states for specific kinds of stabiliser states. These ideas initiated the master thesis work of Matthijs Rijlaarsdam [15] (which, however, has taken a slightly different direction).

1.3. CHAPTER OVERVIEW

This thesis starts with a brief introduction to the mathematics of quantum computing (**Chapter 2**). This chapter assumes a background in linear algebra. Next, in **Chapter 3**, we introduce the central topic of this thesis, quantum repeaters, in more detail. Although the chapter content is quite technical and mostly aimed at those who have a background in quantum information[8], we believe it is fairly well readable to the non-expert. At least it can be almost completely understood without having read Chapter 2.

The body of the thesis consists of the Chapters 4 to 8. It has been divided into two parts. In the first part, we will abstract away from many of the details of the hardware that quantum repeaters consist of and thus arrive at an *abstract hardware model*. We review existing literature on the analysis of abstract models of quantum networks in **Chapter 4**. In **Chapter 5** we provide fast algorithms (computer programs) for characterising the completion time of quantum repeater schemes. In **Chapter 6** we improve the runtime of one of these algorithms further and use it to investigate how much quantum repeater schemes benefit from restarts. **Chapter 7** finishes the first part by providing analytical bounds (that is, formulas instead of numbers produced by a computer program) on the average time that repeater schemes need to finish, and on the probability that they need longer than some given time to do so.

In the second part, we incorporate more details of the hardware. This part consists of a single chapter, **Chapter 8**, where we introduce the software package NetSquid for simulating detailed hardware models. We use NetSquid to investigate how a chain of quantum repeaters would perform when built using nitrogen-vacancy centres in diamond, a promising hardware type for building quantum networks. We also simulate a quantum switch, which is a many-armed repeater. We finish with a conclusion in **Chapter 9**.

REFERENCES

[1] M. A. Nielsen and I. L. Chuang, *Quantum information and quantum computation,* Cambridge: Cambridge University Press **2**, 23 (2000).

[2] A. Montanaro, *Quantum algorithms: an overview,* npj Quantum Information **2**, 15023 (2016).

[8] Chapter 3 will contain three mathematical formulas. According to a law that is attributed to Stephen Hawking, this means that the potential audience for this chapter will be cut into half at least three times. Since an individual reader cannot be cut in half without losing the ability to read, the number of people who will read Chapter 3 is at least eight or larger (two times two times two). Which is precisely why I added the formulas.

[3] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, C. Gidney, M. Giustina, R. Graff, K. Guerin, S. Habegger, M. P. Harrigan, M. J. Hartmann, A. Ho, M. Hoffmann, T. Huang, T. S. Humble, S. V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, J. Kelly, P. V. Klimov, S. Knysh, A. Korotkov, F. Kostritsa, D. Landhuis, M. Lindmark, E. Lucero, D. Lyakh, S. Mandrà, J. R. McClean, M. McEwen, A. Megrant, X. Mi, K. Michielsen, M. Mohseni, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Y. Niu, E. Ostby, A. Petukhov, J. C. Platt, C. Quintana, E. G. Rieffel, P. Roushan, N. C. Rubin, D. Sank, K. J. Satzinger, V. Smelyanskiy, K. J. Sung, M. D. Trevithick, A. Vainsencher, B. Villalonga, T. White, Z. J. Yao, P. Yeh, A. Zalcman, H. Neven, and J. M. Martinis, *Quantum supremacy using a programmable superconducting processor,* Nature **574**, 505 (2019).

[4] M. Reiher, N. Wiebe, K. M. Svore, D. Wecker, and M. Troyer, *Elucidating reaction mechanisms on quantum computers,* Proceedings of the National Academy of Sciences **114**, 7555 (2017), https://www.pnas.org/content/114/29/7555.full.pdf .

[5] *What problems will we solve with a quantum computer?* `https://www.microsoft.com/en-us/research/blog/problems-will-solve-quantum-computer/`, accessed: 7 Oct. 2021.

[6] A. W. Harrow, A. Hassidim, and S. Lloyd, *Quantum algorithm for linear systems of equations,* Phys. Rev. Lett. **103**, 150502 (2009).

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

[8] A. M. Childs, *Secure assisted quantum computation,* Quantum Info. Comput. **5**, 456 (2005).

[9] R. Jozsa, D. S. Abrams, J. P. Dowling, and C. P. Williams, *Quantum clock synchronization based on shared prior entanglement,* Phys. Rev. Lett. **85**, 2010 (2000).

[10] M. H. Abobeih, J. Cramer, M. A. Bakker, N. Kalb, M. Markham, D. J. Twitchen, and T. H. Taminiau, *One-second coherence for a single electron spin coupled to a multi-qubit nuclear-spin environment,* Nature Communications **9**, 2552 (2018).

[11] C. E. Bradley, J. Randall, M. H. Abobeih, R. C. Berrevoets, M. J. Degen, M. A. Bakker, M. Markham, D. J. Twitchen, and T. H. Taminiau, *A ten-qubit solid-state spin register with quantum memory up to one minute,* Phys. Rev. X **9**, 031045 (2019).

[12] D. Gottesman, *The Heisenberg representation of quantum computers,* arXiv:quant-ph/9807006v1 (1998).

[13] S. Aaronson and D. Gottesman, *Improved simulation of stabilizer circuits,* Physical Review A **70**, 052328 (2004).

[14] L. Vinkhuijzen, T. Coopmans, D. Elkouss, V. Dunjko, and A. Laarman, *LIMDD: a decision diagram for simulation of quantum computing including stabilizer states,* arXiv:2108.00931 (2021).

[15] M. Rijlaarsdam, *Improvements of the classical simulation of quantum circuits: Using graph states with local cliffords (master thesis),* Delft University of Technology (2020).

2

QUANTUM COMPUTING IN A NUTSHELL

In this chapter, we first provide a very brief and very much incomplete introduction to the mathematical formalism of quantum computing (sec. 2.1-2.3), after which we describe a common model of the decay of quantum information when it is stored in memory. The first part is based on specific sections from the excellent textbook by Nielsen and Chuang [1] and in this part, we will omit references to this work for brevity. For a more thorough introduction to quantum computing than provided in this chapter, we refer to their book. Together with the next chapter, Chapter 3, this chapter forms the preliminaries to the core of this thesis, Chapters 4-8. These preliminaries should be sufficient to understand most of this thesis. For this chapter, we assume that the reader is familiar with basic linear algebra.

2.1. A QUANTUM BIT AND HOW TO OPERATE ON IT

Classical computing is based on the storage, reading out and modification of bits, which take the value 0 or 1. The unit of quantum information is the quantum bit or qubit, which one could view as a generalisation of the bit. The state of a qubit is described by a vector of two complex numbers α and β (a complex 2-vector),

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

with the constraint that $|\alpha|^2 + |\beta|^2 = 1$. Here, $|z|^2 = a^2 - b^2$ is the squared modulus of a complex number $z = a + b \cdot i$ where a and b are real numbers and i is the complex unit satisfying $i^2 = 1$. Usually, we write $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ so that the state of a qubit is written as $\alpha|0\rangle + \beta|1\rangle$. We retrieve the two possible values that a regular bit can have (zero or one) by setting $\alpha = 1$ and $\beta = 0$ (for the state $|0\rangle$), or $\alpha = 0$ and $\beta = 1$ (for the state $|1\rangle$). If neither α nor β is zero, then we say that the qubit is in *superposition* of $|0\rangle$ and $|1\rangle$.

The main operations on a single qubit, or single-qubit *gates*, are 2-by-2 unitary matrices, i.e. matrices U satisfying $U^\dagger \cdot U = \mathbb{1}_2$. Here, $\cdot$ denotes matrix multiplication, we use the symbol $(.)^\dagger$ to denote the adjoint (obtained by taking the complex conjugate $a+bi \mapsto a-bi$ of each matrix element following by transposing the matrix) and the matrix $\mathbb{1}_2$ is the identity matrix (i.e. the unique matrix which maps each complex 2-vector to itself) given by

$$\mathbb{1}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

A gate U maps a state $|\phi\rangle$ to $U|\phi\rangle$. Examples of single-qubit gates are the Pauli matrices, which are $\mathbb{1}_2$ and

$$\mathsf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \mathsf{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \mathsf{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

or the Hadamard gate

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{2.1}$$

As example *measurement* or *readout* of a single qubit, consider the *computational basis* given by $|0\rangle$ and $|1\rangle$. Measuring a quantum state $\alpha|0\rangle + \beta|1\rangle$ in this basis yields a random outcome; the outcome 0 occurs with probability $|\alpha|^2$ and the outcome 1 with probability $|\beta|^2$. After measuring outcome 0 (1), the state of the qubit will be $|0\rangle$ ($|1\rangle$). In general, if we say that we perform a projective measurement on a quantum state $|\phi\rangle$ in the orthonormal basis $\{|m_0\rangle, |m_1\rangle\}$, then

$$\Pr(\text{outcome } 0) = |\langle\phi|m_0\rangle|^2, \quad \Pr(\text{outcome } 1) = |\langle\phi|m_1\rangle|^2$$

where $\langle\phi|m\rangle$ is the inner product between states $|\phi\rangle$ and $|m\rangle$, given by the single entry of $\langle\phi|\cdot|m\rangle$ with $\langle\phi| = \big(|\phi\rangle\big)^\dagger$. Alternatively, we will say that we measured $|\phi\rangle$ in the M-basis, where M is the matrix $M = |m_0\rangle\langle m_0| - |m_1\rangle\langle m_1|$. Upon receiving outcome $x \in \{0,1\}$, the post-measurement state is $|m_x\rangle$.

In practice, we encounter scenarios such as: we do not know which quantum state our particle is in, but it is either $|0\rangle$ or $|1\rangle$, each occurring with probability $1/2$. For situations like this, it is convenient to use the *density matrix*, which in this example scenario is

$$\rho = \frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1|.$$

The density matrix is convenient if we are continuing to perform gates or measurement on the qubit, without having to individually track the two possible scenarios (i.e. that the qubit is either in state $|0\rangle$ or in state $|1\rangle$). In general, if we have a collection of states $|\phi_1\rangle, |\phi_2\rangle, \ldots$ occurring with probabilities $p_1, p_2, \ldots$, then the density matrix is defined as

$$\rho = p_1 \big|\phi_1\big\rangle\big\langle\phi_1\big| + p_2 \big|\phi_2\big\rangle\big\langle\phi_2\big| + \ldots. \tag{2.2}$$

If one of the probabilities equals 1 (and hence the others are all zero), then we call ρ a *pure state*, and a *mixed state* otherwise. If we apply a gate U to this qubit, the density matrix is updated as $\rho \mapsto U\rho U^\dagger$. Measuring ρ in the M-basis (where $M = |m_0\rangle\langle m_0| - |m_1\rangle\langle m_1|$ as above) yields binary outcome x with probability $\langle m_x|\rho|m_x\rangle$. Upon receiving outcome x, the post-measurement density matrix is the pure state $|m_x\rangle\langle m_x|$.

2.2. MULTIPLE QUANTUM BITS AND ENTANGLEMENT

The state of n qubits is described by a vector of 2^n complex entries, whose squared moduli sum up to 1. As example, consider two qubits, which are individually in the states $|\phi_1\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{i}{\sqrt{2}}|1\rangle$ and $|\phi_2\rangle = \frac{2}{\sqrt{5}|0\rangle} + \frac{1}{\sqrt{5}}|1\rangle$. Their joint state is given by

$$|\phi_1\rangle \otimes |\phi_2\rangle = \begin{pmatrix} \frac{1}{\sqrt{2}} \cdot \frac{2}{\sqrt{5}} \\ \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{5}} \\ \frac{i}{\sqrt{2}} \cdot \frac{2}{\sqrt{5}} \\ \frac{i}{\sqrt{2}} \cdot \frac{1}{\sqrt{5}} \end{pmatrix} = \begin{pmatrix} \frac{2}{\sqrt{10}} \\ \frac{1}{\sqrt{10}} \\ \frac{2i}{\sqrt{10}} \\ \frac{i}{\sqrt{10}} \end{pmatrix}.$$

Here, the symbol $\otimes$ denotes the tensor product. We will omit it whenever it is clear from the context we are dealing with a multi-qubit state; for example, instead of $|0\rangle \otimes |1\rangle$ we will write $|0\rangle|1\rangle$ or $|01\rangle$. We can repeat this reasoning to describe the joint state of more than 2 qubits. The definition of the single-qubit density matrix in eq. 2.2 carries over directly to a density matrix for multiple qubits.

A gate on n qubits is an 2^n-by-2^n unitary matrix. An example of a two-qubit gate is the controlled-X gate (also called CNOT)

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{2.3}$$

When the two qubits to which the CNOT is applied are computational-basis states, the CNOT 'reads' the value of the first qubit (control qubit), and flips the second qubit's (target qubit) value. For example, $\text{CNOT}|0\rangle \otimes |1\rangle = |0\rangle \otimes |1\rangle$ and $\text{CNOT}|1\rangle \otimes |1\rangle = |1\rangle \otimes |0\rangle$. Identical to the single-qubit case, an n-qubit gate U maps an n-qubit density matrix ρ to $U\rho U^\dagger$.

If we perform a projective measurement the first qubit of an n-qubit state ρ in the basis $\{|m_0\rangle, |m_1\rangle\}$, then $\Pr(\text{outcome x}) = \text{Tr}(S_x)$ for $x = 0, 1$, where Tr denotes the trace of a matrix, i.e. the sum of its diagonal elements, and

$$S_x = \left(|m_x\rangle\langle m_x| \otimes \underbrace{\mathbb{1}_2 \otimes \cdots \otimes \mathbb{1}_2}_{n-1 \text{ times}} \right) \rho \left(|m_x\rangle\langle m_x| \otimes \underbrace{\mathbb{1}_2 \otimes \cdots \otimes \mathbb{1}_2}_{n-1 \text{ times}} \right).$$

Upon receiving outcome x, the post measurement density matrix is $S_x / \Pr(\text{outcome x})$.

A fascinating feature of multiple qubits is that they can be *entangled*, that is, that their joint state *cannot* be described by simply giving the states of the individual qubits (in fact, if qubits are entangled, it does not even make sense to talk about their individual states). By definition, we call a two-qubit state *separable* if it can be written as eq. (2.2) where all $|\phi_\square\rangle$ for which $p_\square \neq 0$ can be written as tensor products of single-qubit states. A two-qubit state is called entangled otherwise. A prime example of two-qubit entangled

states are the Bell states

$$\begin{aligned} |\Phi^+\rangle &= (|00\rangle + |11\rangle)/\sqrt{2} \\ |\Psi^+\rangle &= (|01\rangle + |10\rangle)/\sqrt{2} \\ |\Phi^-\rangle &= (|00\rangle - |11\rangle)/\sqrt{2} \\ |\Psi^-\rangle &= (|01\rangle - |10\rangle)/\sqrt{2} \end{aligned}$$

which are named after John Bell, who showed that entanglement enables correlations between remote parties which is stronger than possible in classical systems [2]. They are also referred to as the 'Bell basis', since the four Bell states form a basis of the two-qubit vector space.

When we start with the product state $|00\rangle$, we can produce $|\Phi^+\rangle$ by first applying the Hadamard gate from eq. (2.1) to the first qubit, followed by a CNOT gate from eq. (2.3) where the control qubit is the first qubit. The other Bell states can be produced from $|\Phi^+\rangle$ by applying one of the Pauli matrices to one of the two qubits. By applying any combination of single-qubit gates to the Bell states, we obtain a class of states which refer to as maximally-entangled two-qubit states or EPR pairs, named after the authors (Einstein, Podolski and Rosen) of a famous thought experiment involving entanglement [3].

We can now *measure in the Bell basis* by performing the reverse operation: we start with a two-qubit state, then perform the CNOT, followed by the Hadamard, and measure both qubits (denote the outcomes as a and b). If we started with one of the four Bell states, then which of the four can be found from applying $X^a \cdot Z^b$ to one of the qubits of $|\Phi^+\rangle$.

2.3. STATE QUALITY

The gates and measurement as performed by real-life hardware are not perfect; thus, the state that is actually produced (denoted by ρ_{actual}) is often different from the state ρ_{ideal} that we aimed to produce. A common measure for comparing ρ_{actual} to ρ_{ideal} (on the same number of qubits) is the *fidelity*, which is defined as

$$F(\rho_{\text{actual}}, \rho_{\text{ideal}}) = \text{Tr}\left(\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}\right)^2.$$

If ρ_{ideal} is the pure state $|\phi_{\text{ideal}}\rangle\langle\phi_{\text{ideal}}|$, then the fidelity can also be written as

$$F(\rho_{\text{actual}}, |\phi_{\text{ideal}}\rangle\langle\phi_{\text{ideal}}|) = \langle\phi_{\text{ideal}}|\rho_{\text{actual}}|\phi_{\text{ideal}}\rangle. \tag{2.4}$$

The fidelity is a value between 0 and 1. The value 1 is only achieved if the two states ρ_{actual} and ρ_{ideal} are equal. If $|\phi_{\text{ideal}}\rangle$ is a Bell state, then it straightforward to show that its fidelity with a separable two-qubit state is never larger than $\frac{1}{2}$. Throughout the thesis we will often say 'fidelity' when it is clear from the context we mean the Bell-state fidelity, i.e. the fidelity with a perfect Bell state.

A common model for an imperfect quantum state makes use of the depolarising channel $\mathcal{N}_p^{\text{depol}}$, which maps an n-qubit quantum state ρ to

$$\mathcal{N}_p^{\text{depol}}(\rho) = (1-p)\rho + p\frac{\mathbb{1}_{2^n}}{2^n}$$

where p is the depolarising probability and $\mathbb{1}_{2^n}$ the identity operator on 2^n-length complex vectors. If ρ is a perfect Bell state, for example $|\Phi^+\rangle$, then we call the resulting state

$$w\left|\Phi^+\right\rangle\left\langle\Phi^+\right| + (1-w)\mathbb{1}_4/4$$

a Werner state [4] and refer to $w = 1 - p_{\text{depol}}$ as the Werner parameter. It is straightforward to compute, using eq. (2.4), that the value $w = 1$ corresponds to a perfect Bell state while any $w < \frac{1}{4}$ corresponds to a Bell-state fidelity of less than $\frac{1}{2}$, i.e. below the classical bound.

Another often-used single-qubit noise model is the Z-dephasing channel $\mathcal{N}_p^{\text{deph}}$ (or dephasing channel for short), which maps a single-qubit state ρ to

$$\mathcal{N}_p^{\text{deph}}(\rho) = (1-p)\rho + pZ\rho Z^\dagger \tag{2.5}$$

with p the dephasing probability.

Finally, we mention the amplitude-damping channel $\mathcal{N}_p^{\text{AD}}$, defined as

$$\mathcal{N}_p^{\text{AD}}(\rho) = E_0\rho E_0^\dagger + E_1\rho E_1^\dagger$$

where $E_0 = |0\rangle\langle 0| + \sqrt{1-p}\,|1\rangle\langle 1|$ and $E_1 = \sqrt{p}\,|0\rangle\langle 1|$, where $p \in [0,1]$ is the amplitude-damping parameter.

Above, we have given a brief introduction to quantum mechanics; for a more complete and elaborate introduction, we refer to [1].

2.4. IMPERFECT MEMORIES

Since the results presented in this thesis focus in particular on time-effects, we highlight a model for a particularly relevant source of time-dependent state decay: memory noise. That is, the decay of a single qubit's quantum state, when stored in a quantum memory such as the spin of a particle, due to the qubit's interaction with the environment. Such decay is for example caused by interactions between the quantum memory and surrounding spins, such as the nuclear spins surrounding the nitrogen-vacancy centre we will introduce in sec. 3.4, inhomogeneity in an external magnetic field that is applied or the exchange of energy with the environment, among others [1, 5]. In this thesis, we follow a common description of the time evolution of the stored quantum state as described by the two parameters T_1 and T_2 (see [5] for a more elaborate introduction). The longitudinal coherence time or *relaxation time* T_1 describes the rate at which our qubit reaches its thermal state. The transverse coherence time or *dephasing time* T_2 describes how fast the qubit loses coherence, i.e. how fast a proper superposition of $|0\rangle$ and $|1\rangle$ tends to a classical mixture of $|0\rangle\langle 0|$ and $|1\rangle\langle 1|$.

The description of coherence in terms of T_1 and T_2 as we give below relies on various assumptions [6], in particular Markovianity: the environment does not have memory, i.e. each interaction with our qubit is independent of any interactions in the past. Under these assumptions, consider a qubit (e.g. spin-$\frac{1}{2}$ particle) which was originally in the state

$$\rho_{\text{init}} = \begin{pmatrix} a & b \\ b^* & 1-a \end{pmatrix}$$

for a real number a and a complex number b. For a specific coupling of the spin to the environment (details in [7]), the spin will decohere after time t to

$$\rho(t) = \begin{pmatrix} (a-a_0)e^{-t/T_1} + a_0 & be^{-t/T_2} \\ b^* e^{-t/T_2} & (a-a_0)e^{-t/T_1} + 1 - a_0 \end{pmatrix} \tag{2.6}$$

where $a_0|0\rangle\langle 0| + (1-a_0)|1\rangle\langle 1|$ is the thermal state of the qubit for $a_0 = \exp(\omega/(k\tau))/\left[\exp(\omega/(k\tau)) + \exp(-\omega/(k\tau))\right]$ that depends on the environment's temperature τ and the energy gap ω between the $|0\rangle$ and $|1\rangle$ states [1, 7]. Given eq. (2.6), we see that T_1 and T_2 live up to their names: a low relaxation time T_1 yields faster convergence of the diagonal entries of our qubit's state to the thermal state, while a low dephasing time T_2 yields faster disappearance of the off-diagonal entries and so results in decoherence.

In order to connect T_1 and T_2 to the noise maps from sec. 2.3, we consider two scenarios. First, $a_0 = \frac{1}{2}$, which occurs if $\tau \to \infty$ or $\omega = 0$. In that case, the thermal state is $(|0\rangle\langle 0| + |1\rangle\langle 1|)/2$ and we can write eq. (2.6) as

$$\rho(t) = \mathcal{N}_q^{\text{depol}}\left(\mathcal{N}_p^{\text{deph}}(\rho_{\text{init}})\right) \tag{2.7}$$

where $q = 1 - e^{-t/T_1}$ is the depolarising probability and $p = \frac{1}{2}\left(1 - e^{-t/T_2}\right)$ is the dephasing probability ($\mathcal{N}^{\text{depol}}$ and $\mathcal{N}^{\text{deph}}$ commute, so they may also be swapped in eq. (2.7)).

Another scenario is $a_0 = 1$ which occurs when $\omega > 0$ and $\tau = 0$, i.e. the qubit's state will thermalise to the ground state $|0\rangle$. In this scenario, eq. (2.6) can be rewritten as

$$\rho(t) = \mathcal{N}_r^{\text{AD}}\left(\mathcal{N}_p^{\text{deph}}(\rho_{\text{init}})\right) \tag{2.8}$$

where $r = 1 - e^{-t/T_1}$ is the amplitude damping parameter and $p = \frac{1}{2}\left(1 - e^{-t/T_2'}\right)$ is the dephasing probability, where $\frac{1}{T_2'} = \frac{1}{T_2} - \frac{1}{2T_1}$. (Note that $\mathcal{N}^{\text{AD}}$ and $\mathcal{N}^{\text{deph}}$ commute, so they may also be swapped in eq. (2.8).)

Although in reality, the above picture does not always neatly capture the dynamics of a quantum memory, experiments have confirmed that the use of a relaxation time T_1 and a dephasing time T_2 as above describes many situations fairly well [5–7]. In practice, there are inhomogeneities in the applied external fields and microscopic variations. Such variations average to a much shorter observed dephasing time T_2^* [7], which can be prolonged using dynamical decoupling techniques [8].

We finish by emphasising the Markovianity assumption in the explanation above. If we model the decay of a single-qubit quantum state in memory by a quantum channel $\mathcal{N}_t$, where t is the time the qubit spent in memory, then the Markovian assumption implies that $\mathcal{N}_t(\mathcal{N}_{t'}(\rho)) = \mathcal{N}_{t+t'}(\rho)$ for all times $t, t' \geq 0$ and all single-qubit quantum states ρ. It is straightforward to verify that the two scenarios described above, where $\mathcal{N}_t = \mathcal{N}_q^{\text{depol}} \circ \mathcal{N}_p^{\text{deph}}$ and $\mathcal{N}_t = \mathcal{N}_r^{\text{AD}} \circ \mathcal{N}_p^{\text{deph}}$ with $p(t), q(t)$ and $r(t)$ as defined above, indeed satisfy this property. Because of this Markovian feature, we do not have to apply all noise at each time step in our analysis, but instead can 'save' all noise until the qubit is taken out of the memory, or when it is acted upon.

REFERENCES

[1] M. A. Nielsen and I. L. Chuang, *Quantum information and quantum computation,* Cambridge: Cambridge University Press **2**, 23 (2000).

[2] J. S. Bell, *On the Einstein Podolsky Rosen paradox,* Physics Physique Fizika **1**, 195 (1964).

[3] A. Einstein, B. Podolsky, and N. Rosen, *Can quantum-mechanical description of physical reality be considered complete?* Physical review **47**, 777 (1935).

[4] R. F. Werner, *Quantum states with Einstein-Podolsky-Rosen correlations admitting a hidden-variable model,* Phys. Rev. A **40**, 4277 (1989).

[5] I. Žutić, J. Fabian, and S. Das Sarma, *Spintronics: Fundamentals and applications,* Rev. Mod. Phys. **76**, 323 (2004).

[6] L. Chirolli and G. Burkard, *Decoherence in solid-state qubits,* Advances in Physics **57**, 225 (2008), https://doi.org/10.1080/00018730802218067 .

[7] X. Hu, R. de Sousa, and S. D. Sarma, *Decoherence and dephasing in spin-based solid state quantum computers,* in *Foundations Of Quantum Mechanics In The Light Of New Technology: ISQM—Tokyo'01* (World Scientific, 2002) pp. 3–11.

[8] L. Viola, E. Knill, and S. Lloyd, *Dynamical decoupling of open quantum systems,* Phys. Rev. Lett. **82**, 2417 (1999).

3

PRELIMINARIES: BUILDING A QUANTUM INTERNET, BASED ON QUANTUM REPEATERS

The Quantum Internet is the vision of a worldwide network for transmitting both quantum information as well as classical messages, enabling various applications that are impossible by the classical Internet alone[1, 2]. The most commonly mentioned application is secure communication through quantum key distribution (QKD), whose security (no one else but the two communicating parties know the content of the messages) is in principle guaranteed by the laws of quantum physics [3, 4]. The information carriers over long distances are single photons, which carry a single quantum bit of information, where the $|0\rangle$ / $|1\rangle$ states as for example encoded as presence/absence of the photon or as horizontal/vertical polarisation [5]. Although metropolitan-size quantum key distribution networks based upon this line of research already exist [6], this approach does not scale to a worldwide network due to photon loss in the transmission medium [5]. Typically, the medium is glass fibre, in which the photon loss increases exponentially with the fibre's length.

This problem is overcome by the use of quantum repeaters [7]. In classical networking, a repeater is a device which amplifies the light pulse that encodes a classical message [8]. Long distances can be bridged by dividing them into smaller segments and positioning repeaters in between. For quantum information, however, the amplification (i.e. reading information and transmitting it again) is prohibited by the no-cloning theorem [9] and therefore the quantum equivalent of a repeater works differently.

The original proposal for a quantum repeater [7] is based on the generation of entanglement between spatially-separated quantum devices which are relatively close together, followed by the connection of this short-distance entanglement into long-distance entanglement. The remote entanglement that is thus distributed by quantum repeaters also enables a host of other applications, including clock synchronisation [10], distributed sensing [11, 12], secure delegated quantum computing [13] and extending

the baseline of telescopes [14]. Moreover, entanglement is also needed for QKD in the ultimate cryptographic scenario, where one does not trust the devices that produce the quantum bits [4].

In this chapter, we will describe the basic working of a quantum repeater and explain the building blocks for the type of quantum repeaters which are closest to experimental reach. These building blocks can be assembled in different ways, each giving rise to a protocol which delivers entanglement between spatially-separated parties. We finish by describing the two hardware models we study in this thesis for assessing the various protocols that can be constructed from the building blocks.

3.1. HOW A QUANTUM REPEATER WORKS

In its simplest form, the original quantum repeater proposal [7] is based on quantum teleportation [15], which is the transmission of a single qubit at the cost of consuming an entangled pair of qubits (we call such a pair a 'link' from here on). The protocol starts with two parties, which we call Alice and Bob. Alice has a single qubit she wants to transmit to Bob; in addition, Alice and Bob share a link. The quantum teleportation protocol is now as follows: first, Alice performs a measurement in the Bell basis on the two qubits she holds. Next, she sends the measurement outcome as a classical message to Bob. Last, Bob performs a local quantum operation on his qubit; which operation that is, depends on the measurement outcome he received from Alice. The resulting situation is that Bob's qubit, which was originally part of the link with Alice, is now in precisely the same state as the qubit Alice originally wanted to send.

The wonderful feature of teleportation is that the state of Alice's qubit is also preserved *if it were part of an entangled state before starting the teleportation protocol*. In the simplest form, a quantum repeater scheme makes use of this feature by consuming one link to transmit one qubit of another link. More concretely, consider two parties, Carol and Bob, who would like to be entangled. For this they use a third node positioned precisely in between them. This node, which we will call Alice, is the quantum repeater. See fig. 3.1(a). Now, Carol first generates fresh entanglement with the repeater Alice (we will describe more in detail how this fresh entanglement generation can be done, in sec. 3.2.1). After that, Alice generates fresh entanglement with Bob. The last step is that Alice and Bob perform the quantum teleportation scheme on the qubits they hold (Bob holds a single qubit, while Alice holds two, one for each of the links Alice-Bob and Alice-Carol). The result is that Bob holds the qubit state that Alice originally held, which is now entangled with Carol.

When one performs the maths, one will find out that it is actually not needed that Alice and Carol establish their entanglement first. In fact, entanglement may be generated in parallel over both segments Alice-Carol and Alice-Bob, and the Bell-state measurement is performed whenever both entangled pairs of qubits have been generated.

The single-repeater design as described above can be extended to multiple repeaters [7], as depicted in fig. 3.1(b). This is done by dividing the distance between two parties (which we call 'end nodes') into many segments which are short enough for generating fresh entanglement (i.e. photon loss is at an acceptable level) and putting repeaters at the segments' edges. The repeaters perform Bell-state measurements on the two links they generate, one to the left and one to the right, and send the outcome of the end

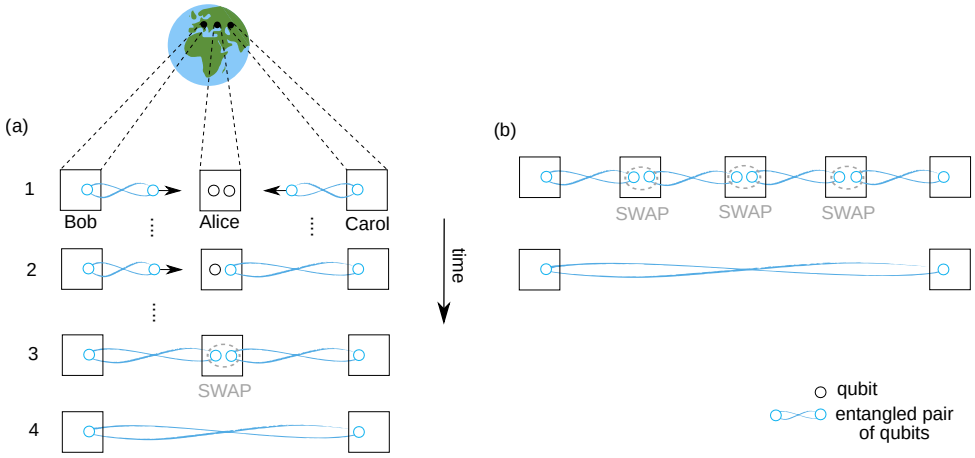


Figure 3.1: A simple quantum repeater scheme for establishing entanglement between spatially separated nodes. (a) A single repeater (Alice), positioned in between two nodes (Carol and Bob). First, (1) fresh entanglement is generated over the segments Carol-Alice and Alice-Bob in parallel. (2) Entanglement has been generated between Alice and Carol, but not yet between Alice and Bob. (3) As soon as entanglement has been established between both Carol-Alice and Alice-Bob, the repeater node Alice performs an entanglement swap (a Bell-state measurement) to connect the two entangled pairs of qubits, resulting into (4) entanglement between Carol and Bob. (b) A chain of quantum repeaters, which enables the distribution of entanglement over in-principle arbitrarily long distances. The figure of the globe is only included to illustrate that Alice, Bob and Charly are not at the same location; the distances between them are not necessarily realistic.

nodes. After all repeaters have performed their measurements, the end nodes will share entanglement; they only need to perform the outcome-dependent correction operation to correct their entangled state to a known form. In principle, chains of quantum repeaters could bridge arbitrarily large distances, and thus beat the photon-loss-limited distance that can be covered by direct entanglement generation [16, 17].

In reality, however, various sources of noise result in an degradation of the produced end-to-end entanglement when we add more repeaters [18]. For example, in current hardware, the Bell-state measurement and correction operations are not perfect. Neither are the memories which are used by a node to store qubits while the other necessary links are being generated: the longer the qubit is stored, the more its state will decay (see sec. 2.4). As a consequence of this noise, the entanglement between the end nodes of the repeater chain will be lost and instead we will end up with a state that is useless for quantum communication, for example below the tolerable error rates for QKD [19].

In the next section, we will see two ways to mitigate this noise. This first is entanglement distillation [20], a probabilistic conversion of two or more low-quality links into a single high-quality one. The other is a cut-off [21–32], where we discard entanglement which is supposedly of low quality, but then have to regenerate it. Entanglement distillation enables boosting state quality beyond the quality of fresh generated entanglement between adjacent nodes. In contrast, cut-offs, which mitigate memory noise, can only prevent degradation. Unfortunately, both measures yield significantly longer delivery times. Also, while distillation can boost state quality arbitrarily in the ideal case, it might

even decrease state quality if the local operations and memories are too imperfect. It is a priori therefore not clear how much distillation or cut-offs we should add, and how we should incorporate these into the protocol, to meet a prespecified entanglement delivery rate or state quality.

For this reason, we would like to explore various ways of building repeater protocols, in order to find one that meets our performance targets. In the remainder of this section, we will give a structured approach to constructing repeater protocols from the various operations we have discussed so far and finish with notes on how to analyse such protocols on real hardware.

This section, and the thesis in general, focuses on the so-called first generation quantum repeater protocols. These rely on quantum memories, which are currently imperfect, and moreover are limited in the rate at which end-to-end entanglement can be delivered due to two reasons: first, the fact that entanglement generation between adjacent nodes requires two-way messaging (see sec. 3.2.1 below). Second, the fact that operations on the thus produced entanglement, such as Bell-state measurements, are probabilistic. There also exist repeater schemes which make use of quantum error correction to avoid the use of probabilistic operations. In addition, some schemes employ photon-loss-tolerant means of establishing entanglement between adjacent nodes. Consequently, such schemes are no longer limited in rate. For these schemes, however, the local operations (gates and measurements) should be of very high quality [18], which are further out of reach of current hardware. In this thesis, we will not treat these later-generation repeaters. For an overview of the different repeater generations, we refer to the review paper [5].

3.2. BUILDING BLOCKS OF FIRST-GENERATION QUANTUM REPEATERS

Here, we describe four building blocks for constructing quantum repeater protocols for distributing entanglement over long-distances. We will also call the building blocks PROTOCOL-UNITS.

3.2.1. HERALDED GENERATION OF FRESH ENTANGLEMENT

By entanglement generation (GENERATE), we refer to the delivery of a fresh Bell state between two nodes in the network which are directly connected through a communication channel, such as an optical fibre. We refer to the generated entanglement as an 'elementary link'. There exist multiple schemes for the generation of elementary links [5], and they all rely on creating entanglement between a qubit, held by the node, together with a photon that the node emits. In sec. 8.6.4, we will describe a commonly used scheme [33, 34] where two nodes perform this local-entanglement generation, after which the two photons are transmitted to a station, positioned in between the nodes. At the station, the photon states interfere and proceed to two detectors. Depending on whether a photon is detected in each of the two detectors, entanglement between the two nodes' local qubits has been established. Whether the entanglement generation has succeeded, is communicated by sending a classical message to the two nodes. Note that entanglement generation is thus *heralded*: the nodes know whether they will have gener-

ated an elementary link, or whether they should try again. There exist more heralded-entanglement-generation schemes (see [5] for references), and in each of them, the generation is performed in discrete attempts until the first successful attempt[1] For each scheme, the attempt duration cannot exceed L/c, where L is the distance between the nodes and c speed of light in the photon transmission medium.

3.2.2. ENTANGLEMENT SWAPPING

The next building block is the Bell-state measurement at a quantum repeater, which converts two short-distance links into a single long-distance one[36]. We refer to this operation as an entanglement swap (SWAP). As explained above, it consists of a local quantum operation (including a measurement) which entangles the two remote qubits, together with the transmission of a classical message to both involved nodes to inform them of the measurement outcome, which determines the exact quantum state they hold. Depending on the used hardware, entanglement swaps either succeed with unit probability, such as the spins in the nitrogen-vacancy centre as introduced in sec. 3.4 [37, 38] or are probabilistic, for example when using atomic-ensemble memories where the entanglement swap is implemented using photon interference [39]. In case of failure, the short-distance links are lost (i.e. reduce to a separable state).

3.2.3. ENTANGLEMENT DISTILLATION

A quick calculation shows that if an entanglement swap is performed on two Werner states (see sec. 2.3) with Werner parameters w_A and w_B, then the resulting entanglement (after performing the correction operation) is a Werner state with parameter $w_A \cdot w_B$. Hence, if GENERATE produces Werner states with parameter w, then a chain of repeaters will produce a single long-distance Werner state with parameter w^M, where M is the number of swaps. Thus, the end-to-end state's Werner parameter, and hence its fidelity, decreases exponentially with the number of swaps and will soon drop below the classical bound (sec. 2.3), even if the swaps are implemented perfectly. As a consequence of this exponential decrease, the number of quantum repeaters that can be used, and thus the distance over which entanglement can be distributed, is limited.

This limit can be overcome by use of entanglement distillation (DISTILL) [40–42], which probabilistically converts two imperfect links (non-maximally entangled two-qubit states) into a single link of a higher quality (i.e. larger fidelity with the ideal Bell state)[2]. There exist several entanglement distillation schemes; we have depicted a commonly used one [40] in fig. 3.2. When the two input states are Werner states with parameters w_A and w_B, each larger than 1/2 but strictly less than 1, then this scheme outputs a Werner state with parameter (see sec. 5.7.1 for details of the derivation)

$$w_{\text{out}} = \frac{w_A + w_B + 4w_A w_B}{6p_{\text{success}}}$$

[1] In some scenarios the photon emission can be repeated already before the arrival of the heralding message, such as when local qubits can be measured directly after creating qubit-photon entanglement (e.g. in QKD) or in particular cases of multiplexing [35]. In this thesis, however, we will assume that a local qubit cannot be used until the heralding message for that qubit has arrived.

[2] There also exist entanglement distillation schemes which act on more than two links, but we will not consider those in this thesis.

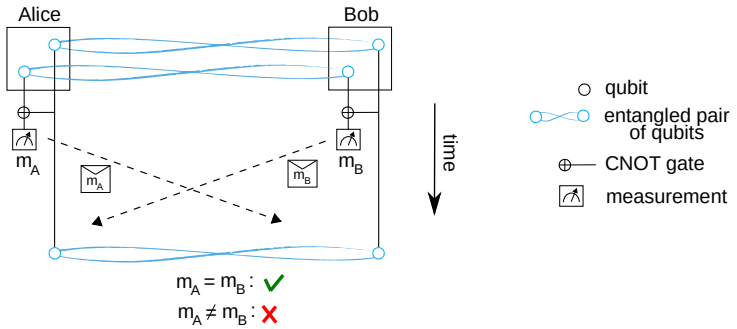


Figure 3.2: The entanglement distillation scheme from [40] for probabilistically converting two low-quality states into a single high-quality state. The figure depicts two nodes, Alice and Bob, who start out with with two entangled states. In the scheme, both Alice and Bob first perform a local gate and a measurement, with binary outcomes m_A and m_B. Next, Alice and Bob send a classical message to each other containing the measurement outcome. In case the measurement outcomes are the same, then we declare the distillation attempt a success and the resulting state is of higher quality than the states that Alice and Bob started out with.

with success probability

$$p_{\text{success}} = \frac{1 + w_A \cdot w_B}{2}.$$

Since w_{out} is strictly larger than w_A and w_B, we see that this scheme in principle enables boosting the quality of the state at the cost of having to produce two states as input.

3.2.4. DISCARDING ENTANGLEMENT: A CUT-OFF

Consider the scenario from fig. 3.1, where two end nodes first generate fresh entanglement (GENERATE) with a single repeater positioned in between them, after which the repeater performs an entanglement swap to establish entanglement between the two end nodes. Since the entanglement generation occurs in probabilistically succeeding attempts, it is likely that the two elementary links are not produced within the same attempt. Consequently, one of the two links needs to wait before generation of the other, and will be stored in a quantum memory in the meantime. If the quantum memory is imperfect, the link's quality will decrease; the longer it is stored, the lower its quality will be.

We thus see that the quality of a link is not only affected by entanglement swaps on imperfect links, but also by memory noise. Fortunately, the memory noise can be mitigated by discarding the link after it has been stored in memory for longer than some timeout time [21–32]. We refer to this discarding as a 'timeout cut-off' or, in case it is clear we mean a timeout, simply 'cut-off' (CUT-OFF). The disadvantage of a cut-off is, of course, that the link needs to be regenerated, and thus we can view the cutoff as a trade-off between the quality and the delivery duration of an end-to-end link.

3.3. HOW TO BUILD REPEATER PROTOCOLS FROM THESE BUILDING BLOCKS

Here, we describe how to construct protocols for chains of quantum repeaters from the building blocks described in sec. 3.2. We divide the possible protocols into two categories: `tree-shaped` and `non-tree-shaped`.

First, we introduce `tree-shaped` type protocols using fig. 3.3. Fig. 3.3(a) depicts the building blocks and fig. 3.3(b) shows how building blocks can be stacked together. In general, a `tree-shaped` protocol on a chain of nodes starts with one or multiple GEN blocks between each pair of adjacent nodes for fresh elementary link generation. A protocol then consists of stacking instances of the other three PROTOCOL-UNITs in such a way that the output link(s) of one are used as input link(s) to the other. The only restriction on how the PROTOCOL-UNITs can be stacked is that both output links of CUT-OFF are used as inputs for one DIST or SWAP block. As a consequence of the stacking, no pair of building blocks wait for the same links before proceeding. Hence, the resulting protocol has a tree structure. If a block at the root of a tree fails, then its input links are discarded and the GEN blocks at the tree's leaves will restart.

Fig. 3.3(c-e) show example `tree-shaped` protocols. Fig. 3.3(c) depicts the simplest single-repeater scheme we have described several times in this section already: entanglement is generated over two segments in parallel, and the repeater node performs an entanglement swap as soon as both elementary links have been generated. This protocol can be performed in a nested fashion on any repeater chain where the number of nodes is a power of 2, see fig. 3.3(c). We refer to this protocol as NESTED-SWAP-ONLY.

Next, in fig. 3.3(d) we depict a variant to the NESTED-SWAP-ONLY single-repeater protocol where the links that the entanglement swap acts upon, are not elementary links but are links which are the result of successful entanglement distillation on two elementary links. Of course, one could perform the entanglement distillation multiple times ($d > 1$) also: instead of with two elementary links, one then starts with 2^d elementary links, which are in a first nesting level distilled to 2^{d-1} links. This proceeds repeats over the d nesting levels until a single link is outputted. Fig. 3.3(e) shows the resulting scheme for $d = 2$.We will refer to these nested schemes as d-NESTED-WITH-DISTILL, or just as NESTED-WITH-DISTILL in case $d = 1$. We note that the NESTED-SWAP-ONLY and NESTED-WITH-DISTILL schemes were originally introduced by Briegel et al.[7, 43].

The NESTED-SWAP-ONLY and NESTED-WITH-DISTILL schemes, as they are described here, act on a number of segments which is a power of two. However, the building blocks can also be stacked in an asymmetric fashion. An example is the protocol depicted fig. 3.3(e). Considering asymmetric protocols might be advantageous for example when it is expected that part of the chain produces lower-quality links, so that we might want to apply more rounds of entanglement distillation. Lower-quality links are produced, for example, when the distances for GENERATE are not identical for the two segments. Since the success probability of GENERATE decreases with growing distance (due to increased photon loss), producing multiple elementary links in parallel over this segment implies longer waiting times for the first finished link, which subsequently decays in memory.

In the `tree-shaped`-type protocols, the events (swap, distill, cutoff) are performed in a predefined order. For example, in the NESTED-SWAP-ONLY scheme from fig. 3.3(d),

node C will only perform an entanglement swap once it holds links with A and E, i.e. when nodes B and D have swapped. Although convenient in the construction, it is not clear that this order will yield the most performant repeater protocols. As alternative, node C could swap whenever it holds entanglement with any node on its left and with any node on its right, regardless of whether B and D have performed their entanglement swap. Applying this logic to each node, where a node swaps as soon as possible (a.s.a.p.), gives rise to a `non-tree-shaped`-type protocol we call SWAP-ASAP.

3.4. TWO MODELS FOR IMPLEMENTING THE BUILDING BLOCKS IN HARDWARE

Above, we have described the construction of quantum repeater protocols from building blocks on a high-level. In order to study how these protocols perform on real hardware, in this thesis we consider two models for the hardware present at a node: an abstract model and a more detailed model based on the nitrogen-vacancy (NV) centre in diamond.

Abstract model. In the abstract model, a node is unrestricted in the type of protocol unit it wants to perform, and when it does so. To be precise, we assume

- a node has an unlimited number of quantum memories, as well as an unlimited number of connections (for transmission of photons and classical messages) to all other nodes[3];
- each quantum memory can be used both as storage qubit, as well as communication qubit (i.e. which can be used for heralded generation of fresh entanglement);
- a node can perform any quantum operation on any subset of its quantum memories;
- any two quantum operations, acting on disjoint sets of quantum memories, can be performed in parallel.

Nitrogen-vacancy (NV) model. The NV centre is a defect in the lattice structure of diamond, which consists of a nitrogen atom and an adjacent vacant site (for a thorough overview of NV centre technology, see [44, 45] and references therein). It forms an electronic spin-1 system, of which two levels are used as the qubit. We will refer to the electron spin-1 system simply as 'the electron'. The system can be excited by optical laser light, which produces a photon when it decays back; this property can be used for generating entanglement with a remote party. Single-qubit operations are performed either using optical laser light (measurement, initialisation) or microwave pulses (gates).

A small (roughly 1%) fraction of the carbon atoms in naturally-occurring diamond consists of carbon-13 isotopes. If such atoms are close to the electron, their nuclear spins exhibit a hyperfine coupling with the electron and these couplings can be addressed through microwave pulses, and thus quantum operations can be performed on

[3] We mean that if at some point during the protocol the node requires an idle memory, then we assume such a memory is present. This stands in contrast to the scenario where we would consider adding memories as 'free', because in that case, one could run an unlimited number of repeater chains in parallel, obtaining an infinitely large rate of delivering end-to-end entanglement.

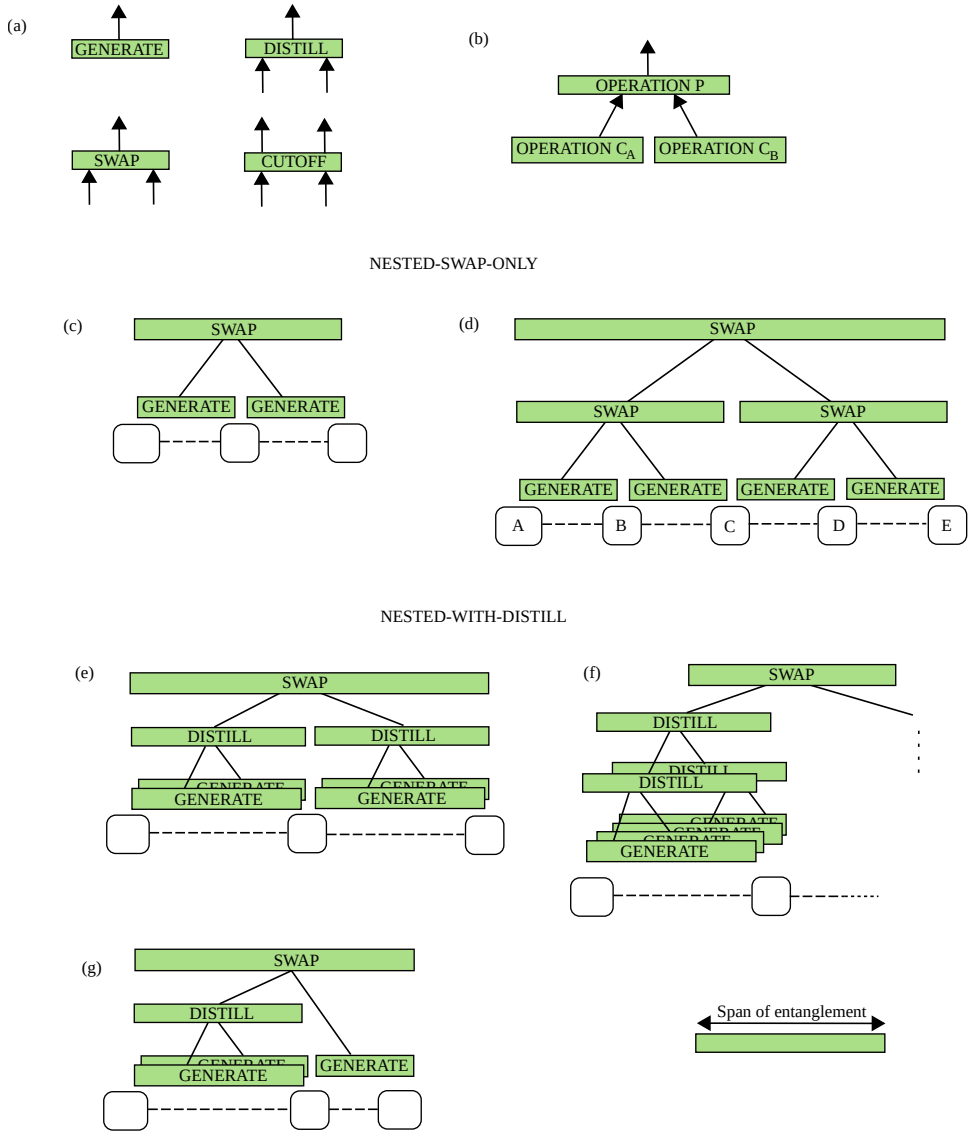


Figure 3.3: Examples of `tree-shaped`-type protocols on chains of quantum repeaters, constructed with the building blocks from sec. 3.2. (a) The four building blocks. The arrows depict the number of input and output links. (b) A building block P obtains its input links as the output of blocks C_A and C_B. If the operation P is probabilistic and fails, blocks C_A and C_B will have to regenerate the links. (c) The NESTED-SWAP-ONLY scheme for a single repeater. A run of this scheme is depicted in fig. 3.1(a). (d) the NESTED-SWAP-ONLY scheme can be nested to span a number of segments 2^n for some integer n; depicted is the case $n = 2$. (e-f) the NESTED-WITH-DISTILL scheme, which is the NESTED-SWAP-ONLY scheme with d nested rounds of entanglement distillation at each swap nesting level. The figure depicts the cases (e) $d = 1$ and (f) $d = 2$. (g) An example repeater protocol which is asymmetric.

the nuclear spins. Moreover, through the use of dynamical decoupling pulse sequences, the carbon nuclear spins can be decoupled from the electron spin; in addition to their longer coherence times, this makes the carbon nuclear spins good candidates for quantum memories.

Let us highlight some of the restrictions of a single NV centre that the abstract model does not have:

- a node has many carbon nuclear memories, but only a single electron spin;
- only the electron spin can be used as communication qubit;
- the operations that can be performed on the carbon nuclear spins are limited;
- only a single operation can be performed at a time.

Let us compare the two models. Experimental control of a single NV has been brought to a high level through years of development [45]. In particular, the NV has been shown to be able to implement all four protocol building blocks [23, 34, 37, 38], which makes it a promising candidate for quantum network hardware, with the realisation of a three-node network at laboratory scale as a recent highlight [37]. The limitations listed above, however, make the analysis and optimisation of long-distance quantum repeater protocols and networks complicated. For example, the fact that only a single operation can be performed at a time necessitates the scheduling of operations and protocol units [46]. Tracking all possible ways of scheduling operations, and their influence on the decay of entanglement stored in memory in the meantime (recall that the electron and carbon memories have finite coherence times) make analytical analysis of the NV model highly challenging. The analysis of a detailed model of NV centres for quantum networks, which takes all such aspects into account, has so far been done analytically only for a single repeater[24].

In contrast, the abstract model is more demanding to realise but is easier to study analytically. Also, although the abstract model does not apply to a node containing a single NV centre, it is a natural one in the context of multiple in-parallel-operating memories, where multi-qubit gates are performed probabilistically. An example of such a case are proposals for atomic-ensemble based quantum memories with linear-optical Bell-state measurements [39]. Also, the abstract-model assumptions can be made to hold for NV at the cost of requiring many NV centres per node, and the use of schemes for multi-NV quantum operations by consuming entanglement between them as resource (see for example the nonlocal controlled-Z operation in [47]), and thus require the invocation of GENERATE.

This thesis has been divided into two parts. In the first part, we will study the abstract model for all `tree-shaped`-type protocols. We will analytically investigate the delivery time and fidelity of such schemes, and develop fast algorithms for the evaluation of the resulting mathematical expressions when they become to complicated to track by hand. In the second part, we will introduce the discrete-event simulator NetSquid and use it to simulate the NV model for both the NESTED-WITH-DISTILL (a `tree-shaped`-type protocol) and the SWAP-ASAP scheme (a `non-tree-shaped`-type protocol), and compare their performance. Regarding the NESTED-WITH-DISTILL protocol, we will adjust the protocol

to accommodate the restrictions of the NV centre as outlined above. We will also investigate a quantum switch [48], which can be thought of as a quantum repeater for delivering multipartite entangled states, and simulate a `non-tree-shaped` type protocol with the abstract model with a limited number of memories.

We thus benefit from the advantages of both the abstract model (simpler, enables analysis of large networks) and the more detailed NV model (closer to physical reality) for the analysis and design of quantum networks.

REFERENCES

[1] H. J. Kimble, *The quantum internet,* Nature **453**, 1023 (2008).

[2] S. Wehner, D. Elkouss, and R. Hanson, *Quantum internet: A vision for the road ahead,* Science **362** (2018), 10.1126/science.aam9288, https://science.sciencemag.org/content/362/6412/eaam9288.full.pdf .

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

[4] A. K. Ekert, *Quantum cryptography based on Bell's theorem,* Phys. Rev. Lett. **67**, 661 (1991).

[5] W. J. Munro, K. Azuma, K. Tamaki, and K. Nemoto, *Inside quantum repeaters,* IEEE Journal of Selected Topics in Quantum Electronics **21**, 78 (2015).

[6] J. F. Dynes, A. Wonfor, W. W.-S. Tam, A. W. Sharpe, R. Takahashi, M. Lucamarini, A. Plews, Z. L. Yuan, A. R. Dixon, J. Cho, Y. Tanizawa, J.-P. Elbers, H. Greißer, I. H. White, R. V. Penty, and A. J. Shields, *Cambridge quantum network,* npj Quantum Information **5**, 101 (2019).

[7] H.-J. Briegel, W. Dür, J. I. Cirac, and P. Zoller, *Quantum repeaters: The role of imperfect local operations in quantum communication,* Phys. Rev. Lett. **81**, 5932 (1998).

[8] S. Hong, *Wireless: From Marconi's black-box to the audion* (MIT press, 2010).

[9] W. K. Wootters and W. H. Zurek, *A single quantum cannot be cloned,* Nature **299**, 802 (1982).

[10] R. Jozsa, D. S. Abrams, J. P. Dowling, and C. P. Williams, *Quantum clock synchronization based on shared prior entanglement,* Phys. Rev. Lett. **85**, 2010 (2000).

[11] W. Ge, K. Jacobs, Z. Eldredge, A. V. Gorshkov, and M. Foss-Feig, *Distributed quantum metrology with linear networks and separable inputs,* Phys. Rev. Lett. **121**, 043604 (2018).

[12] Q. Zhuang, Z. Zhang, and J. H. Shapiro, *Distributed quantum sensing using continuous-variable multipartite entanglement,* Phys. Rev. A **97**, 032329 (2018).

[13] A. M. Childs, *Secure assisted quantum computation,* Quantum Info. Comput. **5**, 456 (2005).

[14] A. Kellerer, *Quantum telescopes,* Astronomy & Geophysics **55**, 3.28 (2014), http://oup.prod.sis.lan/astrogeo/article-pdf/55/3/3.28/657336/10.1093_astrogeo_atu126.pdf .

[15] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, *Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels,* Phys. Rev. Lett. **70**, 1895 (1993).

[16] K. Azuma, A. Mizutani, and H.-K. Lo, *Fundamental rate-loss trade-off for the quantum internet,* Nature Communications **7**, 1 (2016).

[17] S. Pirandola, R. Laurenza, C. Ottaviani, and L. Banchi, *Fundamental limits of repeaterless quantum communications,* Nature Communications **8**, 1 (2017).

[18] S. Muralidharan, L. Li, J. Kim, N. Lütkenhaus, M. D. Lukin, and L. Jiang, *Optimal architectures for long distance quantum communication,* Scientific Reports **6**, 20463 EP (2016), article.

[19] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, *The security of practical quantum key distribution,* Rev. Mod. Phys. **81**, 1301 (2009).

[20] W. Dür, H.-J. Briegel, J. I. Cirac, and P. Zoller, *Quantum repeaters based on entanglement purification,* Phys. Rev. A **59**, 169 (1999).

[21] O. A. Collins, S. D. Jenkins, A. Kuzmich, and T. A. B. Kennedy, *Multiplexed memory-insensitive quantum repeaters,* Phys. Rev. Lett. **98**, 060502 (2007).

[22] L. Praxmeyer, *Reposition time in probabilistic imperfect memories,* arXiv preprint arXiv:1309.3407 (2013), arXiv:1309.3407 .

[23] N. Kalb, A. A. Reiserer, P. C. Humphreys, J. J. W. Bakermans, S. J. Kamerling, N. H. Nickerson, S. C. Benjamin, D. J. Twitchen, M. Markham, and R. Hanson, *Entanglement distillation between solid-state quantum network nodes,* Science **356**, 928 (2017).

[24] F. Rozpędek, R. Yehia, K. Goodenough, M. Ruf, P. C. Humphreys, R. Hanson, S. Wehner, and D. Elkouss, *Near-term quantum-repeater experiments with nitrogen-vacancy centers: Overcoming the limitations of direct transmission,* Phys. Rev. A **99**, 052330 (2019).

[25] F. Rozpędek, K. Goodenough, J. Ribeiro, N. Kalb, V. C. Vivoli, A. Reiserer, R. Hanson, S. Wehner, and D. Elkouss, *Parameter regimes for a single sequential quantum repeater,* Quantum Science and Technology (2018).

[26] S. Santra, L. Jiang, and V. S. Malinovsky, *Quantum repeater architecture with hierarchically optimized memory buffer times,* Quantum Science and Technology **4**, 025010 (2019).

[27] K. Chakraborty, F. Rozpędek, A. Dahlberg, and S. Wehner, *Distributed routing in a quantum internet,* arXiv:1907.11630 (2019), arXiv:1907.11630 .

[28] P. van Loock, W. Alt, C. Becher, O. Benson, H. Boche, C. Deppe, J. Eschner, S. Höfling, D. Meschede, P. Michler, F. Schmidt, and H. Weinfurter, *Extending quantum links: Modules for fiber- and memory-based quantum repeaters,* arXiv:1912.10123 (2019), arXiv:1912.10123 .

[29] F. Schmidt and P. van Loock, *Memory-assisted long-distance phase-matching quantum key distribution,* Phys. Rev. A **102**, 042614 (2020).

[30] S. Khatri, C. T. Matyas, A. U. Siddiqui, and J. P. Dowling, *Practical figures of merit and thresholds for entanglement distribution in quantum networks,* Phys. Rev. Research **1**, 023032 (2019).

[31] E. Shchukin, F. Schmidt, and P. van Loock, *Waiting time in quantum repeaters with probabilistic entanglement swapping,* Phys. Rev. A **100**, 032322 (2019).

[32] Y. Wu, J. Liu, and C. Simon, *Near-term performance of quantum repeaters with imperfect ensemble-based quantum memories,* Phys. Rev. A **101**, 042301 (2020).

[33] C. Cabrillo, J. I. Cirac, P. García-Fernández, and P. Zoller, *Creation of entangled states of distant atoms by interference,* Phys. Rev. A **59**, 1025 (1999).

[34] P. C. Humphreys, N. Kalb, J. P. J. Morits, R. N. Schouten, R. F. L. Vermeulen, D. J. Twitchen, M. Markham, and R. Hanson, *Deterministic delivery of remote entanglement on a quantum network,* Nature **558**, 268 (2018).

[35] S. B. van Dam, P. C. Humphreys, F. Rozpędek, S. Wehner, and R. Hanson, *Multiplexed entanglement generation over quantum networks using multi-qubit nodes,* Quantum Science and Technology **2**, 034002 (2017).

[36] M. Żukowski, A. Zeilinger, M. A. Horne, and A. K. Ekert, *"Event-ready-detectors" Bell experiment via entanglement swapping,* Phys. Rev. Lett. **71**, 4287 (1993).

[37] M. Pompili, S. L. N. Hermans, S. Baier, H. K. C. Beukers, P. C. Humphreys, R. N. Schouten, R. F. L. Vermeulen, M. J. Tiggelman, L. dos Santos Martins, B. Dirkse, S. Wehner, and R. Hanson, *Realization of a multinode quantum network of remote solid-state qubits,* Science **372**, 259 (2021), https://science.sciencemag.org/content/372/6539/259.full.pdf .

[38] W. Pfaff, B. J. Hensen, H. Bernien, S. B. van Dam, M. S. Blok, T. H. Taminiau, M. J. Tiggelman, R. N. Schouten, M. Markham, D. J. Twitchen, and R. Hanson, *Unconditional quantum teleportation between distant solid-state quantum bits,* Science **345**, 532 (2014), http://science.sciencemag.org/content/345/6196/532.full.pdf .

[39] N. Sangouard, C. Simon, H. de Riedmatten, and N. Gisin, *Quantum repeaters based on atomic ensembles and linear optics,* Rev. Mod. Phys. **83**, 33 (2011).

[40] C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, and W. K. Wootters, *Purification of noisy entanglement and faithful teleportation via noisy channels,* Phys. Rev. Lett. **76**, 722 (1996).

[41] D. Deutsch, A. Ekert, R. Jozsa, C. Macchiavello, S. Popescu, and A. Sanpera, *Quantum privacy amplification and the security of quantum cryptography over noisy channels,* Phys. Rev. Lett. **77**, 2818 (1996).

[42] W. Dür and H. Briegel, *Entanglement purification and quantum error correction,* Reports on Progress in Physics **70**, 1381 (2007).

[43] W. Dür, H.-J. Briegel, J. I. Cirac, and P. Zoller, *Quantum repeaters based on entanglement purification,* Phys. Rev. A **59**, 169 (1999).

[44] M. W. Doherty, N. B. Manson, P. Delaney, F. Jelezko, J. Wrachtrup, and L. C. Hollenberg, *The nitrogen-vacancy colour centre in diamond,* Physics Reports **528**, 1 (2013).

[45] D. D. Awschalom, R. Hanson, J. Wrachtrup, and B. B. Zhou, *Quantum technologies with optically interfaced solid-state spins,* Nature Photonics **12**, 516 (2018).

[46] A. Dahlberg, M. Skrzypczyk, T. Coopmans, L. Wubben, F. Rozpędek, M. Pompili, A. Stolk, P. Pawełczak, R. Knegjens, J. de Oliveira Filho, R. Hanson, and S. Wehner, *A link layer protocol for quantum networks,* in *Proceedings of the ACM Special Interest Group on Data Communication*, SIGCOMM '19 (Association for Computing Machinery, New York, NY, USA, 2019) pp. 159–173.

[47] N. Nickerson, *Practical fault-tolerant quantum computing (phd thesis),* (2015).

[48] G. Vardoyan, S. Guha, P. Nain, and D. Towsley, *On the stochastic analysis of a quantum entanglement switch,* SIGMETRICS Perform. Eval. Rev. **47**, 27 (2019).

I

ANALYSIS OF ABSTRACT MODELS OF QUANTUM NETWORKS

4

REVIEW OF EXISTING TOOLS FOR ASSESSING ABSTRACT QUANTUM NETWORKS

In this chapter, we review existing analytical and semi-analytical tools for assessing quantum networks using abstract models. Regarding quantum network simulators, which are focused on more detailed models of hardware, we give a brief overview in Chapter 8.

In this chapter, we review analytical tools for characterising the performance of quantum networks and the algorithms that immediately follow the analytical expressions. In particular, we consider the literature that studies the time it takes to distribute remote entanglement over a quantum network, referred to as the waiting time, and the quality of the entanglement. Due to their more modest quantum information processing requirements, we devote a large part of the chapter to quantum repeaters which are built from probabilistic schemes, i.e. the so-called first-generation repeater [1]. As a consequence of the probabilistic nature of such schemes, the waiting time is a random variable; thus, it is not represented by a single number but instead by a probability distribution. Our presentation focuses on the fidelity with respect to the desired maximally entangled state as a measure of entanglement quality. However, many of the tools can directly be used for estimating other figures of merit such as the secret key rate.

This chapter is organised as follows. We start in Section 4.1 with the modelling of a quantum network, which includes the mathematical abstraction of different components in a quantum network. In Section 4.2, we discuss the analytical tools used in evaluating the performance of networks. In some cases, those analytical tools yield closed-form expressions, of which the evaluation requires the assistance of numerical algorithms. We discuss three such cases in Section 4.3: Markov chain methods, probability-tracking algorithms, and sampling with Monte Carlo methods. Finally,we consider the

This chapter has been published, with minor changes, as part of: K. Azuma, S. Bäuml, T. Coopmans, D. Elkouss and B. Li, *Tools for quantum network design*, AVS Quantum Science 3, 014101 (2021)

analysis of quantum repeater protocols that include quantum error correction in Section 4.4. We have chosen to limit the scope of this chapter to discrete variable protocols.

4.1. ABSTRACT MODELS OF QUANTUM NETWORKS

Here, we build upon the introduction of network components in Section 3.2 and summarise common models for them, with an emphasis on how they contribute to the statistics of the waiting time and quality of the entangled state. Similar to Section 3.2, we refer in this section to a pair of entangled qubits shared by spatially-separated nodes, as a 'link'.

Entanglement generation. Recall that by entanglement generation, we refer to the production of a fresh Bell state (an 'elementary link') between two nodes in the network which are directly connected through a communication channel, such as an optical fibre. There are several schemes for the generation of elementary links [1], and in each of them, the generation is performed in discrete attempts until the first successful attempt. We assume that each attempt is of constant duration Δ and has constant success probability p_{gen}. The attempt duration Δ is determined by the distance and speed of light in the medium; in the rest of this section, we set $\Delta = 1$ for simplicity. It is also commonly assumed that the distinct attempts are independent and thus that the state ρ that is produced is constant, i.e. it is independent of the number of attempts required to produce it. The state ρ is a noisy Bell state which typically incorporates different sources of noise, photon loss, and detector inefficiency.

Entanglement swapping. Quantum repeaters overcome the fundamental distance limit over which elementary-link generation can be performed [2], see Chapter 3 Repeaters perform entanglement swaps to connect two short-distance links into a single long-distance one [3]. Typically, entanglement swaps are probabilistic, with a fixed success probability p_{swap} which is normally independent of the states swapped but depends on the physical implementation. For matter qubits that can be controlled directly, an entanglement swap can be implemented with deterministic quantum gates, i.e. $p_{\text{swap}} = 1$. If entanglement swapping is implemented with optical components, the entanglement swapping becomes probabilistic, i.e., $p_{\text{swap}} < 1$ and typically $p_{\text{swap}} \leq 0.5$ [4]. There are also more sophisticated optical swapping schemes with a probability larger than one half [5–7]. In some models, where the memory decoherence to the vacuum state is considered, the success probability can also be a variable [8].

Entanglement distillation. Entanglement distillation is the probabilistic conversion of multiple low-quality entangled pairs of qubits into a single one of high quality [9]. In contrast with entanglement swapping, the success probability p_{dist} depends on the entangled states that are distilled [9, 10].

Entangled state representation. Arguably, the simplest model of the fresh elementary link state is a Werner state [11], which characterises the state with a single parameter w:

$$\hat{\rho} = w\,|\Phi_2\rangle\langle\Phi_2| + (1-w)\hat{I}/4\,,$$

where $|\Phi_2\rangle$ is the desired maximally-entangled two-qubit state and $\hat{I}/4$ the maximally mixed state on two qubits. Although operations such as entanglement distillation do not always output a Werner state, any two-qubit state can be transformed into a Werner

state with LOCC without changing the fidelity [12]. A more general model is a probabilistic mixture of the four Bell states. This representation is convenient as it includes the resulting state after the application of random Pauli gates on a perfect Bell state. In principle, one could also track the full density matrix, though many studies choose the previous two representations to simplify the analysis. Given the density matrix $\hat{\rho}$ of a state, its fidelity with a pure target state $|\phi\rangle$ is given by $\langle\phi|\,\hat{\rho}\,|\phi\rangle$. Throughout the section, the target state will be a Bell state.

Noise modelling. Imperfections of the quantum devices, for example, operational noise and detector inefficiencies, are commonly modelled by depolarising, dephasing, or amplitude damping channels. The first two can be incorporated relatively simply into analytical derivations as they correspond to the random application of Pauli gates. Amplitude damping requires tracking the full density matrix. One could, however, replace an amplitude damping channel with the more pessimistic choice of a depolarisation channel, which does not change the output state's fidelity with the target state, or alternatively twirl the damped state by applying random Pauli operations [13].

Particularly relevant in the context of entanglement generation using probabilistic components is the noise caused by time-dependent memory decoherence: in case multiple links are needed, the earliest link is generally generated before the others are ready and thus needs to be stored in a quantum memory. The storage time leads to a decrease in the quality of the entanglement, and the longer the qubit is stored, the more its quality degrades. Due to the interplay between waiting time and time-dependent decay of entanglement quality, memory noise is particularly hard to capture. Sometimes this problem is sidestepped by analysing protocols with running time qualitatively shorter than the memory decoherence time.

Node model. For simplicity, the network nodes can be modelled by a fully-connected quantum information processing device capable of generating entanglement in parallel with its neighbours. However, it is important to note that many platforms do not conform to this model. For instance, NV-centres in a single diamond have a single optical interface. Hence, if nodes hold only a single NV centre, entanglement generation can only be attempted with one adjacent node at a time. Moreover, the connectivity between the qubits follows a star topology, i.e. direct two-qubit gates between arbitrary qubits are not possible.

Cut-off. Due to memory decoherence, the quality of the stored entanglement decreases as the waiting time grows. One common strategy to compensate for memory decoherence is cut-offs: if a link remains idle for too long, it is discarded. By discarding entanglement whose storage time exceeds some pre-specified threshold, one improves the quality of the delivered entanglement at the cost of longer waiting time.

Additionally, it is possible to build on top of this idea a simplified model of memory decoherence: the quantum information is preserved perfectly for a fixed duration and then lost [14, 15].

Note that the inclusion of cut-offs in entanglement distribution schemes complicates their analysis because of the additional effect of waiting time on the state quality.

Nested protocols. One particularly relevant network topology is the repeater chain, where all nodes are arranged in a line. Nested protocols offer a structured approach to distributing entanglement across a repeater chain [16–23]. In this section, unless explic-

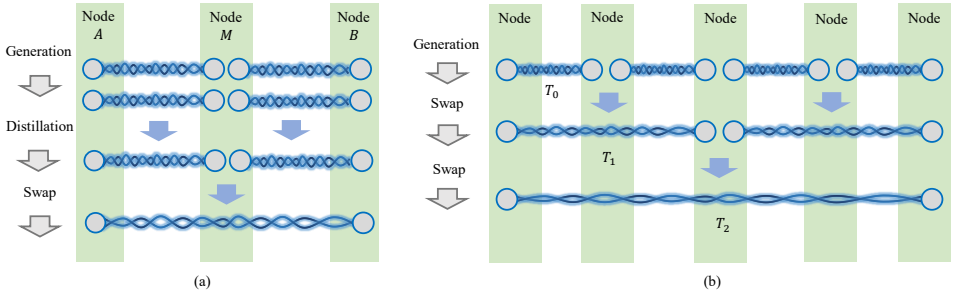


Figure 4.1: Two examples of a nested (`tree-shaped`-type) repeater protocol. **(a)** The NESTED-WITH-DISTILL protocol (Section 3.3) as example with $d = 1$ round of distillation and $n = 1$ nesting levels, on $2^n + 1 = 3$ nodes. Nodes A and M generate two entangled pairs in parallel, followed by performing entanglement distillation on the two pairs, and repeat this procedure until the distillation step has succeeded. Since the protocol is nested, nodes M and B do the same. Once both distillation steps on each side of M succeed, M performs an entanglement swap, which produces entanglement between A and B. If the entanglement swap fails, then the protocol restarts, i.e. A-M and M-B start with entanglement generation again. **(b)** The NESTED-SWAP-ONLY scheme (Section 3.3) as an example with $n = 2$ nesting levels (5 nodes) without entanglement distillation. At each nesting level, the distance that entanglement spans is doubled. By T_n, we denote the random variable describing the delivery time of entanglement at level n. In Section 4, we consider nested repeater protocols of entanglement generation and swapping such as in (b). Whenever the protocol includes entanglement distillation, as in (a), or cut-offs, it is mentioned explicitly.

4

itly mentioned, we follow the BDCZ scheme[16], i.e. the `tree-shaped`-type schemes as defined in Section 3, with the restriction that each entanglement swap doubles the distance that an entangled pair spans. In such a scheme, the number of repeater segments is 2^n ($2^n + 1$ nodes) where n is the number of nesting levels at which an entanglement swap is performed. We depict examples of BDCZ protocols in Fig. 4.1. We denote the waiting time random variable of a repeater scheme on 2^n segments as T_n. Many of the tools for determining the waiting time statistics and quality of the produced entanglement discussed below also apply to other schemes than nested repeater protocols.

4.2. ANALYTICAL STUDY OF THE WAITING TIME AND FIDELITY

In this section, we present analytical tools to compute the waiting time and the fidelity of the entangled state produced between the end nodes of a repeater chain.

We consider the nested repeater chain protocol on 2^n segments (see Section 4.1) with only entanglement swapping, i.e. no distillation or cut-offs unless explicitly mentioned. For simplicity, we assume that the generation probability p_{gen} is the same for each pair of adjacent nodes and the swapping probability p_{swap} is equal at each nesting level. We also assume that the nodes are capable of generating entanglement in parallel. Finally, we ignore the (constant) duration of local operations and classical communication for simplicity, although all of the tools mentioned are capable of incorporating these.

We first investigate methods that instead of tracking the full probability distribution, only track an approximation of the average waiting time and quantum state at each nesting level of the entanglement-distribution protocol. To demonstrate the tools used in computing the distributions, we include an explicit calculation for a protocol on two

nodes with a single repeater positioned in between. As this exact calculation cannot be directly generalised to a higher nesting level in a nested protocol with more than a single repeater, we then consider the idea of approximating the waiting time by assuming it follows the statistics of elementary-link generation, where the mean waiting time is computed using the approximation from the previous level. Finally, we review the mathematical tools and approximation methods used to analyse deterministic swapping protocols and distillation-based repeater schemes.

4.2.1. THE MEAN-ONLY APPROACH

In many early analyses of repeater protocols, only the mean waiting time is considered for each nesting level: it is assumed that the entanglement is delivered after a fixed time duration determined by the generation rate. We refer to it as the mean-only approach. In this approach, the mean waiting time is computed as the product of the mean waiting time at each nesting level ($1/p_{\text{gen}}$ at the bottom level, $1/p_{\text{swap}}$ at the higher levels), yielding $T_n = 1/p_{\text{swap}}^n p_{\text{gen}}$ [16, 17]. This approach can be refined by noting that at each nesting level the protocol proceeds only when two adjacent pairs are ready. Then, the mean waiting time can be approximated by $T_n = (3/2)^n/p_{\text{swap}}^n p_{\text{gen}}$ [19, 22, 24–29]. The factor $3/2$ comes from the fact that in the limit of very small success probability, the waiting time of preparing two links is approximately $3/2$ times that of one link. We discuss the statistics behind this factor later in Section 4.2.3 and Chapter 7.

The mean-only approach is a good approximation when p_{gen} and p_{swap} are much smaller than 1[24, 25]. However, it only approximates the mean, i.e. it does not provide the entire probability distribution of the waiting time. Hence, it is not suited for investigating time-dependent aspects such as memory noise or cut-offs. With this method, memory decoherence is either approximated by an inefficiency constant [28] or studied only for the communication time [30]. To provide a better estimation, one needs to consider the waiting time distribution and the statistics it results in, which we discuss below.

4.2.2. SINGLE REPEATER SWAP PROTOCOL

Here, we explicitly compute the probability distribution of the waiting time for the simplest repeater chain: a single repeater between two end nodes. We also derive an expression for the mean fidelity decay due to memory decoherence. Many problems regarding single-repeater protocols have an analytical solution because the entanglement generation follows a known distribution. By studying this simple scenario, we demonstrate the common concepts and methods used to describe and compute the statistics of waiting time and fidelity. In later sections, we use this calculation as a basis for the analysis of nested repeater protocols of more nodes.

We describe the waiting time of elementary-link generation as a random variable T_0, following a geometric distribution given by

$$\Pr(T_0 = t) = p(1-p)^{t-1}, \tag{4.1}$$

where $t \in \{1, 2, 3...\}$ and $p = p_{\text{gen}}$. This distribution plays a central role in the statistics of entanglement distribution, as we see in the remaining part of this section. In the limit

of $p_{\text{gen}} \ll 1$, the geometric distribution can be approximated by an exponential distribution,

$$\Pr(T_0 = t) = p_{\text{gen}} \cdot \exp\left(-p_{\text{gen}}\, t\right) \tag{4.2}$$

which is a continuous distribution with $t \geq 0$. Note that we have set the attempt duration Δ of entanglement generation to 1 (Section 4.1).

To perform an entanglement swap, both elementary links have to be prepared first. We define the time used in preparing them as M_0:

$$M_0 = \max(T_0, T_0'), \tag{4.3}$$

where T_0' is an independent copy of T_0. The distribution of M_0 can be computed using the fact that

$$\Pr(\max(X, Y) \leq t) = \Pr(X \leq t) \cdot \Pr(Y \leq t) \tag{4.4}$$

for any independent random variables X and Y. The mean of M_0, i.e. the waiting time until both elementary links have been prepared, is given by [26]:

$$E[M_0] = \frac{3 - 2p_{\text{gen}}}{(2 - p_{\text{gen}})\, p_{\text{gen}}}. \tag{4.5}$$

After two elementary links are prepared, the repeater node performs an entanglement swap, which is a probabilistic operation with success probability p_{swap}. The total waiting time for generating the entanglement between two end nodes is therefore

$$T_1 = \sum_{k=1}^{K} M_0^{(k)}, \tag{4.6}$$

where K represents the number of swap attempts until it succeeds and $M_0^{(k)}$ are independent copies of M_0. Eq. (4.6) is referred to as a compound distribution since the number of summands K is also a random variable. For an entanglement swap, the number of attempts K also follows a geometric distribution (Eq. (4.1)) with success probability $p = p_{\text{swap}}$. Because K and M_0 are independent, the average waiting time is given by

$$E[T_1] = E[M_0] \cdot E[K] = \frac{3 - 2p_{\text{gen}}}{(2 - p_{\text{gen}})\, p_{\text{gen}}} \cdot \frac{1}{p_{\text{swap}}}. \tag{4.7}$$

The intuition behind Eq. (4.7) is that, on average, the repeater node requires $E[K]$ swap attempts until the first successful swap, and for each swap attempt, $E[M_0]$ attempts are needed to prepare the two elementary links.

Computing the fidelity of the two elementary links just before swapping can be done as follows. If the generation of elementary links is not deterministic, i.e. if $p_{\text{gen}} < 1$, the two elementary links are in general not produced at the same time, requiring the earlier of the two to be stored in a quantum memory. This storage time results in decoherence of the earlier link. To estimate the fidelity decrease, we need to first compute the distribution of the memory storage time, i.e. the time difference between the generation of

the earlier and the later link. We define $q_g = 1 - p_{gen}$. The probability that one link is prepared j steps before the other is given by [27, 31]

$$\Pr\left(T_0 - T_0' = j\right) = \sum_{t=1}^{\infty} p_{gen}^2 q_g^{2(t-1)+j} = \frac{p_{gen} q_g^j}{2 - p_{gen}}. \tag{4.8}$$

Here we assume that $T_0 > T_0'$. Modelling the fidelity decrease as exponential-decaying function of the storage time, the fidelity of the earlier link decays by a factor $\Gamma = E\left[\exp\left(-|T_0 - T_0'|/t_{coh}\right)\right]$, where t_{coh} denotes the memory coherence time. Plugging in Eq. (4.8), we obtain

$$\Gamma = \frac{p_{gen}}{2 - p_{gen}} + 2\sum_{j=1}^{\infty} \exp\left(\frac{j}{t_{coh}}\right) \cdot \frac{p_{gen} q_g^j}{2 - p_{gen}}.$$

The factor 2 before the sum corresponds to the possibility that either link can be generated earlier than the other. Finally, the evaluation of the sum gives [27, 31]

$$\Gamma = \frac{p_{gen}}{2 - p_{gen}} \left(1 + \frac{2}{1 - q_g \exp\left(-\frac{1}{t_{coh}}\right)}\right). \tag{4.9}$$

In addition to the single-repeater scenario considered above, analytical results for the memory decay have also been obtained for more advanced single repeater protocols such as a protocol with cut-offs [14] or protocols where two elementary links have to be prepared sequentially [32].

Unfortunately, for higher-level nested protocols, i.e. $n \geq 1$, there is no analytical expression known for the mean waiting time $E[T_n]$ with $p_{swap} < 1$, because T_i for $i > 0$ does not follow a geometric distribution, in contrast to T_0.

4.2.3. APPROXIMATION WITH THE GEOMETRIC DISTRIBUTION AT HIGHER LEVELS

Above, we computed the waiting time probability distribution in the single-repeater scenario. This calculation explicitly relied on the fact that the waiting time distribution of elementary-link generation follows a simple distribution, the geometric distribution (Eq. (4.1)). Unfortunately, for nested repeater chains with more than a single repeater, no exact expression for the waiting time distribution has been found.

However, the waiting time distribution at higher nesting levels can be approximated by assuming that, at a higher level, the waiting time distribution is still geometrically or exponentially distributed (Eq. (4.2)). This approximation is usually used in an iterative manner. One computes the average waiting time at the current level and uses it to define a geometric distribution with the same expectation value. This new distribution is then used to study the next nesting level. In Fig. 4.2, we compare the approximated distribution with the exact one.

Let us give the explicit calculation under the approximation that the waiting time follows a geometric distribution at each nesting level. We first calculate $E[T_{n-1}]$ and then approximate the distribution of T_{n-1} with a geometric distribution (Eq. (4.1)) parameterised by $p = 1/E[T_{n-1}]$. Under this assumption, the mean waiting time $E[T_n]$ can be

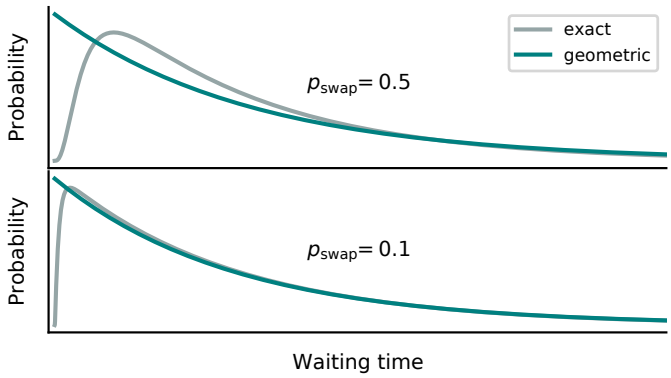


Figure 4.2: The probability distribution of the exact waiting time T_2 of a nested swap protocol with 4 repeater segments (computed with the algorithm from Chapter 6) and, as an approximation to the exact distribution, the geometric distribution from Eq. (4.1) with the same mean, i.e. $p = 1/E[T_2]$. (Top) We see that the two distributions deviate most for short waiting times. This can be explained by noting that the exact probability that all entanglement generation steps and entanglement swaps succeed in the first few steps is very small. This fact is not captured by the approximation. (Bottom) We observe that for small swap success probabilities p_{swap} (both axes are rescaled to compare only the shape of the distributions), the deviation becomes smaller. In Chapter 7, we will give analytical bounds on the waiting time distribution that show that in the limit $p_{\text{swap}} \to 0$, the waiting time distribution becomes an exponential distribution.

computed by a derivation analogous to the one leading to Eq. (4.7) in Section 4.2.2 and is given by

$$E[T_n] = \frac{3 - 2p_{n-1}}{(2 - p_{n-1})p_{n-1}p_{\text{swap}}} \tag{4.10}$$

with $p_{n-1} = 1/E[T_{n-1}]$. In the limit of $p_{\text{gen}} \to 0$ for the bottom level ($n = 0$) and $p_{\text{swap}} \to 0$ for higher levels ($n > 0$), the mean waiting time $E[T_{n-1}] \to \infty$ and thus $p_{n-1} \to 0$. As a consequence, Eq. (4.10) can be approximated as

$$E[T_n] \approx \frac{3}{2p_{n-1}p_{\text{swap}}}. \tag{4.11}$$

Effectively, it means that, on average, the waiting time of generating two links is approximately 3/2 times that of a single link. Applying Eq. (4.11) iteratively over all nesting levels with $E[T_0] = 1/p_{\text{gen}}$ yields

$$E[T_n] \approx \frac{3^n}{2^n p_{\text{swap}}^n p_{\text{gen}}}, \tag{4.12}$$

which is precisely the 3-over-2 approximation mentioned in Section 4.2.1.

The error introduced by the approximations Eq. (4.12) and Eq. (4.10) is shown in Fig. 4.3. As expected, the figure shows that the approximations behave well if the success probabilities of elementary-link generation and swapping are small, i.e. $p_{\text{gen}} \to 0$ and $p_{\text{swap}} \to 0$. The figure also shows that the approximations are not so good for large success probabilities; the deviation from the exact mean waiting time increases as p_{swap} grows, and the deviation is worse for Eq. (4.12) than for Eq. (4.10). In Chapter 7, we will

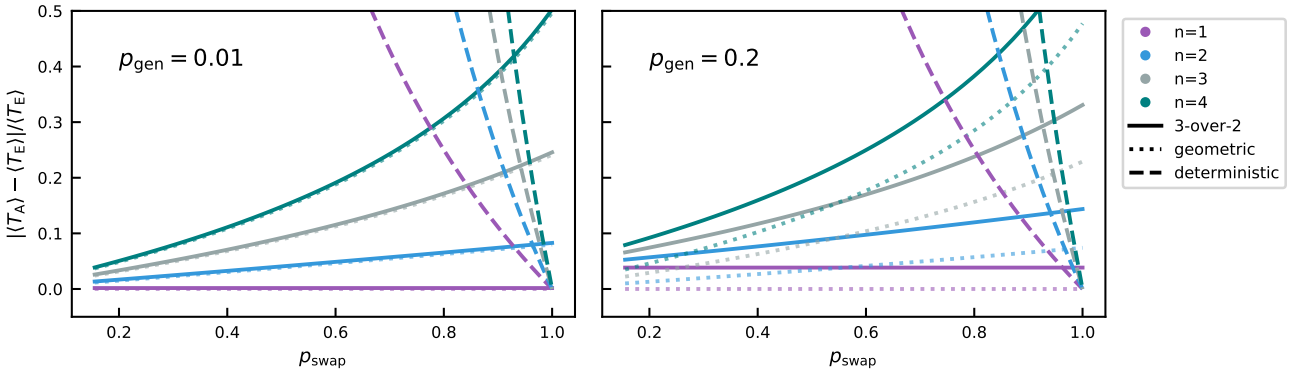


Figure 4.3: Comparing the relative error in the mean waiting time $\left|E\left[T_A\right] - E\left[T_E\right]\right| / E\left[T_E\right]$ among different approximation methods for the mean waiting time of a nested repeater protocol on n nesting levels, where $E\left[T_A\right]$ and $E\left[T_E\right]$ are the approximated and exact mean waiting time, respectively. We plot the relative difference between the approximated and the exact mean waiting time as a function of the swap success probability p_{swap} for nested swap protocols up to level $n = 4$. The three approximation methods are: 1. The approximation with geometric distribution given by Eq. (4.10). It is exact at level 1 and the deviation increases as the levels grow; 2. The 3-over-2 approximation given by Eq. (4.12), which is itself an approximation of Eq. (4.10). In the limit of $p = p_{gen} \ll 1$, the difference between the two approximation vanishes; 3. The deterministic swap approximation (Eq. (4.15)) assuming that the swap always succeeds ($p_{swap} = 1$). Note that, in contrast to the former two, the latter approximation is a lower bound on the exact waiting time. The deviation of the deterministic swap approximation is almost independent of p_{gen}. The exact mean waiting time $E\left[T_E\right]$ is computed using the algorithm in Section 4.3.2. Due to our implementation, we cannot reach the region $p_{swap} \to 0$. However, Shchukin *et al.* numerically show that, in this limit, the deviation of the approximation with the geometric distribution becomes negligible (described in Ref. [33]); we show this fact analytically in Chapter 7 by providing two-sided bounds on the mean which coincide in the small-probability limit.

provide two-sided analytical bounds on the mean waiting time which quantify the quality of the 3-over-2 approximation.

To approximate the fidelity of the produced link between the end nodes of the repeater chain in the presence of memory decoherence, one can use Eq. (4.9) by replacing p_{gen} with $1/E[T_{n-1}]$. The approximation is computed analogously to the procedure described for the average waiting time; that is, for a given level, the average infidelity for the entanglement links just before a swap due to the memory decoherence is calculated and used to derive the initial infidelity for entangled links at the next level. By assuming that the distribution at every level is given by the exponential distribution (Eq. (4.2)), Kuzmin *et al.* designed a semi-analytical method for computing fidelity with more sophisticated memory decay models [34, 35] (see also Section 4.3.2).

A different approach to keep the waiting time distribution geometric at a higher level is to design a special protocol. For example, Santra *et al.*[36] introduce a family of protocols with a memory buffer time. This buffer time is a threshold on the total waiting time for preparing the two links for the swap. If the links are not ready before the buffer time is reached, the protocol aborts and restarts from entanglement generation. The buffer time is slightly different from the memory cut-off (see Section 4.1); with a memory cut-off the protocol aborts if the memory storage time of a single link (instead of both links) exceeds a threshold.

The protocol is designed such that the buffer time at the current level becomes the time step at the next level. As a consequence, the waiting time is geometrically distributed at each nesting level. Note that the protocol results in avoidable additional memory decay as both links have to wait until the buffer time is reached even if they are prepared before that. Despite this, by optimising the buffer time, Santra *et al.* show that this protocol improves the final fidelity for some parameter regimes compared to the nested repeater protocol without buffer times.

An alternative approach was taken for optimising repeater protocols including distillation, where the buffer time is chosen large enough so that the protocol becomes nearly deterministic [37]. At a cost of longer waiting time and lower fidelity, the variance in the fidelity is kept small and the protocol can deliver entanglement at a pre-specified time with high probability.

4.2.4. DETERMINISTIC ENTANGLEMENT SWAP

So far we have focused on the regime where the success probability of entanglement swap is smaller than 1. In this section, we discuss nested repeater protocols with deterministic entanglement swapping ($p_{\text{swap}} = 1$) and without distillation.

First, let us compute the waiting time probability distribution in the case of deterministic swaps without distillation or cut-offs. Recall that we assume that the time required to perform local operations is negligible so that the deterministic entanglement swap has no contribution to the waiting time. For a repeater scheme with n nesting levels, the total waiting time is the time until all $N = 2^n$ elementary links are prepared, i.e. the maximum of N independent copies of T_0:

$$T_N = \max(T_0^{(1)}, T_0^{(2)}, \ldots, T_0^{(N)}). \tag{4.13}$$

The cumulative distribution of T_N from Eq. (4.13) is given by the general version of

Eq. (4.4) for the maximum of N independent and identically distributed random variables:

$$\Pr(T_N \leq t) = \Pr\left(\max(T_0^{(1)}, \ldots, T_0^{(N)}) \leq t\right) = \Pr(T_0 \leq t)^N$$

from which the probability distribution of T_N can be computed using

$$\Pr(T_N = t) = \Pr(T_N \leq t) - \Pr(T_N \leq t-1).$$

By $T_{N,k}$, we denote the random variable describing the time at which the first k elementary links of the N segments are generated. We first give the expression for $\Pr(T_{N,k} \leq t)$, the probability that at least k links are generated before t. This is equivalent to the probability that, at time t, the number of elementary links that have not yet been generated is $N-k$ or less [38]:

$$\Pr(T_{N,k} \leq t) = \sum_{j=0}^{N-k} \binom{N}{j} (1 - \Pr(T_0 \leq t))^j \Pr(T_0 \leq t)^{N-j}$$

where $\Pr(T_0 \leq t) = 1-(1-p_{\text{gen}})^t$ since T_0 is geometrically distributed with success probability p_{gen}. The probability that precisely k of the N segments are generated at time t is

$$\Pr(T_{N,k} = t) = \Pr(T_{N,k} \leq t) - \Pr(T_{N,k} \leq t-1),$$

from which the mean waiting time is calculated as

$$E[T_{N,k}] = \sum_{t=1}^{\infty} t \cdot \Pr(T_{N,k} = t). \tag{4.14}$$

The mean waiting time $E[T_{N,k}]$ from Eq. (4.14) can be computed by solving a recurrence formula where $E[T_{N,k}]$ is expressed as function of $E[T_{N,k-1}]$ [39, 40]. For $k = N$, i.e. the waiting time that all elementary entanglement are prepared [39], the solution reads

$$E[T_{N,N}](p) = \sum_{k=1}^{N} \binom{N}{k} \frac{(-1)^{k+1}}{1-(1-p_{\text{gen}})^k}. \tag{4.15}$$

For $p_{\text{gen}} \ll 1$, this expression can be approximated by [31, 33, 38]

$$E[T_{N,N}](p) \approx \sum_{k=1}^{N} \binom{N}{k} \frac{(-1)^{k+1}}{k p_{\text{gen}}} = \sum_{k=1}^{N} \frac{1}{k p_{\text{gen}}} = \frac{H(N)}{p_{\text{gen}}} \tag{4.16}$$

with

$$H(N) = \sum_{k=1}^{N} \frac{1}{k} \approx \gamma + ln(N) + \frac{1}{2N} + \mathcal{O}(\frac{1}{N^2}),$$

where $H(N)$ is the N-th harmonic number and $\gamma \approx 0.57721$ is the Euler-Mascheroni constant. In separate work, Praxmeyer included finite memory time with cut-off into the calculation and obtained [40]

$$E[T_{N,N}] = \frac{1-(1-q_{\text{g}}^{\tau})^N + (1-q_{\text{g}}^N)\left[\tau - \sum_{j=1}^{\tau-1}(1-q_{\text{g}}^j)^N\right]}{(1-q_{\text{g}}^{\tau+1})^N - q_{\text{g}}^N(1-q_{\text{g}}^{\tau})^N}, \tag{4.17}$$

where τ is the cut-off threshold and $q_g = 1 - p_{gen}$.

Similar derivations as the ones above can be used for the waiting time until the first, instead of the last, elementary link has been generated. Those derivations are relevant for the analysis of multiplexed repeater protocols and the mean waiting time in those cases has been analysed in [14, 38].

To our knowledge, in contrast to the waiting time, there is no exact fidelity calculation with exponential memory decoherence for deterministic swap. A lower bound on the fidelity can be obtained by assuming the worst case, i.e. the swap is performed only after all elementary links are generated [38, 39].

These expressions presented here for the deterministic-swap case also apply to repeater chains where the numbers of segments is not a power of 2, as well as to more general network topology [41]. The reason for this is that if the swaps are deterministic, the waiting time equals the time until all elementary links in the network have been prepared. Thus, the nested structure does not exist and the only relevant parameters are the number of elementary links and the elementary-link success probability p_{gen}.

The waiting time in the deterministic-swap case can be used as an approximation to the case where p_{swap} is slightly lower than 1 and bounds from below the waiting time for general p_{swap}. The quality of the approximation is shown in Fig. 4.3.

4.2.5. METHODS FOR ANALYSING DISTILLATION-BASED REPEATER SCHEMES WITH MEMORY-DECOHERENCE

In contrast to entanglement swapping, distillation has a fidelity-dependent success probability. In the absence of decoherence, the fidelity of a pair does not decrease while it is waiting for other components to succeed. Hence, the success probability p_{dist} is a constant for each level and distillation can be studied in the same way as entanglement swapping. However, in the presence of decoherence, fidelity and success probability become correlated, which complicates the analysis.

We finish by mentioning two tools for bounding the fidelity and generation rate of distillation-based repeater schemes in the presence of memory decoherence. First, upper bounds on the achievable fidelity can be derived using fixed-point analysis [10, 16]. In this approach, one makes use of the fact that entanglement distillation does not improve the fidelity when the quality of the input links is sufficiently high. Such a fidelity is thus a fixed point of the entanglement distillation procedure and it depends on the quality of the local operations [16] and memories [30]. If the fixed-point is an attractor and the input links have fidelity lower than the fixed-point, repeated application of entanglement distillation cannot boost fidelity beyond the fixed-point and it thus forms an upper bound. Next, lower bounds on the fidelity can be trivially obtained for protocols that impose a fidelity cut-off, i.e. protocols that discard the entanglement if the fidelity is lower than a certain threshold [42]. Because the distillation success probability is a monotonic function of the fidelity of the input states, a lower bound on the fidelity by a cut-off also directly yields a lower bound on the success probability.

4.3. NUMERICAL TOOLS FOR EVALUATING ANALYTICAL EXPRESSIONS

Above, in Section 4.2, we reviewed analytical tools for computing the probability distribution of the waiting time for generating remote entanglement and of the entanglement quality. For models that do not include memory decoherence, distillation, or cutoffs, these tools are sufficient. For more complex models and for the analysis of many-node networks, the tools presented above are often still applicable but analytically evaluating the resulting expressions to compact, closed-form expressions is unfeasible. An example of such a case is a nested repeater chain with cut-offs and non-deterministic swapping. In this case, no concise analytical expression for the waiting time is known. A priory, it is possible to write down a recursive analytical expression for the waiting time using a similar reasoning to Section 4.2.2, where the single-repeater case was treated. However, the recursion relation has so far not been solved for general repeater chains. Fortunately, numerical tools enable the evaluation of such expressions. In this section, we treat three classes of such tools: Markov Chain algorithms, probability-tracking algorithms and Monte Carlo methods for abstract models.

4.3.1. MARKOV CHAIN METHODS

In many repeater protocols, the change of the entanglement in the network in the next time step only depends on the existing entanglement. Shchukin *et al.* used this idea to model the network as a discrete Markov chain [33], which can be visually depicted as a directed graph, an example of which is shown in Fig. 4.4. A vertex in this graph is a state of the network, i.e. the collection of entanglement that exists at a given point in time. The network transitions from one state to the other with a fixed probability, which is visualised by directed edges of the labelled graph. At each time step, a network randomly transitions from its current state to the next state according to the transition probabilities over outgoing edges. For example, in the single-repeater protocol depicted in Fig. 4.4, the transition from the 'an entangled pair exists on each of the two segments' state (11) to 'entanglement exists between end nodes' ($\overline{11}$) occurs with the entanglement swapping success probability p_{swap} (entanglement swapping succeeded), whereas the transition to 'no entanglement' (state 00) has probability $1 - p_{\text{swap}}$ (entanglement swapping failed and the two involved links are lost).

An equivalent representation of a Markov chain is the transition probability matrix (TPM), where entry (j,k) represents the transition probability from state j to state k. Since a single transition corresponds to a single time step, the waiting time distribution equals the distribution of the number of edges traversed before reaching a predefined target state, such as 'entanglement between the end nodes of the repeater chain' ($\overline{11}$ in Fig. 4.4). Shchukin *et al.* used this equivalence to compute the average waiting time, as well as its variance, by solving a linear equation system that has the same size as the number of states in the Markov chain.

The original proposal by Shchukin *et al.* included an analysis of the waiting time. Later, the idea was refined to include memory decoherence by Vinay *et al.* [42] They computed the fidelity distribution by assigning a noise parameter to certain transition edges and calculated how many times the edges are traversed given that the entanglement dis-

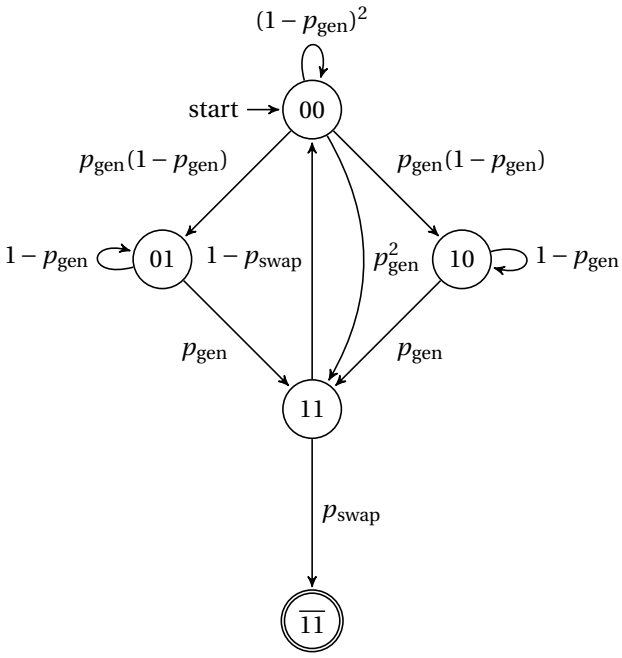


Figure 4.4: The directed graph for a Markov chain of the single repeater swapping protocol, which consists of two end nodes with a repeater node positioned in between. A vertex in the graph corresponds to a state of the network, while the labelled edges represent possible transitions between states with corresponding transition probabilities. The Markov state 00 represents the initial state with no entanglement; state 01 (10) is the state with one elementary link on the right (left) segment; state 11 is the state with an elementary link on both segments; state $\overline{11}$ denotes the state after the successful entanglement swapping by the repeater node, yielding entanglement between the end nodes. The double cycle indicates that this Markov state is absorbing, i.e. has only incoming transitions. Such an absorbing state corresponds to the protocol being finished. Reprinted with permission from E. Shchukin, F. Schmidt, and P. van Loock, Phys. Rev. A, vol. 100, p. 032322, (2019)[33]. Copyright 2019 by the American Physical Society.

tribution is completed at time t. With this noise model, the Markov chain method was used to study the BDCZ protocol [16], which includes entanglement distillation. Due to the assumption of the Markov process, i.e. the system has no memory of the past, this method cannot handle fidelity-dependent success probability without assigning each possible fidelity a state representation. As an alternative, Vinay *et al.* provided a lower bound to the final fidelity using fidelity cutoffs (see Section 4.2.5).

Apart from repeater chain protocols, Markov chains have also been used to study more general network protocols, such as a quantum switch by Vardoyan *et al.* [43, 44], who used the continuous-time Markov chains as an approximation to discrete-time Markov chains. In this model, the transition probability is replaced by the transition rate. Compared to their discrete counterparts, continuous-time Markov chains simplify the analysis in various aspects. For instance, Vardoyan *et al.* included a model of decoherence where the states are lost at a fixed rate by adding an additional transition edge indicating the loss of one entangled pair. Moreover, Vardoyan *et al.* show that the quality of the approximation is high in many scenarios [43].

The Markov chain method is rather general and flexible: in principle, the waiting time of any entanglement distribution protocol with predefined transition probabilities can be calculated, regardless of the topology or entanglement swapping policy (such as swapping as soon as two links are available, regardless of the segments on which this entanglement has been produced, or swapping only between predefined segments). However, this method is computationally expensive. The size of the TPM is the same as the number of possible Markov states and, in general, grows exponentially with the number of nodes.

This rapid growth of the size of TPM can be partially mitigated. For instance, by grouping equivalent Markov states and treating them as one state, the complexity can be drastically reduced. With this technique, Shchukin *et al.* gave examples for the BDCZ protocol with analytical expressions for up to 4 nodes, while numerically they reached 32 nodes [33]. Vinay *et al.* reduced the computational complexity of this approach using probability generating functions and complex analysis, but the scaling remains exponential [42]. To process a larger number of nodes, Shchukin *et al.* proposed to use the average waiting time to replace the random variable for low-level sub-protocols. This idea is similar to approximating the waiting time distribution at every nesting level of a repeater protocol with the bottom-level distribution (see Section 4.2.3).

Finally, in a recent development, Khatri [45] introduced a method for describing network protocols based on quantum partially observable Markov decision processes [46]. A quantum partially observable Markov decision process is a reinforcement-learning based framework for protocol optimisation. In this framework, the protocol obtains feedback from its actions in the form of classical information about the quantum state that the network holds, which it uses to optimise the next action it will perform. As an application of the method, Khatri found analytical solutions for optimising a cut-off for elementary link generation under different constraints. It is an interesting open question whether this approach can be extended for efficiently characterising and optimising protocols over large repeater chains and networks.

4.3.2. PROBABILITY-TRACKING ALGORITHMS

Next, we treat the algorithm that we have developed in [47, 48] and which we present in more detail in Chapters 5 (initial version) and 6 (runtime improvement). The algorithm tracks the full waiting time probability distribution and the average fidelity of the delivered quantum state. We explain this method via a concrete example, a symmetric nested repeater protocol with 2^n segments and no entanglement distillation (depicted in Fig. 4.1(b) for $n = 2$). In Section 4.2.2, we treated the case for $n = 1$, which resulted in an expression for the waiting time random variable consisting of the maximum of two copies of the waiting time of the bottom level (Eq. (4.3)) and a compound sum (Eq. (4.6)). The first element, the maximum, corresponds to the fact that an entanglement swap acts on two links that need both be generated, so one needs to wait until the latest of the two links has been prepared. The second element is the sum of the waiting time until the first successful swap attempt. Since the number of attempts is probabilistic, the result is a compound sum. The analysis for the $n = 1$ case can be generalised to an arbitrary number of nesting levels n and yields a recursive expression of the waiting time T_n which alternates between compound sums and maximums of two copies of the waiting time T_{n-1} on the previous level. Unfortunately, as discussed in Section 4.2, to our knowledge, this recursive expression of T_n has not been analytically evaluated for $n > 1$. Hence, various approximation methods were introduced in that section. The exact evaluation can, however, be achieved with numerical tools. By truncating the waiting time at a prespecified time t_{trunc}, the waiting time distribution becomes finite. The evaluation with numerical tools leads to an algorithm that tracks both the truncated probability distribution of T_n and the associated fidelity, see [47, 48] and Chapters 5 and 6.

On the 2^n-segment nested repeater protocol, the algorithm computes the waiting time distribution as follows. If $n = 0$, i.e. if there is no repeater and the two end nodes obtain entanglement by direct generation, the waiting time T_0 follows a geometric distribution (Eq. (4.1)). If $n = 1$, i.e. there is a single repeater and two segments, then the algorithm evaluates Eq. (4.6), which describes how the probability distribution of the waiting time T_1 can be obtained from the distribution of T_0. Although the two elements in Eq. (4.6), the maximum and the geometric compound sum, can in principle be evaluated sequentially[47], a computationally faster approach is to separate the probability distributions of failed and successful swap attempts[48]. For $n > 1$, the algorithm is applied iteratively over the nesting levels until level n has been reached. The algorithm can be extended in polynomial time to also track the average fidelity of the produced quantum state. This fidelity is a function of the delivery time and it can include the effect of memory decoherence.

Although the example protocol above only consists of entanglement swapping, the algorithm is applicable to any protocol which is composed of entanglement distillation and cut-offs, in addition to entanglement swaps[48]. The algorithm presupposes that the protocol is composed of these three operations in a predefined order, e.g. which entangled pairs are swapped must be known in advance. The algorithm scales polynomially in the number of nodes and in the truncation time t_{trunc} and has been used to track over 1000 nodes for some parameter regimes [47].

A related approach to the probability-tracking algorithm explained above is taken by Kuzmin *et al.* [34, 35]. This method assumes that the waiting time of an elementary link

is exponentially distributed (Eq. (4.2)), after which the mean waiting time for the next level is computed by evaluating a continuous integral, as well as the quantum state in the presence of memory decay. These are then used as input to the next nesting level, by assuming that at that level, the waiting time follows an exponential distribution also. With this approximation, the calculation of the maximum in Eq. (4.3) is simplified.

4.3.3. SAMPLING THE ANALYTICAL EXPRESSIONS WITH MONTE CARLO METHODS

So far in this section, we have discussed two methods that compute the statistical distribution of the waiting time and produced quantum state. For a given model, both of them evaluate the analytical expression exactly up to machine precision. An alternative to this semi-analytical computation is to sample the expressions on random variables for the waiting time and the delivered state using a Monte Carlo approach, which we developed in [47] and explain in more detail in Chapter 5. Instead of tracking the whole distribution, this approach samples a pair of waiting time and the produced state between the end nodes. By sampling many times, the probability distribution of the waiting time and the quantum state can be reconstructed.

Again, let us take the 2^n-segment nested repeater protocol as an example to explain the algorithm. The individual sample pairs are produced by iterating over the different components of the repeater protocol, following its nested structure. At each component, a pair is sampled recursively, following the expressions on random variables, which thus become expressions on individual events. For instance, Eq. (4.3) requires sampling two instances of T_0 for entanglement generation and then taking the maximum of both to produce a sample of M_0. Also, memory decoherence can be calculated from the time difference of two events. Similarly, the method can handle cut-offs. Furthermore, distillation can also be included in the protocol, since the input states to a distillation attempt, which determine its success probability, are also sampled. For nested protocols, the Monte Carlo algorithm can be defined as a recursive function, following the nested structure of the protocols.

4.4. SECOND AND THIRD GENERATION REPEATER PROTOCOLS

So far, we have only treated first-generation quantum repeater protocols, i.e. protocols for which the building blocks – fresh entanglement generation, entanglement swapping, and entanglement distillation – are probabilistic operations. The quantum repeater proposals that do not fall into this category make explicit use of quantum error correction codes and are referred to as second-generation (probabilistic entanglement generation, deterministic entanglement swapping, and one-way quantum error correction) and third-generation repeaters (loss-tolerant entanglement generation) [1, 49].

In first-generation repeaters, entanglement generation and swapping are probabilistic, and once it has succeeded, the entanglement is kept in quantum memories and needs to wait until other components performed in parallel have succeeded. Consequently, the waiting time probability distribution and state quality are a complex function of the success probabilities of components in the repeater chain.

In contrast, for second-generation repeater protocols, such as [50–53], entanglement

swapping and (one-way) quantum error correction are no longer probabilistic (although entanglement generation is still probabilistic, it may be parallelised to achieve near-unit generation success probability). As a result, the time at which the entire repeater chain finishes with a single attempt at generating end-to-end entanglement is simply a sum of the (constant) times that the individual components take. Not only there is no waiting for other components, but there are also no feedback loops here, i.e. components that need to restart in case others have failed. The unit time step at which such repeater chains operate is an attempt at end-to-end entanglement generation (i.e. the sum of the individual component times); the probability that such an attempt succeeds is the product of all individual steps succeeding. For this reason, the distribution of the waiting time is a geometric distribution.

A similar reasoning applies to third generation repeaters [49, 54–64], where entanglement between adjacent nodes is established almost deterministically, rather than probabilistically, by encoding part of locally-generated entanglement into a large state of photons, followed by transmission of the encoded state. Commonly, the analysis of the propagation of operational errors (for 2nd and 3rd generation) and the propagation of physical loss errors into logical errors (for 3rd generation) is based on work on quantum error correction codes (combined with explicit counting of the combinations of losses of photons which yield errors beyond recovery) and optical quantum computation. We consider such tools out of scope for this review chapter.

REFERENCES

[1] W. J. Munro, K. Azuma, K. Tamaki, and K. Nemoto, *Inside quantum repeaters,* IEEE Journal of Selected Topics in Quantum Electronics **21**, 78 (2015).

[2] M. Takeoka, S. Guha, and M. M. Wilde, *Fundamental rate-loss tradeoff for optical quantum key distribution,* Nature Communications **5**, 5235 (2014).

[3] M. Żukowski, A. Zeilinger, M. A. Horne, and A. K. Ekert, *"Event-ready-detectors" Bell experiment via entanglement swapping,* Phys. Rev. Lett. **71**, 4287 (1993).

[4] J. Calsamiglia and N. Lütkenhaus, *Maximum efficiency of a linear-optical bell-state analyzer,* Applied Physics B **72**, 67 (2001).

[5] W. P. Grice, *Arbitrarily complete bell-state measurement using only linear optical elements,* Physical Review A **84**, 042331 (2011).

[6] A. Olivo and F. Grosshans, *Ancilla-assisted linear optical bell measurements and their optimality,* Physical Review A **98**, 042323 (2018).

[7] F. Ewert and P. van Loock, *3/4-efficient bell measurement with passive linear optics and unentangled ancillae,* Phyisical Review Letters **113**, 140403 (2014).

[8] Y. Wu, J. Liu, and C. Simon, *Near-term performance of quantum repeaters with imperfect ensemble-based quantum memories,* Phys. Rev. A **101**, 042301 (2020).

[9] C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, and W. K. Wootters, *Purification of noisy entanglement and faithful teleportation via noisy channels,* Phys. Rev. Lett. **76**, 722 (1996).

[10] D. Deutsch, A. Ekert, R. Jozsa, C. Macchiavello, S. Popescu, and A. Sanpera, *Quantum privacy amplification and the security of quantum cryptography over noisy channels,* Phys. Rev. Lett. **77**, 2818 (1996).

[11] R. F. Werner, *Quantum states with Einstein-Podolsky-Rosen correlations admitting a hidden-variable model,* Phys. Rev. A **40**, 4277 (1989).

[12] W. Dür and H. Briegel, *Entanglement purification and quantum error correction,* Reports on Progress in Physics **70**, 1381 (2007).

[13] W. Dür, H.-J. Briegel, J. I. Cirac, and P. Zoller, *Quantum repeaters based on entanglement purification,* Phys. Rev. A **59**, 169 (1999).

[14] O. A. Collins, S. D. Jenkins, A. Kuzmich, and T. A. B. Kennedy, *Multiplexed memory-insensitive quantum repeaters,* Phys. Rev. Lett. **98**, 060502 (2007).

[15] S. Abruzzo, H. Kampermann, and D. Bruß, *Measurement-device-independent quantum key distribution with quantum memories,* Phys. Rev. A **89**, 012301 (2014).

[16] H.-J. Briegel, W. Dür, J. I. Cirac, and P. Zoller, *Quantum repeaters: The role of imperfect local operations in quantum communication,* Phys. Rev. Lett. **81**, 5932 (1998).

[17] L.-M. Duan, M. D. Lukin, J. I. Cirac, and P. Zoller, *Long-distance quantum communication with atomic ensembles and linear optics,* Nature **414**, 413 EP (2001), article.

[18] P. Kok, C. P. Williams, and J. P. Dowling, *Construction of a quantum repeater with linear optics,* Physical Review A **68**, 022301 (2003).

[19] L. Childress, J. M. Taylor, A. S. Sørensen, and M. D. Lukin, *Fault-tolerant quantum repeaters with minimal physical resources and implementations based on single-photon emitters,* Phys. Rev. A **72**, 052330 (2005).

[20] P. Van Loock, T. Ladd, K. Sanaka, F. Yamaguchi, K. Nemoto, W. Munro, and Y. Yamamoto, *Hybrid quantum repeater using bright coherent light,* Phyisical Review Letters **96**, 240501 (2006).

[21] W. Munro, R. Van Meter, S. G. Louis, and K. Nemoto, *High-bandwidth hybrid quantum repeater,* Phyisical Review Letters **101**, 040502 (2008).

[22] K. Azuma, H. Takeda, M. Koashi, and N. Imoto, *Quantum repeaters and computation by a single module: Remote nondestructive parity measurement,* Physical Review A **85**, 062309 (2012).

[23] M. Zwerger, W. Dür, and H. Briegel, *Measurement-based quantum repeaters,* Physical Review A **85**, 062326 (2012).

[24] L. Jiang, J. M. Taylor, and M. D. Lukin, *Fast and robust approach to long-distance quantum communication with atomic ensembles,* Phys. Rev. A **76**, 012301 (2007).

[25] J. B. Brask and A. S. Sørensen, *Memory imperfections in atomic-ensemble-based quantum repeaters,* Phys. Rev. A **78**, 012350 (2008).

[26] N. Sangouard, C. Simon, H. de Riedmatten, and N. Gisin, *Quantum repeaters based on atomic ensembles and linear optics,* Rev. Mod. Phys. **83**, 33 (2011).

[27] P. van Loock, W. Alt, C. Becher, O. Benson, H. Boche, C. Deppe, J. Eschner, S. Höfling, D. Meschede, P. Michler, F. Schmidt, and H. Weinfurter, *Extending quantum links: Modules for fiber- and memory-based quantum repeaters,* arXiv:1912.10123 (2019), arXiv:1912.10123 .

[28] S. Abruzzo, S. Bratzik, N. K. Bernardes, H. Kampermann, P. van Loock, and D. Bruß, *Quantum repeaters and quantum key distribution: Analysis of secret-key rates,* Phys. Rev. A **87**, 052315 (2013).

[29] F. Kimiaee Asadi, N. Lauk, S. Wein, N. Sinclair, C. O'Brien, and C. Simon, *Quantum repeaters with individual rare-earth ions at telecommunication wavelengths,* Quantum **2**, 93 (2018).

[30] L. Hartmann, B. Kraus, H.-J. Briegel, and W. Dür, *Role of memory errors in quantum repeaters,* Phys. Rev. A **75**, 032310 (2007).

[31] F. Schmidt and P. van Loock, *Memory-assisted long-distance phase-matching quantum key distribution,* arXiv preprint arXiv:1910:03333 (2019), arXiv:1910.03333 .

[32] F. Rozpędek, K. Goodenough, J. Ribeiro, N. Kalb, V. C. Vivoli, A. Reiserer, R. Hanson, S. Wehner, and D. Elkouss, *Parameter regimes for a single sequential quantum repeater,* Quantum Science and Technology (2018).

[33] E. Shchukin, F. Schmidt, and P. van Loock, *Waiting time in quantum repeaters with probabilistic entanglement swapping,* Phys. Rev. A **100**, 032322 (2019).

[34] V. V. Kuzmin, D. V. Vasilyev, N. Sangouard, W. Dür, and C. A. Muschik, *Scalable repeater architectures for multi-party states,* npj Quantum Information **5**, 115 (2019).

[35] V. V. Kuzmin and D. V. Vasilyev, *Diagrammatic technique for simulation of large-scale quantum repeater networks with dissipating quantum memories,* Physical Review A **103**, 032618 (2021).

[36] S. Santra, L. Jiang, and V. S. Malinovsky, *Quantum repeater architecture with hierarchically optimized memory buffer times,* Quantum Science and Technology **4**, 025010 (2019).

[37] K. Goodenough, D. Elkouss, and S. Wehner, *Optimising repeater schemes for the quantum internet,* arXiv preprint arXiv:2006.12221 (2020).

[38] S. E. Vinay and P. Kok, *Practical repeaters for ultralong-distance quantum communication,* Phys. Rev. A **95**, 052336 (2017).

[39] N. K. Bernardes, L. Praxmeyer, and P. van Loock, *Rate analysis for a hybrid quantum repeater,* Phys. Rev. A **83**, 012323 (2011).

[40] L. Praxmeyer, *Reposition time in probabilistic imperfect memories,* arXiv preprint arXiv:1309.3407 (2013), arXiv:1309.3407 .

[41] S. Khatri, C. T. Matyas, A. U. Siddiqui, and J. P. Dowling, *Practical figures of merit and thresholds for entanglement distribution in quantum networks,* Phys. Rev. Research **1**, 023032 (2019).

[42] S. E. Vinay and P. Kok, *Statistical analysis of quantum-entangled-network generation,* Phys. Rev. A **99**, 042313 (2019).

[43] G. Vardoyan, S. Guha, P. Nain, and D. Towsley, *On the capacity region of bipartite and tripartite entanglement switching,* arXiv:1901.06786 (2019).

[44] G. Vardoyan, S. Guha, P. Nain, and D. Towsley, *On the stochastic analysis of a quantum entanglement switch,* SIGMETRICS Perform. Eval. Rev. **47**, 27 (2019).

[45] S. Khatri, *Policies for elementary link generation in quantum networks,* arXiv preprint arXiv:2007.03193 (2020).

[46] J. Barry, D. T. Barry, and S. Aaronson, *Quantum partially observable markov decision processes,* Physical Review A **90**, 032311 (2014).

[47] S. Brand, T. Coopmans, and D. Elkouss, *Efficient computation of the waiting time and fidelity in quantum repeater chains,* IEEE Journal on Selected Areas in Communications **38**, 619 (2020).

[48] B. Li, T. Coopmans, and D. Elkouss, *Efficient optimization of cutoffs in quantum repeater chains,* IEEE Transactions on Quantum Engineering **2**, 1 (2021).

[49] S. Muralidharan, L. Li, J. Kim, N. Lütkenhaus, M. D. Lukin, and L. Jiang, *Optimal architectures for long distance quantum communication,* Scientific Reports **6**, 20463 EP (2016), article.

[50] L. Jiang, J. M. Taylor, K. Nemoto, W. J. Munro, R. Van Meter, and M. D. Lukin, *Quantum repeater with encoding,* Physical Review A **79**, 032325 (2009).

[51] W. Munro, K. Harrison, A. Stephens, S. Devitt, and K. Nemoto, *From quantum multiplexing to high-performance quantum networking,* Nature Photonics **4**, 792 (2010).

[52] Y. Li, S. D. Barrett, T. M. Stace, and S. C. Benjamin, *Long range failure-tolerant entanglement distribution,* New Journal of Physics **15**, 023012 (2013).

[53] P. Mazurek, A. Grudka, M. Horodecki, P. Horodecki, J. Łodyga, Ł. Pankowski, and A. Przysiężna, *Long-distance quantum communication over noisy networks without long-time quantum memory,* Physical Review A **90**, 062311 (2014).

[54] E. Knill and R. Laflamme, *Concatenated quantum codes,* arXiv preprint quant-ph/9608012 (1996).

[55] A. G. Fowler, D. S. Wang, C. D. Hill, T. D. Ladd, R. Van Meter, and L. C. Hollenberg, *Surface code quantum communication,* Physical Review Letters **104**, 180503 (2010).

[56] W. J. Munro, A. M. Stephens, S. J. Devitt, K. A. Harrison, and K. Nemoto, *Quantum communication without the necessity of quantum memories,* Nature Photonics **6**, 777 (2012).

[57] K. Azuma, K. Tamaki, and H.-K. Lo, *All-photonic quantum repeaters,* Nature Communications **6**, 6787 (2015).

[58] S. Muralidharan, J. Kim, N. Lütkenhaus, M. D. Lukin, and L. Jiang, *Ultrafast and fault-tolerant quantum communication across long distances,* Phys. Rev. Lett. **112**, 250501 (2014).

[59] A. N. Glaudell, E. Waks, and J. M. Taylor, *Serialized quantum error correction protocol for high-bandwidth quantum repeaters,* New Journal of Physics **18**, 093008 (2016).

[60] F. Ewert, M. Bergmann, and P. van Loock, *Ultrafast long-distance quantum communication with static linear optics,* Phys. Rev. Lett. **117**, 210501 (2016).

[61] F. Ewert and P. van Loock, *Ultrafast fault-tolerant long-distance quantum communication with static linear optics,* Phys. Rev. A **95**, 012327 (2017).

[62] M. Pant, H. Krovi, D. Englund, and S. Guha, *Rate-distance tradeoff and resource costs for all-optical quantum repeaters,* Physical Review A **95**, 012304 (2017).

[63] S.-W. Lee, T. C. Ralph, and H. Jeong, *Fundamental building block for all-optical scalable quantum networks,* Phys. Rev. A **100**, 052303 (2019).

[64] J. Borregaard, H. Pichler, T. Schröder, M. D. Lukin, P. Lodahl, and A. S. Sørensen, *One-way quantum repeater based on near-deterministic photon-emitter interfaces,* Physical Review X **10**, 021071 (2020).

5

EFFICIENT COMPUTATION OF THE WAITING TIME AND FIDELITY IN QUANTUM REPEATER CHAINS

In this chapter, we provide two efficient algorithms for determining the generation time and fidelity of the first generated entangled pair between the end nodes of a quantum repeater chain. The runtime of the algorithms increases polynomially with the number of segments of the chain, which improves upon the exponential runtime of existing algorithms. Our first algorithm is probabilistic and can analyse refined versions of repeater chain protocols which include intermediate entanglement distillation. Our second algorithm computes the waiting time distribution up to a pre-specified truncation time, has faster runtime than the first one and is moreover exact up to machine precision (we will give an even faster version of the algorithm in Chapter 6). Using our proof-of-principle implementation, we are able to analyse repeater chains of thousands of segments for some parameter regimes. The algorithms thus serve as useful tools for the analysis of large quantum repeater chain protocols and topologies of the future quantum internet.

This chapter has been published, with minor changes, as: S. Brand*, T. Coopmans* and D. Elkouss, *Efficient computation of the waiting time and fidelity in quantum repeater chains*, IEEE Journal on Selected Areas in Communications 38, 619 (2020), where * denotes equally contributing authors.

In Chapter 3, we have seen that the quantum internet enables many applications that are impossible with its classical counterpart. One of the key elements to enable the applications is the distribution of entanglement between remote parties. In this chapter, we aim at fully characterising the behaviour of an important class of entanglement distribution protocols over repeater chains as a tool for the analysis of quantum networks.

A large number of quantum repeater protocols have been proposed [1–22] and to a large extent they can be classified [17, 23] depending on whether or not they use error correction codes to handle these issues. In the absence of coding, losses can be dealt with via heralded entanglement generation and errors via entanglement distillation [24–31]. In this chapter, we will focus our interest in this type of protocols as their implementation is closer to experimental reach.

Existing analytical work is mostly aimed at estimating the mean waiting time or fidelity (see also [12, 32, 33] for other figures of merit). Some of this work builds on an approximation of the mean waiting time under the small-probability assumption [6, 13, 21, 34], while for a small number of segments or for some protocols it is possible to compute the waiting time probability distribution exactly [2, 20, 33, 35, 36]. However, depending on the application different statistics become relevant. For instance, in the presence of decoherence, one is also interested in the variations around the mean. In order to connect two segments via an intermediate repeater, both segments need to produce an entangled pair. When the first pair in one of the segments is ready, it has to wait until the second segment finalises, and it decoheres while waiting. In this context, one may need to discard the entanglement after some maximum amount of time [33, 37, 38]. Entanglement is also used as a resource for implementing non-local gates in distributed quantum computers [39]. In this context, it is relevant to understand the time it takes to generate a pair of the desired quality with probability larger than some threshold, i.e. in the cumulative distribution. Here, we undertake the problem of fully characterising the probability distribution of the waiting time and the associated fidelity to the maximally entangled state.

An algorithm to characterise the full waiting time distribution was first obtained in [35] using Markov chain theory (see also sec. 4.3.1 in Chapter 4). Its runtime scales with the number of vertices in the Markov chain, which grows exponentially with the number of repeater segments. In more recent work, Vinay and Kok show how to improve the runtime using results from complex analysis [36]. However, this method still remains exponential in the number of repeater segments. Here, we provide two algorithms for computing the full distribution of the waiting time and fidelity following the same model as in [35]. Both algorithms are polynomial in the number of segments. Our main tool is the description of the waiting time and fidelity of the first produced end-to-end link as a recursively defined random variable, in line with the recursive structure of the repeater chain protocol. The first algorithm is a Monte Carlo algorithm which samples from this random variable, whereas our second algorithm is deterministic and computes the waiting time distribution up to a pre-specified truncation time. The power of the former algorithm lies in its extendability: it can be used to analyse refined versions of repeater chain protocols which include intermediate entanglement distillation. The second algorithm is faster and exact: it computes the probability distribution of the waiting time and corresponding fidelity up to a pre-specified truncation point where the only source

of error is machine precision. The speed of our algorithms allows us to analyse repeater chains with more than a thousand segments for some parameter regimes.

The organisation of this chapter is as follows. In sec. 5.1, we introduce notation and the family of repeater protocols under study. Then, in sec. 5.2, we recursively define the waiting time and fidelity of the generation of a single entangled pair between the end nodes as a random variable. In sec. 5.3, we provide the two algorithms for computing the probability distribution of this random variable. We show in sec. 5.4 how to calculate tighter numerical bounds on the mean waiting time than known in previous work. Numerical results are given in sec. 5.5. In Section 5.6 we discuss the results obtained and provide an outlook for future research.

5.1. PRELIMINARIES

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

5.1.1. QUANTUM REPEATER CHAINS

In the family of repeater chain protocols we study in this chapter, nodes are able to perform the following three actions: generate fresh entanglement with adjacent nodes, transform short-range entanglement into long-range entanglement by means of entanglement swapping, and increase the quality of links through entanglement distillation. We refer to sec. 3.2 in Chapter 3 for a more in-detail description of these three actions. We model the entanglement swap and entanglement distillation as operations which succeed probabilistically: in the case of failure, both involved entangled pairs are lost.

The repeater chain protocols we study in this chapter are the `tree-shaped`-type protocols, which are all based on the seminal work of Briegel et al. [7]. Their scheme was designed for a repeater chain of $N = W^n$ segments with $n \in \{1,2,\ldots\}$; for simplicity, we assume $W = 2$ here. We distinguish between two versions of the protocol: NESTED-SWAP-ONLY and d-NESTED-WITH-DISTILL. In NESTED-SWAP-ONLY, nodes generate elementary entanglement and transform it into end-to-end entanglement by means of entanglement swaps in a particular order explained below. The d-NESTED-WITH-DISTILL scheme is identical to the NESTED-SWAP-ONLY version except for the fact that every n-hop link is produced 2^d times for some integer $d \geq 1$ and then turned into a single high-quality link by performing entanglement distillation multiple times (more details below). We introduced these schemes briefly in Chapter 3 and will describe them in more detail here.

We start by explaining how the NESTED-SWAP-ONLY protocol works for two segments and subsequently generalise to 2^n segments. On a chain of two segments, the NESTED-SWAP-ONLY scheme starts with both end nodes generating a single entangled pair with the repeater node (fig. 5.1(a)). Once a link is generated, the two involved nodes store the state in memory. As soon as both pairs have been produced, the repeater node performs an entanglement swap on the two qubits it holds, which probabilistically produces a 2-hop link between the end nodes. In the case that the entanglement swap did not succeed, both end nodes will be notified of the failure by the heralding message from the repeater node and subsequently each restart generation of the single-hop entangle-

ment.

In the generalisation to repeater chains of 2^n segments (fig. 5.1(b) and (c)), the NESTED-SWAP-ONLY scheme starts with the two-segment repeater scheme as explained above on the first and second segment, on the third and fourth, and so on until segments $2^n - 1$ and 2^n. Approximately half of the intermediate nodes are thus involved in two instances of the scheme; as soon as both instances have finished generating 2-hop entanglement, the node will perform an entanglement swap to generate a single 4-hop entangled link. In case the entanglement swap fails, all nodes under the span of the 4 hops will start to generate single-hop entanglement again as part of the two-segment scheme. In general, to produce entanglement that spans 2^ℓ hops, the node that is located precisely in the middle of this span will wait for the production of two $2^{\ell-1}$-hop links and then perform an entanglement swap (see also fig. 5.1(c)). The failure of this swap requires to regenerate both $2^{\ell-1}$-hop links. We refer to $\ell \in \{0, 1, \ldots, n\}$ as the 'nesting level' of the protocol, such that single-hop entanglement is produced at the base level $\ell = 0$ and the entanglement swap at level $\ell \geq 1$ transforms two $2^{\ell-1}$-hop links into a single 2^ℓ-hop entangled pair.

In addition to the steps of the NESTED-SWAP-ONLY version as described above, the original proposal of Briegel et al. included entanglement distillation in order to increase the quality of the input links to each entanglement swap. In this work, we specifically define a version of the repeater protocol, denoted by d-NESTED-WITH-DISTILL for some $d \geq 1$, where d rounds of distillation are performed at every nesting level. That is, instead of a single link, 2^d links are generated at every nesting level. These links are subsequently used as input to a recurrence distillation scheme: our description of this scheme follows the review work by Dür and Briegel [24]. In the first step of the recurrence protocol, the 2^d links are split up in pairs and used as input to entanglement distillation, which produces 2^{d-1} entangled pairs of higher quality. This process is repeated with the remaining links until only a single link is left, which is then used by the repeater node as input link to the entanglement swap. Rather than waiting for all 2^d links to have been generated before performing the first distillation step, the protocol performs the entanglement distillation as soon as two links are available. The failure of a distillation step requires the two involved nodes to regenerate the links. For $d = 0$, the d-NESTED-WITH-DISTILL scheme is identical to NESTED-SWAP-ONLY since no distillation is performed.

Generating, distilling and swapping entanglement can in general all be probabilistic operations, which makes the total time it takes to distribute a single entangled pair between the end nodes of a repeater chain a random variable. We use the notation T_n to refer to the waiting time until a single end-to-end link in a 2^n-segment repeater chain has been produced. By F_n we refer to the link's fidelity, a measure of the quality of the state (see sec. 5.1.2). Every time the quantities T_n and F_n are used in this chapter we explicitly state whether they correspond to the waiting time of the NESTED-SWAP-ONLY version or the d-NESTED-WITH-DISTILL version for given d. The goal of this chapter is to find the joint probability distribution of T_n and F_n for both schemes.

5.1.2. MODEL

In the quantum repeater protocols we study (see sec. 5.1.1), nodes can generate, store, distill and swap entangled links. We show here how we model each of these four opera-

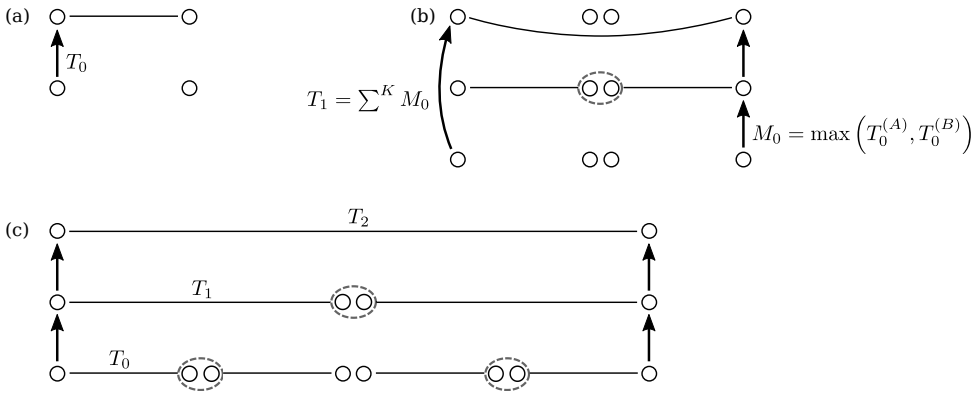


Figure 5.1: The NESTED-SWAP-ONLY version of the BDCZ protocol [7] and its completion time T_n as a random variable, where 2^n is the number of segments in the repeater chain (see also sec. 5.1.1). (a) For two segments, T_0 represents the waiting time for the generation of a single link between two nodes without any intermediate repeater nodes. (b) Nested level structure of the protocol over $2^1 = 2$ segments. The production of entanglement over two segments first requires the generation of two links, each of which spans a single segment. The total time until both links have been generated equals M_0, the maximum of their individual generation times $T_0^{(A)}$ and $T_0^{(B)}$, which are independent random variables that are identically distributed (i.i.d.). Once the two links have been produced, a probabilistic entanglement swap is performed at both links. Failure of the entanglement swap requires the two single-hop links to be regenerated, each of which adds to the total waiting time T_1. The random variable K corresponds to the number of failing entanglement swaps up to and including the first successful swap. In this chapter we assume that K follows a geometric distribution with parameter p_{swap} (see sec. 5.2.1). (c) A link that spans 2^n segments is produced in a nested fashion, where at each nesting level two links are produced and subsequently swapped.

tions.

For the generation of single-hop entanglement between two adjacent nodes, we choose generation schemes which perform heralded attempts of fixed duration L/c where c is the speed of light and L is the distance over which entanglement is generated [23]. In this chapter we study the topology where all nodes are equally spaced with distance $L = L_0$.

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

The first step of the entanglement swapping, the Bell-state measurement, is modelled as a probabilistic operation with fixed success probability $0 < p_{\text{swap}} \leq 1$ which is identical for all nodes. This success probability is independent of the state of the qubits that it acts upon. For simplicity, we assume that the duration of the Bell-state measurement is negligible. The Bell-state measurement is followed by a classical heralding signal to notify the nodes holding the other sides of the pair whether the Bell-state measurement was successful. An entanglement swap on two 2^n-hop links thus takes $2^n \cdot L_0/c$ time. Although our algorithms can account for this communication time (see sec. 5.2.3), we will assume this time to be negligible in most of this work.

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

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

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

We assume that the single-hop entangled states that are generated are two-qubit Werner states parameterised by a single parameter $0 \leq w_0 \leq 1$ [40]:

$$\rho(w_0) = w_0 |\Phi^+\rangle\langle\Phi^+| + (1-w_0)\frac{\mathbb{1}_4}{4} \tag{5.1}$$

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

$$F\left(\rho(w), |\Phi^+\rangle\langle\Phi^+|\right) = \langle\Phi^+|\rho(w)|\Phi^+\rangle = (1+3w)/4. \tag{5.2}$$

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

$$w_{\text{decayed}} = w \cdot e^{-\Delta t / T_{\text{coh}}} \tag{5.3}$$

where T_{coh} is the joint coherence time of the two quantum memories holding the qubits. We assume that access to a quantum memory is on-demand, i.e. the quantum states can be stored and retrieved at any time and moreover there is no fidelity penalty associated with such memory access.

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

$$\rho_{\text{swap}} = \rho(w \cdot w'). \tag{5.4}$$

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

As base for entanglement distillation, we use the BBPSSW-scheme [26]. We modify it slightly by bringing the output state back into Werner form. The last step does not change the Bell-state fidelity of the output state. If two Werner states with parameters w_A and w_B are used as input to entanglement distillation, both the output Werner parameter w_{dist} and the success probability p_{dist} depend on the Werner parameters w_A and w_B of the states it acts upon (see appendix 5.7.1):

$$w_{\text{dist}}(w_A, w_B) = \frac{1 + w_A + w_B + 5w_A w_B}{6p_{\text{dist}}} - \frac{1}{3} \tag{5.5}$$

$$p_{\text{dist}}(w_A, w_B) = (1 + w_A w_B)/2. \tag{5.6}$$

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

5.1.3. NOTATION: RANDOM VARIABLES

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

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

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

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

An example of such a function is addition. Define $Z := X + Y$ where X and Y are independent, then the probability distribution p_Z of Z is given by the convolution of the distributions p_X and p_Y, denoted as $p_Z = p_X * p_Y$, which means [41]

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

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

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

In this section, we derive expressions for the waiting time and fidelity of the first generated end-to-end link in the NESTED-SWAP-ONLY repeater chain protocol. First, in sec. 5.2.1, we derive a recursive definition for the random variable T_n, which represents the waiting time in a 2^n-segment repeater chain. Section 5.2.2 is devoted to extending this definition to the Werner parameter W_n of the pair, which stands in one-to-one correspondence to its fidelity F_n using eq. (5.2):

$$F_n = (1 + 3W_n)/4. \tag{5.7}$$

In sec. 5.2.3, we show how to include the communication time after the entanglement swap and in sec. 5.2.4, we extend the analysis of the waiting time and Werner parameter in the NESTED-SWAP-ONLY protocol to the d-NESTED-WITH-DISTILL scheme.

5.2.1. RECURSIVE EXPRESSION FOR THE WAITING TIME IN THE NESTED-SWAP-ONLY PROTOCOL

In the following, we will derive a recursive expression for the waiting time T_n of a NESTED-SWAP-ONLY repeater chain of 2^n segments (see also fig. 5.1).

Before stating the expression, let us note that all three operations in the repeater chain protocols we study in this work, entanglement generation over a single hop, distillation and swapping, take a duration that is a multiple of L_0/c, the time to send information over a single segment (see sec. 5.1.2 for our assumptions on the duration of operations). For this reason, it is common to denote the waiting time in discrete units of L_0/c, which is a convention we comply with for T_n.

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

Waiting time in the NESTED-SWAP-ONLY protocol

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

$$T_{n+1} := \sum_{j=1}^{K_n} M_n^{(j)} \tag{5.8}$$

where M_n is an auxiliary random variable given by

$$M_n := g_{\text{T}}\left(T_n^{(A)}, T_n^{(B)}\right) \tag{5.9}$$

and the function g_{T} is defined as

$$g_{\text{T}}(t_A, t_B) := \max\{t_A, t_B\}. \tag{5.10}$$

The sum in eq. (5.8) is taken over the number of entanglement swaps K_n until the first success, which is geometrically distributed with parameter p_{swap} for every n. See fig. 5.1 for a depiction of T_n and K_n.

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

We start with the base case T_0, the waiting time for the generation of elementary entanglement. Since we model the generation of single-hop entanglement by attempts which succeed with a fixed probability p_{gen} (see sec. 5.1.2), the waiting time T_0 is a discrete random variable (in units of L_0/c) which follows a geometric distribution with probability distribution given by $\Pr(T_0 = t) = p_{\text{gen}}(1-p_{\text{gen}})^{t-1}$ for $t \in \{1,2,3,\ldots\}$. For what follows, it will be more convenient to specify T_0 by its cumulative distribution function

$$\Pr(T_0 \leq t) = 1 - (1 - p_{\text{gen}})^t. \tag{5.11}$$

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

$$\begin{aligned} \Pr(M_n \leq t) &= \Pr\left(T_n^{(A)} \leq t \text{ and } T_n^{(B)} \leq t\right) \\ &= \Pr(T_n \leq t)^2 \end{aligned} \tag{5.12}$$

where the last equality follows from the fact that $T_n^{(A)}, T_n^{(B)}$ and T_n are pairwise i.i.d. Since we assume that both the duration of the Bell-state measurement and the communication time of the heralding signal after the entanglement swap are negligible (see sec. 5.1.2), M_n is also the time at which the entanglement swap ends. We will drop the assumption on negligible communication time in sec. 5.2.3.

In order to find the relation between M_n and T_{n+1}, first note that the number of swaps K_n at level n until the first successful swap follows a geometric distribution with parameter p_{swap}. This is a direct consequence of our choice to model the success probability p_{swap} to be independent of the state of the two input links (see sec. 5.1.2). Next, recall that the two input links of a failing entanglement swap are lost and need to be regenerated. The regeneration of fresh entanglement after each failing entanglement swap adds to the waiting time. Thus, T_{n+1} is a *compound random variable*: it is the sum of K_n copies of M_n. Since the number of entanglement swaps K_n is geometrically distributed, we say that T_{n+1} is a *geometric compound sum* of K_n copies of M_n. To be precise, we write

$$T_{n+1} = \sum_{k=1}^{K_n} M_n^{(k)} \tag{5.13}$$

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

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

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

The waiting time T_n is the same quantity as was studied by Shchukin et al. [35]. Indeed, in sec. 5.5, we show that our algorithms for computing the probability distribution of T_n recover their numerical results.

5.2.2. JOINT RECURSIVE EXPRESSION OF WAITING TIME AND WERNER PARAMETER FOR THE NESTED-SWAP-ONLY PROTOCOL

In this section, we extend the expression of the waiting time for the first end-to-end link produced using the NESTED-SWAP-ONLY protocol with the link's state. To be precise, we give a recursive expression for the waiting time T_n and Werner parameter W_n of this state, which is well-defined since all states that the NESTED-SWAP-ONLY repeater

chain protocol holds at any time during its execution are Werner states. The latter statement is a direct consequence of the fact that in our modelling, all operations in the NESTED-SWAP-ONLY protocol only output Werner states: we choose to model the generated single-hop entanglement as Werner states and furthermore the class of Werner states is invariant under memory errors and entanglement swaps (see sec. 5.1.2). The fidelity F_n of the first end-to-end state on 2^n segments can be computed from its Werner parameter using eq. (5.7).

We express the waiting time and Werner parameter as a joint random variable (T_n, W_n). Describing the two as a tuple allows us to capture the fact that the Werner parameter of a link depends on the time it was produced at. In sec. 5.2.1, we found that the failure of multiple swapping attempts corresponds to the sum of their waiting times. In order to extend this description to the tuple of waiting time and Werner parameter, we define the *forgetting sum* $\widehat{\sum}$ on sequences of tuples $\{(x_j, y_j) | 1 \leq j \leq m\}$ for some $m \in \{1, 2, \ldots\}$ as

$$\widehat{\sum}_{j=1}^{m} (x_j, y_j) := \left(\left[\sum_{j=1}^{m} x_j \right], y_m \right). \tag{5.14}$$

In analogy to the geometric compound sum from eq. (5.13), we define the *geometric compound forgetting sum* $(X', Y') := \widehat{\sum}_{j=1}^{K} (X, Y)$, which formally means

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

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

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

Waiting time and Werner parameter in the NESTED-SWAP-ONLY protocol

The joint random variable (T_n, W_n) is defined as follows. The waiting time T_0 is the same as in sec. 5.2.1 and $\Pr(W_0 = w_0) = 1$ where $w_0 \in [0,1]$ is some pre-specified constant that determines the state of the single-hop entanglement that is produced between adjacent nodes. At the recursive step, the waiting time and Werner parameter are given by the geometric compound forgetting sum

$$(T_{n+1}, W_{n+1}) := \widehat{\sum}_{k=1}^{K_n} (M_n, V_n)^{(k)} \tag{5.15}$$

where, as in sec. 5.2.1, K_n follows a geometric distribution with parameter p_{swap}. The auxiliary joint random variable (M_n, V_n) is defined as

$$(M_n, V_n) := g\left((T_n, W_n)^{(A)}, (T_n, W_n)^{(B)}\right). \tag{5.16}$$

The function g is given by

$$\begin{aligned} &g((t_A, w_A), (t_B, w_B)) := \\ &\quad (g_{\text{T}}(t_A, t_B), g_{\text{W}}((t_A, w_A), (t_B, w_B))) \end{aligned} \tag{5.17}$$

where g_{T} is defined in eq. 5.10 and

$$g_{\text{W}}((t_A, w_A), (t_B, w_B)) := w_A \cdot w_B \cdot e^{-|t_A - t_B|/T_{\text{coh}}} \tag{5.18}$$

with T_{coh} the quantum memory coherence time as described in sec. 5.1.2.

We now explain the above expressions. For a single segment ($n = 0$), the waiting time and Werner parameter are uncorrelated because we model the attempts at generating single-hop entanglement to be independent and to each take equally long (see sec. 5.1.2). At the recursive step, an entanglement swap which produces 2^{n+1}-hop entanglement requires the generation of two 2^n-hop links. The expression for the waiting time T_{n+1} is identical to eq. (5.8) in sec. 5.2.1. In order to argue that eq. (5.15) also gives the correct expression for W_{n+1}, we first show that the Werner parameter of the output link of an entanglement swap is given by V_n in eq. (5.16), provided the swap succeeded. Since M_n as defined in eq. (5.16) is identical to its expression in eq. (5.9) in sec. 5.2.1, we only need to argue why g_{W} in eq. (5.18) correctly computes the Werner parameter of the output link after an entanglement swap.

In order to do so, denote by A and B the input links to the entanglement swap and denote by (t_A, w_A) and (t_B, w_B) their respective delivery times and Werner parameters. Without loss of generality, choose $t_A \geq t_B$, i.e. link A is produced after link B. Link A is produced last, so the entanglement swap will be performed directly after its generation and hence link A will enter the entanglement swap with Werner parameter w_A. Link B is produced earliest and will therefore decohere until production of link A. It follows from eq. (5.3) that B's Werner parameter immediately before the swap equals

$$w'_B = w_B \cdot e^{-|t_A - t_B|/T_{\text{coh}}}. \tag{5.19}$$

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

hop state with Werner parameter

$$w_A \cdot w'_B \tag{5.20}$$

as in eq. (5.4), provided the swap is successful. Combining eqs. (5.19) and (5.20) yields the definition of g_W in eq. (5.18).

Note that in the definition of g_W in eq. (5.18) we used the same assumption on the duration of the entanglement swap as in sec. 5.2.1, i.e. that both the Bell-state measurement and the subsequent communication time are negligible (see also sec. 5.1.2). This implies that V_n in eq. (5.16) expresses the Werner parameter of the produced 2^{n+1}-hop link in case the swap is successful. We treat the case of nonzero communication time in sec. 5.2.3.

The last step in finding the Werner parameter W_{n+1} in eq. (5.15) is to bridge the gap with (M_n, V_n) from eq. (5.16). If the entanglement swap fails, then the 2^{n+1}-hop link with its Werner parameter in eq. (5.20) will never be produced since both initial 2^n-hop entangled pairs are lost. Instead, two fresh 2^n-hop links will be generated. In order to find how the Werner parameter on level $n+1$ is expressed as a function of the waiting times and Werner parameters at level n, consider a sequence (m_j, v_j) of waiting times m_j and Werner parameters v_j, where j runs from 1 to the first successful swap k. The m_j correspond to the waiting time until the end of the entanglement swap that transforms two 2^n-hop links into a single 2^{n+1}-hop link and the v_j to the output link's Werner parameter if the swap were successful. We have found in sec. 5.2.1 that the total waiting time is given by $\sum_{j=1}^{k} m_j$, the sum of the duration of the production of the lost pairs (see eq. (5.8)). Note, however, that the Werner parameter of the 2^{n+1}-hop link is only influenced by the links that the *successful* entanglement swap acted upon. Since the entanglement swaps are performed until the first successful one, the output link is the last produced link and therefore its Werner parameter equals v_k. We thus find that the waiting time t_{final} of the first 2^{n+1}-hop link and its Werner parameter w_{final} are given by the forgetting sum from eq. (5.14):

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

Taking into account that the number of swaps k that need to be performed until the first successful one is an instance of the random variable K_n, we arrive at the full recursive expression for the waiting time and Werner parameter at level $n+1$ as given in eq. (5.15).

It is not hard to see that the projection $(T_n, W_n) \mapsto T_n$ recovers the definition of waiting time from 5.2.1. Indeed, following the recursive definition of (T_n, W_n) in eqs. (5.15)-(5.18), the waiting time T_n is not affected by the Werner parameters W_ℓ at lower nesting levels $\ell < n$.

5.2.3. INCLUDING COMMUNICATION TIME

While deriving the expressions for waiting time and Werner parameter of the first produced end-to-end link in secs. 5.2.1 and 5.2.2, we have explicitly assumed that the total time the entanglement swap takes is negligible. Here, we include the communication time of the heralding signal from the entanglement swap into the expressions for M_n and V_n (eqs. (5.9) and (5.16)), which represent the waiting time and Werner parameter

directly after the entanglement swap if it were successful. This communication time equals 2^n time steps (in units of L_0/c) for a swap that transforms two 2^n-hop links into a single 2^{n+1}-hop link (see sec. 5.1.2). The expressions for M_n and V_n are modified by replacing g_T in eq. 5.10 by

$$g_T^n(t_A, t_B) := g_T(t_A, t_B) + 2^n. \tag{5.21}$$

and replacing g_W from eq. (5.18) by

$$\begin{aligned} &g_W^n((t_A, w_A), (t_B, w_B)) \\ &\quad := g_W((t_A, w_A), (t_B, w_B)) \cdot e^{-2^n/T_{\text{coh}}}. \end{aligned} \tag{5.22}$$

Equation (5.21) expresses that the entanglement swap takes 2^n timesteps longer, while eq. (5.22) captures the decoherence of the state during the communication time of the entanglement swap, following eq. (5.3).

5.2.4. WAITING TIME AND WERNER PARAMETER FOR THE d-NESTED-WITH-DISTILL PROTOCOL

In this section, we sketch how to extend the expression of the waiting time T_n and Werner parameter W_n from secs. 5.2.1-5.2.3 to the case of the d-NESTED-WITH-DISTILL repeater protocol presented in sec. 5.1.1. Recall that the d-NESTED-WITH-DISTILL protocol is identical to the NESTED-SWAP-ONLY protocol except for the fact that each entanglement swap is performed on the output of a recurrence distillation scheme with d nesting levels. By a d'-distilled 2^n-hop link we denote a 2^n-hop link which is the result of successful entanglement distillation on two $(d'-1)$-distilled 2^n-hop links and by a 0-distilled 2^n-hop link we mean a link that is the result of a successful entanglement swap on two 2^n-hop links. Thus, every entanglement swap in the d-NESTED-WITH-DISTILL protocol is performed on d-distilled links only.

Note that at every level of the nested swapping, there are d levels of nested distillation. To tackle the 'double nesting' we modify the waiting time in the NESTED-SWAP-ONLY protocol by splitting up the tuple of random variables (T_n, W_n) in eq. (5.15), which represents the waiting time and Werner parameter at level n, into $d+1$ tuples of random variables $(T_n^{d'}, W_n^{d'})$ for $d' \in \{0, 1, \ldots, d\}$. The random variable $T_n^{d'}$ corresponds to the waiting time until the end of the first successful distillation attempt on two d'-distilled 2^n-hop links, and $W_n^{d'}$ to the link's Werner parameter.

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

If we fix the nesting level n, we can straightforwardly apply the analysis of sec. 5.2.2 to the nested distillation. First, we define $(M_n^{d'}, V_n^{d'})$, which characterises a link after a single distillation attempt on two 2^n-hop d'-distilled links in case the attempt is successful. This joint random variable is the analogue of (M_n, V_n) from eq. (5.16), which has the same interpretation but in this case for a swapping attempt. The analysis resulting in eq. (5.16) carries over and yields

$$(M_n^{d'}, V_n^{d'}) := g_D\left((T_n^{d'}, W_n^{d'})^{(A)}, (T_n^{d'}, W_n^{d'})^{(B)}\right). \tag{5.23}$$

where g_{D} is the analogue of g in eq. (5.17) and describes how two input links are transformed into one high-quality link by a successful distillation step:

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

where

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

and w_{dist} is given in eq. (5.5). The function g_{D} outputs a tuple of waiting time and Werner parameter of the output state after distillation. The waiting time requires two links to be generated and is thus given by g_{T} in eq. (5.10). The Werner parameter equals the Werner parameter of distillation as given by w_{dist} in eq. (5.5) on the two input links, of which the earlier suffered decoherence as given in eq. (5.3).

The random variables $(T_n^{d'}, W_n^{d'})$ correspond to the waiting time and Werner parameter after the first successful distillation attempt on two d'-distilled 2^n-hop links, so in line with the analysis leading to eq. (5.15) we obtain

$$(T_n^{d'+1}, W_n^{d'+1}) = \widehat{\sum}_{j=1}^{\mathscr{D}_n^{d'}} \left(M_n^{d'}, V_n^{d'}\right)^{(j)}. \tag{5.24}$$

The random variable $\mathscr{D}_n^{d'}$ corresponds to the number of distillation attempts with two d'-distilled 2^n-hop links as input, up to and including the first successful attempt. It is the analogue of K_n in eq. (5.15), the number of swap attempts until the first success.

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

$$(T_n^0, W_n^0) = (T_n, W_n). \tag{5.25}$$

Second, since an entanglement swap takes as input two d-distilled links, we find that we should replace the definition of (M_n, V_n) in eq. (5.16) by

$$(M_n, V_n) = g\left(\left(T_n^d, W_n^d\right)^{(A)}, \left(T_n^d, W_n^d\right)^{(B)}\right). \tag{5.26}$$

where g is defined in eq. (5.17).

We finish this section by remarking that for the d-NESTED-WITH-DISTILL protocol, we cannot treat waiting time independently of the Werner parameter of the produced link, as we did for the NESTED-SWAP-ONLY scheme in sec. 5.2.1. The reason behind this is the following difference between the nested swaps and the nested distillation: in the former, the success probability p_{swap} and therefore the number of swaps K_n is independent of the time and state of the produced links, whereas the success probability of entanglement distillation is a function of their states (see eq. (5.6)). Consequently, the

summation bound $\mathscr{D}_n^{d'}$ and the Werner parameter $V_n^{d'}$ in the summands $\left(M_n^{d'}, V_n^{d'}\right)$ in eq. (5.24) are correlated. Therefore, both the waiting time and Werner parameter at any swapping level depend on both waiting time and Werner parameter at the levels below.

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

In this section, we present two algorithms for determining the probability distribution of the waiting time T_n and average Werner parameter W_n of the first end-to-end link produced by the repeater chain (see sec. 5.2). The first algorithm is a Monte Carlo algorithm which applies to both families of repeater chain protocols considered in this chapter: NESTED-SWAP-ONLY and d-NESTED-WITH-DISTILL. The second algorithm only applies to the NESTED-SWAP-ONLY protocol and is faster than the first. In Chapter 6 we will present a variant to the second algorithm which can also handle d-NESTED-WITH-DISTILL, and is moreover faster than the one presented here. We summarise the runtime of the different algorithms presented in this section in table 5.1.

5.3.1. FIRST ALGORITHM: MONTE CARLO SIMULATION

The first algorithm is a randomised function which produces a sample from the probability distribution of the joint random variable (T_n, W_n). By running the algorithm many times, sufficient statistics can be produced to reconstruct the distribution of the joint random variable up to arbitrary precision (see below for a rigorous statement). We first outline the algorithm that samples from the waiting time in the NESTED-SWAP-ONLY protocol following sec. 5.2.1, after which we show how to extend it to track the Werner parameter (sec. 5.2.2), how to include the communication time after a swap (sec. 5.2.3) and how to adjust it for the d-NESTED-WITH-DISTILL protocol (sec. 5.2.4). Pseudocode can be found in algorithm 1.

We start by explaining the Monte Carlo algorithm for the waiting time in the NESTED-SWAP-ONLY protocol. Let $s(X)$ denote a randomised function that yields a sample from the random variable X. We remark that if the cumulative distribution function of X is known, then sampling from X can be done efficiently using inverse transform sampling, which is a standard technique to produce a sample from an arbitrary distribution by evaluating its inverse cumulative distribution function on a sample from the uniform distribution on the interval $[0,1]$. We can thus construct the sampler from the waiting time for elementary entanglement, T_0, using the inverse of the cumulative distribution function of T_0 as given in eq. (5.11):

$$s(T_0) = \lceil \log_{(1-p)}(1 - s(U)) \rceil \tag{5.27}$$

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

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

Repeater chain protocol (sec. 5.1.1)		Markov-chain-approach [35] [36]	Algorithms in this work	
			Monte-Carlo	Deterministic
NESTED-SWAP-ONLY	Waiting time up to 99% of the cumulative probabilities	$\Theta\left(\exp\left(N\right)\right)$	$\mathcal{O}\left(\text{poly}\left(N\right)\right)$	$\Theta\left(\text{poly}\left(N\right)\right)$
	Waiting time up to fixed truncation time $t_{\text{trunc}} = 1000$	$\Theta\left(\exp\left(N\right)\right)$	$\mathcal{O}\left(\text{poly}\left(N\right)\right)$	$\Theta\left(\log\left(N\right)\right)$
	Fidelity	×	$\mathcal{O}\left(\text{poly}\left(N\right)\right)$	$\Theta\left(\text{poly}\left(N\right)\right)$
d-NESTED-WITH-DISTILL	Waiting time & fidelity	×	$\mathcal{O}\left(\text{poly}\left(N\right)\right)$	×

Table 5.1: The time complexity of the algorithms for computing waiting time and fidelity of entanglement distribution through repeater chains as presented in this chapter compared to existing algorithms. The algorithms have exponential (exp) or polynomial (poly) runtime in $N = 2^n$, the number of segments in the repeater chain, for $n \in \{1, 2, \ldots\}$. The Monte Carlo algorithm is a randomised algorithm; its presented runtime is the average runtime. The cross (×) indicates that the algorithm is not present.

calls itself:

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

From the recursive expression for the waiting time T_n in sec. 5.2.1 it now follows directly that we can construct a sampler from T_n for $n \geq 1$:

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

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

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

where g_{T} is defined in eq. (5.10).

Using the Dvoretzky-Kiefer-Wolfowitz inequality [42], we determine how many samples from (T_n, W_n) we need in order to obtain bounds on its cumulative probabilities. It follows from this inequality that if $q(t) := \Pr(T_n \leq t)$ denotes the cumulative probability function of the waiting time T_n and $q_m(t)$ the empirical cumulative probabilities after having drawn m samples, then the difference between q and q_m is bounded as

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

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

$$m = -\log(z/2)/(2\epsilon^2) \tag{5.28}$$

Let us emphasise that this number of samples is independent of any parameters of the repeater chain, for instance the number of segments, and thus its contribution to the runtime or space usage of the Monte Carlo algorithm is at most a multiplicative constant, independent of any such parameters.

Following sec. 5.2.2, we modify the Monte Carlo algorithm to also compute the Werner parameter of the sampled produced entangled pair (for pseudocode see algorithm 1). First note that the notation $s(X)$ which samples from a random variable X can also be applied to a joint random variable (X, Y), so that $s((X,Y))$ returns a tuple. We will now define a sampler $s((T_n, W_n))$ where (T_n, W_n) is the joint random variable representing waiting time and Werner parameter of a 2^n-segment NESTED-SWAP-ONLY repeater chain (see sec. 5.2.2). For this, we first need to adapt the sampler of the geometric compound sum s_{sum} to a sampler of the geometric compound forgetting sum (eq. (5.14)) by defining $\hat{s}_{\text{sum}}((X,Y),p)$ where X and Y are arbitrary random variables and $p \in [0,1]$ is the parameter of the geometric distribution:

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

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

A recursive definition of the joint sampling function from (T_n, W_n) follows directly from the joint expression for waiting time T_n and Werner parameter W_n in eqs. (5.15)-(5.18):

$$\begin{aligned} s((T_0,W_0)) &= (s(T_0), w_0) \\ s((T_n,W_n)) &= \hat{s}_{\text{sum}}((M_n,V_n), p_{\text{swap}}) \\ s((M_n,V_n)) &= g(s(T_{n-1},W_{n-1}), s(T_{n-1},W_{n-1})) \end{aligned} \tag{5.29}$$

where w_0 is the Werner parameter of each single-hop link at the time it is produced (see sec. 5.1.2) and the function g is defined in eq. (5.17). In this pseudocode for this Monte Carlo algorithm in algorithm 1, the sampler $s(T_n, W_n)$ is denoted by sample_swap.

Since the expression for (T_n, W_n) from sec. 5.2.2 assumes that the communication time for the heralding signal after the entanglement swap takes negligible time, it is not included in the Monte Carlo algorithm above. Fortunately, the adaptation to include this communication time as in sec. 5.2.3 directly carries over to the Monte Carlo algorithm by replacing g in eq. (5.29) with

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

where g^n_{T} and g^n_{W} are defined in eqs. (5.21) and (5.22).

The time complexity of the Monte Carlo algorithm is a random variable since it is a randomised algorithm. Every call to $s(T_{n+1}, W_{n+1})$ performs the auxiliary function $\hat{s}_{\text{sum}}$ on average $1/p_{\text{swap}}$ times, each of which calls $s(M_n, W_n)$ precisely once and thus $s(T_n, W_n)$ exactly twice by eq. (5.29). Given access to a constant-time sampler from the uniform distribution on $[0,1]$, a sample from the base level $s(T_0, W_0)$ can be obtained in constant time, so a simple inductive argument shows that a drawing a single sample from (T_n, W_n) has average runtime $\mathcal{O}\left((2/p_{\text{swap}})^n\right)$, which equals

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

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

Following sec. 5.2.4, we also adjust the Monte Carlo algorithm to determine the waiting time and average Werner parameter in the d-NESTED-WITH-DISTILL repeater chain protocol. We add a recursive function sample_dist in algorithm 1 for sampling from the random variable $T_n^{d'}$ from eq. (5.24), which represents the waiting time at each level $d' \in \{0,1,\ldots,d\}$ of the nested distillation scheme. The relation between the random variable tuples $\left(T_n^{d'}, W_n^{d'}\right)$ and $\left(M_n^{d'}, V_n^{d'}\right)$ on the one hand and (T_n, W_n) and (M_n, V_n) on the other is mirrored in their implementations sample_dist and sample_swap, respectively: the function sample_swap calls sample_dist following eq. (5.26), which subsequently calls itself recursively for d nesting levels following eq. (5.23) and eq. (5.24) and calls sample_swap at the lowest level in line with eq. (5.25). See algorithm 1 for the full pseudocode of the d-dependent sampler sample_swap for the d-NESTED-WITH-DISTILL protocol.

The average runtime of the sampler for the d-NESTED-WITH-DISTILL protocol is upper bounded by $\mathcal{O}\left(4^d \cdot (2/p_{\text{swap}})^n\right)$. In order to derive this, note that the probability that a distillation attempt succeeds (see eq. (5.6)) is lower bounded by $1/2$ and hence a call to sample_dist(n,d) recursively performs at most $(2/(1/2))^d = 4^d$ calls to sample_swap$(n-1)$ on average. The average runtime of the full algorithm is the product of this number of calls and the average runtime of the NESTED-SWAP-ONLY algorithm $\mathcal{O}\left((2/p_{\text{swap}})^n\right)$ since the recurrence distillation scheme is performed at every swapping level.

Let us finish this section with an analysis of the algorithm's space complexity. For generating a single sample of (T_n, W_n) of the NESTED-SWAP-ONLY protocol, the number of variables that need to be stored grows linearly in the number of segments n. To see this, first note that at level ℓ the algorithm only needs to keep track of two samples of $(T_{\ell-1}, W_{\ell-1})$ at a time, since in the case of a failed swap it may discard the samples after updating the total time used and subsequently reuse the space for storing two fresh samples. In addition, for producing these two samples, only two samples need to be stored at *every* level $< \ell$. The insight here is that at each level the required two samples can be drawn *in sequence* rather than in parallel[1], so that the space needed to draw the first sample can be reused for the second. Therefore, the algorithm needs to keep track of at most two samples at every level, which implies that the total number of variables it stores is linear in the number of levels and thus in the number of segments n. For the d-NESTED-WITH-DISTILL protocol, the scaling is linear in $n \cdot d$ with d the number of distillation steps per nesting level, which can be shown by an analogous argument.

The number of samples that is required to generate a probability distribution histogram with pre-specified precision is independent of the number of segments (see explanation directly below eq. (5.28)). For constructing the histogram, we only need to store the waiting times for which at least a single sample was drawn and hence the number of such waiting times is also independent of the number of segments. We conclude that reproducing the probability distribution of (T_n, W_n) using the Monte Carlo algo-

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

rithm will not exceed polynomial space usage in the number of segments n.

Algorithm 1: Monte-Carlo algorithm sample_swap(n) for producing a single sample of the joint waiting time and Werner parameter (T_n, W_n) for a d-NESTED-WITH-DISTILL quantum repeater chain of 2^n segments as in sec. 5.2.2. Setting $d = 0$ corresponds to the NESTED-SWAP-ONLY repeater chain protocol.

Input : Success probabilities p_{gen} and p_{swap}, Werner parameter of single-hop links w_0, nesting level n, number of distillation rounds at each level d.
Output: Single sample from (T_n, W_n).

if $n = 0$ **then**
  $u \leftarrow$ uniform random sample from $[0,1]$
  return $(\lceil \log_{(1-p)}(1-u) \rceil, w_0)$ // (eq. (5.27))
else if $n \geq 1$ **then**
  $(t_A, w_A) \leftarrow$ sample_dist(n, d)
  $(t_B, w_B) \leftarrow$ sample_dist(n, d)
  $t, w \leftarrow g((t_A, w_A), (t_B, w_B))$ // (eq. (5.17))
  $u \leftarrow$ uniform random sample from $[0,1]$
  if $u \leq p_{\text{swap}}$ **then**
    return t, w
  else
    $t_{\text{retry}}, w_{\text{retry}} \leftarrow$ sample_swap(n)
    return $t + t_{\text{retry}}, w_{\text{retry}}$
  end
end

Auxiliary function sample_dist(n, d) :
if $d = 0$ **then**
  return sample_swap$(n-1)$
else
  $(t_A, w_A) \leftarrow$ sample_dist$(n, d-1)$
  $(t_B, w_B) \leftarrow$ sample_dist$(n, d-1)$
  $t, w \leftarrow g_{\text{D}}((t_A, w_A), (t_B, w_B))$ // eq. (5.24)
  $u \leftarrow$ uniform random sample from $[0,1]$
  // Success probability: eq. (5.6)
  if $u \leq p_{\text{dist}}(w_A, w_B)$ **then**
    return t, w
  else
    $t_{\text{retry}}, w_{\text{retry}} \leftarrow$ sample_dist(n, d)
    return $t + t_{\text{retry}}, w_{\text{retry}}$
  end
end

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

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

$C \leftarrow$ 3-dim. array of zeros,
    size $(n+1)\times(t_{\text{trunc}}+1)\times(t_{\text{trunc}}+1)$
$T \leftarrow$ 2-dim. array of zeros, size $(n+1)\times(t_{\text{trunc}}+1)$
$M \leftarrow$ 1-dim. array of zeros, of size $(t_{\text{trunc}}+1)$

// Base level probs (eq. (5.11))
for $t \in \{0,1,\ldots,t_{\text{trunc}}\}$ **do**
    $T[0,t] \leftarrow 1-(1-p_{\text{gen}})^t$
end

// Probabilities on higher levels
for $\ell \in \{0,1,\ldots,n-1\}$ **do**
    // Maximum of two copies (eq. (5.30))
    for $t \in \{1,2,\ldots,t_{\text{trunc}}\}$ **do**
        $M[t] \leftarrow T[\ell,t]^2 - T[\ell,t-1]^2$
    end
    // Conditional probs... (eq.(5.31))
    for $k \in \{1,2,\ldots,t_{\text{trunc}}\}$ **do**
        set column $C[\ell,k]$ to output of convolve(C[ℓ,k−1],k,M)
    end
    // ...and the marginals (eq. (5.32))
    for $t \in \{1,2,\ldots,t_{\text{trunc}}\}$ **do**
        for $k \in \{1,2,\ldots,t_{\text{trunc}}\}$ **do**
            term $\leftarrow p_{\text{swap}}(1-p_{\text{swap}})^{k-1}\cdot C[\ell,k,t]$
            add term to $T[\ell+1,t]$
        end
    end
    // Convert T to cumulative probs
    for $t \in \{1,2,\ldots,t_{\text{trunc}}\}$ **do**
        $T[\ell+1,t] \leftarrow T[\ell+1,t]+T[\ell+1,t-1]$
    end
end
return T

Auxiliary function convolve(S,k,M) :
if $k = 1$ **then**
    return M
else
    // Compute convolution of two arrays using Fast Fourier Transforms
    array_with_sum_distribution ← fast_convolution_algorithm(S,M)
    return array_with_sum_distribution
end

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

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

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

The following should be inserted directly after line 23 within the loop over ℓ in algorithm 2:

for $k \in \{1,2,\ldots,t_{\text{trunc}}\}$ **do**
  for $t_A \in \{1,2,\ldots,t_{\text{trunc}}\}$ **do**
    for $t_B \in \{1,2,\ldots,t_{\text{trunc}}\}$ **do**
      `// The average Werner parameter (eq. (5.35))`
      $w_A \leftarrow W[\ell, t_A]$
      $w_B \leftarrow W[\ell, t_B]$
      $t, w \leftarrow g((t_A, w_A), (t_B, w_B))$ `// (see eq. (5.17))`
      `// Add terms to the numerator of eq. (5.33)...`
      $p \leftarrow p_{\text{swap}} \left(1 - p_{\text{swap}}\right)^{k-1}$
      $p \leftarrow p \cdot T[\ell, t_A] \cdot T[\ell, t_B]$
      if $k = 1$ **then**
        `// ... in the case of a single swap (eq. (5.34))`
        $W[\ell+1, t] \leftarrow W[\ell+1, t] + w \cdot p$
      else
        `// ... in the case of multiple swaps (eq. (5.36))`
        for $t_{\text{fail}} \in \{0,1,2,\ldots,t_{\text{trunc}}-1\}$ **do**
          $t_{\text{deliver}} \leftarrow t + t_{\text{fail}}$
          add $w \cdot p \cdot C[\ell+1, k-1, t_{\text{fail}}]$ to $W[\ell+1, t_{\text{deliver}}]$
        end
      end
    end
  end
end
for $t \in \{1,2,\ldots,t_{\text{trunc}}\}$ **do**
  `// Normalise by dividing by the denominator of eq. (5.33)`
  $W[\ell+1, t] \leftarrow W[\ell+1, t] / T[\ell+1, t]$
end

5.3.2. SECOND ALGORITHM: DETERMINISTIC COMPUTATION

In this section, we present our full second algorithm, which computes the probability distribution of the waiting time and average Werner parameter up to some pre-specified truncation time t_{trunc}. The algorithm applies to the NESTED-SWAP-ONLY repeater protocol. In what follows, we first show how to compute the probability distribution of the waiting time T_n of the NESTED-SWAP-ONLY protocol by recursion (see sec. 5.2.1). After this, we outline how our algorithm performs a modified version of this computation on the finite domain $\{1,2,\ldots,t_{\text{trunc}}\}$. We finish the section by extending its computation to include the average Werner parameter (sec. 5.2.2).

Let us start by showing how to derive the probability distribution of the waiting time T_n in the NESTED-SWAP-ONLY protocol. For a single repeater segment ($n=0$), the waiting time follows the geometric distribution as given in eq. (5.11). For nesting levels $\ell \in \{0,1,2,\ldots,n\}$, the relation between the probability distributions of M_ℓ and T_ℓ follows straightforwardly from eq. (5.12):

$$\Pr(M_\ell = t) = \Pr(T_\ell \leq t)^2 - \Pr(T_\ell \leq t-1)^2. \tag{5.30}$$

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

$$\begin{aligned} \Pr\left(T_{\ell+1} = t | K_\ell = k\right) &= \Pr\left(\sum_{j=1}^{k} M_\ell^{(j)} = t\right) \\ &= \left[\circledast_{j=1}^{k} m_\ell\right](t) \end{aligned} \tag{5.31}$$

where we have denoted $m_\ell(t) := \Pr(M_\ell = t)$ and $*$ denotes convolution of functions (see sec. 5.1.3). The marginal probability distribution of $T_{\ell+1}$ is calculated from the distribution of the conditional random variable $T_{\ell+1}|K_\ell$ as

$$\Pr(T_{\ell+1} = t) = \sum_{k=1}^{\infty} p_{\text{swap}}(1-p_{\text{swap}})^{k-1} \Pr\left(T_{\ell+1} = t | K_\ell = k\right) \tag{5.32}$$

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

Our algorithm computes the probability distribution of T_n by iterating the procedure in the eqs. (5.30), (5.31) and (5.32) over ℓ from 0 to $n-1$ and is outlined in algorithm 2. Its implementation follows naturally from the equations above except for the following remarks. First, in the algorithm, the sum in eq. (5.32) is truncated at the pre-specified truncation time t_{trunc}. That this truncation yields correct probabilities $\Pr(T_{\ell+1} = t)$ for all $t \in \{0,1,\ldots,t_{\text{trunc}}\}$ follows from the fact that $\Pr(T_{\ell+1} = t | K_\ell > t) = 0$ since the generation of entanglement over any number of hops takes at least a single time step. Second, the convolutions in eq. (5.31) can be computed iteratively over k by noting that $\Pr(T_{\ell+1} = t | K_\ell = k+1)$ equals the convolution of $\Pr(T_{\ell+1} = t | K_\ell = k)$ and $m_\ell(t)$. Moreover, for a single convolution we use a well-known algorithm based on

Fast Fourier Transforms [43] which we denote by fast_convolution_algorithm in algorithm 2. This subroutine computes the convolution of two arrays of size t_{trunc} in time $\Theta(t_{\text{trunc}} \log t_{\text{trunc}})$.

The time complexity of the deterministic algorithm 2 equals $\Theta(n \cdot t_{\text{trunc}}^2 \log t_{\text{trunc}})$: the iteration over a single level is dominated by the $\Theta(t_{\text{trunc}}^2 \log t_{\text{trunc}})$ runtime of the convolutions in eq. (5.31) because eqs. (5.30) and (5.32) are performed in linear time in t_{trunc} by looping through an array of t_{trunc} elements. In sec. 5.4.2, we give an explicit expression for the truncation time t_{trunc} which ensures that $\Pr(T_n \leq t_{\text{trunc}}) \geq 0.99$. This expression is polynomial in the number of repeater segments, which implies that algorithm 2 runs in polynomial time in the number of segments also.

We extend our deterministic algorithm to also compute the average Werner parameter $W_n(t) := E[W_n | T_n = t]$ of the end-to-end link produced at time t by a 2^n-segment NESTED-SWAP-ONLY repeater chain (see sec. 5.2.2). The computation of the average Werner parameter at each level from 0 to n is performed after completion of the computation of the waiting time probabilities at the same level.

Let us explain the algorithm here (see algorithm (3) for pseudocode). At the base level the fidelity $W_0(t)$ equals the constant Werner parameter w_0 as in sec. 5.2.2 for all $t \in \{1, 2, \ldots, t_{\text{trunc}}\}$. At a higher level, the Werner parameter of a link which is delivered at time t is the output of g_W from eq. (5.18), averaged over all possible *realisations* of waiting times T_n which yield $T_n = t$. In order to precisely define what we mean by 'realisation', note that the waiting time T_n and average Werner parameter W_n as expressed recursively in sec. 5.2.2 are a function of K_n copies of (T_{n-1}, W_{n-1}), the waiting time and Werner parameter at one level lower. Regarding (T_n, W_n) as a function with K_n and all such copies of (T_{n-1}, W_{n-1}) as input, we define a 'realisation' of (T_n, W_n) as its evaluation on particular instances of these copies.

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

$$W_\ell(t) = \frac{\sum\limits_{r:\ r \text{ delivers link at } t} p_\ell(r) \cdot W_\ell^{\text{av}}(r)}{\sum\limits_{r:\ r \text{ delivers link at } t} p_\ell(r)} \tag{5.33}$$

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

In what follows, we will derive expressions for $p_\ell(r)$ and $W_\ell^{\text{av}}(r)$. This will give us an explicit expression for $W_\ell(t)$ and it is this expression that our algorithm evaluates. We distinguish between two cases of realisations for computing $p_\ell(r)$. In the first case, only a single swap (i.e. $K_\ell = 1$) is needed to produce the 2^ℓ-hop entanglement, i.e. the first swap from level $\ell - 1$ to ℓ is successful. The realisations r that belong to this case can be parameterised by the times t_A and t_B at which the two $2^{\ell-1}$-hop links are generated. The total probability of occurrence of these realisations, each of which delivers a 2^ℓ-hop link at time $g_{\text{T}}(t_A, t_B)$ (see eq. (5.10)), is given by

$$p_\ell(r) = \Pr(K_\ell = 1) \Pr(T_\ell = t_A) \Pr(T_\ell = t_B) \tag{5.34}$$

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

$$W_\ell^{\text{av}}(r) = g_{\text{W}}((t_A, W_{\ell-1}(t_A)), (t_B, W_{\ell-1}(t_B))) \tag{5.35}$$

where g_W is given in eq. 5.18.

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

$$\begin{aligned} p_\ell(r) \quad = \quad & \Pr(K_\ell = k) \cdot \Pr(T_\ell = t_{\text{fail}} | K_\ell = k-1) \\ & \cdot \Pr(T_\ell = t_A) \cdot \Pr(T_\ell = t_B) \end{aligned} \tag{5.36}$$

while the average Werner parameter $W_\ell^{\text{av}}(r)$ of the 2^ℓ-hop that is produced by each of these realisations is identical to the first case and is given in eq. (5.35). Each realisation in this group delivers a 2^ℓ-hop link at time $t_{\text{fail}} + g_{\text{T}}(t_A, t_B)$ (see eq. (5.10)).

Our algorithm loops over each group of realisations, evaluates their probabilities of success in eqs. (5.34) and (5.36) and their average Werner parameter in eq. (5.35) and subsequently computes $W_\ell(t)$ using eq. (5.33). The domain of the time parameters t_A, t_B and t_{fail} is bounded from above by t_{trunc} since no short-range link that is used to produce a long-range link at time $\leq t_{\text{trunc}}$ can take longer than t_{trunc}. Also, the total number of swaps K_n runs up to t_{trunc} since it cannot exceed the time at which the end-to-end link is delivered by the same reasoning as the truncation of the sum in eq. (5.32), i.e. $\Pr(T_{n+1} = t | K_n > t) = 0$. The pseudocode of the deterministic algorithm for computing the average Werner parameter can be found in algorithm 3.

The time complexity of the Werner-parameter algorithm can be inferred directly from algorithm 3 by the four loops with domain of size $\Theta(t_{\text{trunc}})$, which implies that the full time complexity is $\Theta(n \cdot t_{\text{trunc}}^4)$. This is polynomial in the number of repeater chain segments (see sec. 5.4.2).

5.3.3. POSSIBLE EXTENSIONS

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

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

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

where $|\phi_{++}\rangle := (|0\rangle\otimes|0\rangle \pm |1\rangle\otimes|1\rangle)/\sqrt{2}$ and $|\phi_{-\pm}\rangle := (|0\rangle\otimes|1\rangle \pm |1\rangle\otimes|0\rangle)/\sqrt{2}$ are the four Bell states and the Bell coefficients $p_{j,k}$ are probabilities which sum to 1. The implementation of the Monte Carlo method would have the Werner parameter W_n replaced by a joint random variable on the four[2] Bell coefficients $(p_{++}, p_{+-}, p_{-+}, p_{--})$, while the deterministic algorithm would compute the average over each of these four coefficients individually in a fashion similar to the average of the Werner parameter (eq. (5.33)). Successful entanglement swap and distillation operations both map Bell-diagonal states to Bell-diagonal states [24] and could thus each be formulated as an operation on the four Bell coefficients.
An example of a different model of memory decoherence noise (currently eq. (5.3)) is the application of the Pauli operator $Z := |0\rangle\langle 0| - |1\rangle\langle 1|$ on one of the two qubits with probability

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

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

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

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

In addition to more general state and noise models, both algorithms could also be applied to more general network topologies than a chain. An example is the generation of a Greenberger-Horne-Zeilinger (GHZ) state [44] in a star network, where there is a single central node and each of the other nodes (the leaves) is connected to this single central node only. All leave nodes start by generating an elementary link with the central node, after which the central node performs a local operation to convert these links into a single GHZ state on all the leave nodes, e.g. by a combination of two-qubit controlled-rotation gates and single-qubit measurements [39]. Similar to our model of the swap operation, we could model the local operation that produces the GHZ state as probabilistic, motivated by probabilistic two-qubit operations in linear photonics [45]. In the same spirit as the NESTED-SWAP-ONLY protocol and our analysis of it in sec. 5.2, the central node waits for all links to have been generated (which corresponds to the maximum of their individual waiting times) while failure of the local operation requires regeneration of the elementary links (which corresponds to the geometric compound sum). Since both maximums and geometric sums of random variables can be treated by the two algorithms, both could be used to sample the produced state and waiting time in the star network.

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

5.4. BOUNDS ON THE MEAN WAITING TIME

In this section, we first show how to use the deterministic algorithm from section 5.3.2 to obtain bounds on the mean of the waiting time T_n, which improve upon a common analytical approximation. Then we give an explicit expression for the choice of truncation time in the algorithm for which 99% of probability mass of T_n is captured.

5.4.1. NUMERICAL MEAN USING THE DETERMINISTIC ALGORITHM

Here, we show how to obtain bounds on the mean of T_n using the deterministic algorithm from section 5.3.2. Such bounds are interesting since a common approximation to the mean in the regime of small success probabilities p_{gen} and p_{swap}, the *3-over-2-formula* [6, 13, 21, 34]

$$E[T_n] \approx \left(\frac{3}{2p_{\text{swap}}}\right)^n \cdot \frac{1}{p_{\text{gen}}}, \tag{5.37}$$

overestimates the waiting time for large success probabilities. For example, it can be seen in [35, fig. 7(a)] (reproduced in this chapter as fig. 5.7, top plot) that for $p_{\text{gen}} = p_{\text{swap}} = 1$, the ratio [true mean]/[approximation] of the true mean $E[T_n]$ and the approximation in eq. (5.37) decreases as a function of the number of segments and equals 0.2 for a chain of 16 segments, i.e. an overestimation by a factor $\frac{1}{0.2} = 5$. In fig. 5.7 (bottom plot), it can be seen that this overestimation grows to more than a factor $\frac{1}{0.05} = 20$ for a chain of 2048 segments.

The bounds are obtained in two steps. First, we perform the deterministic algorithm to compute the probability distribution of T_n, as described in section 5.3.2. Since this probability distribution is only computed by the algorithm on the truncated domain $\{0, 1, \ldots, t_{\text{trunc}}\}$, we cannot calculate the mean of T_n. Instead, we compute its *empirical mean*, which we define for random variable X with the nonnegative integers as domain as

$$E[X, t_{\text{trunc}}] := \sum_{t=1}^{t_{\text{trunc}}} \Pr(X \geq t). \tag{5.38}$$

Note that the empirical mean reduces to the real mean for $t_{\text{trunc}} \to \infty$ (see section 5.1.3).

In the second step, we quantify how well the empirical mean of T_n approximates its real mean. We need two tools for doing so. As first tool, we introduce the random variable T_n^{upper}, which is identical to T_n except for the fact that the two links required for the entanglement swap are produced sequentially at every level rather than in parallel. We proceed analogously to the first step: we perform a modified version of the deterministic algorithm to compute the probability distribution of T_n^{upper} and we compute its empirical mean (details and formal definition of T_n^{upper} can be found in appendix 5.7.2). In contrast to T_n, we are able to compute the real mean of T_n^{upper}, which equals $E[T_n^{\text{upper}}] = (2/p_{\text{swap}})^n \cdot 1/p_{\text{gen}}$ (proof in appendix 5.7.2). The second tool is the following proposition, which states that for T_n the empirical mean converges at least as fast to the real mean with increasing truncation time as for T_n^{upper}.

Proposition 1. *The difference between the real mean and the empirical mean (eq. (5.38)) of the waiting time is bounded as*

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

and the two bounds coincide for $t_{\text{trunc}} \to \infty$. *The random variable* T_n^{upper} *is formally defined in appendix 5.7.2.*

The main tool for proving proposition 1 is the fact that T_n^{upper} stochastically dominates T_n for every nesting level n, which means that $\Pr(T_n \geq t) \leq \Pr\left(T_n^{\text{upper}} \geq t\right)$ for all $t \in \{0, 1, 2, \ldots\}$. We formally prove the proposition and give a more detailed version of the computation of the probability distribution of T_n^{upper} in appendix 5.7.2.

5.4.2. CHOOSING A TRUNCATION TIME FOR THE DETERMINISTIC ALGORITHM

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

$$\Pr(T_n \geq t_{\text{trunc}}) \leq E[T_n] / t_{\text{trunc}}. \tag{5.39}$$

We upper bound the mean of T_n in eq. (5.39) by invoking proposition 1 with $t_{\text{trunc}} = 0$. The latter reduces to $E[T_n] \leq (2/p_{\text{swap}})^n \cdot 1/p_{\text{gen}}$ and thus implies

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

Consequently, setting

$$t_{\text{trunc}} = \left(\frac{2}{p_{\text{swap}}}\right)^n \cdot \frac{1}{p_{\text{gen}}} \cdot \frac{1}{1 - 0.99} \tag{5.40}$$

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

5.5. NUMERICAL RESULTS

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

Our proof-of-principle implementation can be found in [46]. The reported computation times have been obtained from single-threaded computations on commodity hardware (specifically: a single logical processor of an Intel i7-4770K CPU @ 3.85 GHz). In the plot captions in this section, we state the computation time for the largest number of repeater segments because computing the distribution of (T_n, W_n) requires finding the distribution of (T_{n-1}, W_{n-1}) first (see sec. 5.2).

First we consider the waiting time in the NESTED-SWAP-ONLY protocol. Our algorithms are able to recover the results from Shchukin et al. [35], both the full distribution of waiting time exactly (fig. 5.2, top plot) as well as its mean up to arbitrary precision (fig. 5.7, top plot), and extend these results from 16 to 8192 and to 2048 repeater segments, respectively (figs. 5.2, 5.3 and 5.7). In fig. 5.3 we compare results from both our

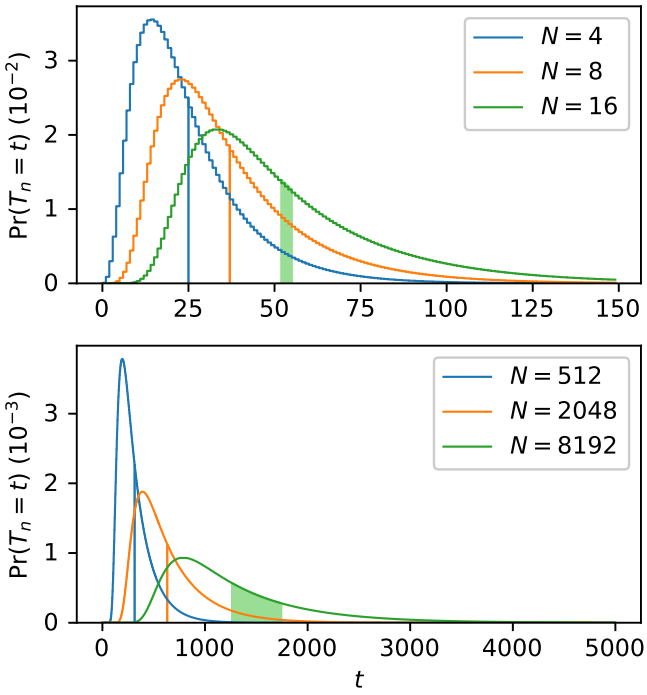


Figure 5.2: Probability distributions of the waiting time and bounds on the mean (vertical shaded areas, see 5.4.1) calculated by the deterministic algorithm for the NESTED-SWAP-ONLY protocol. The repeater chain parameters are $p_{\text{gen}} = 0.1$, $p_{\text{swap}} = 0.9$, and the number of repeater segments is given by N. The top plot recovers the results from Shchukin et al. [35, Fig.10(a)]. Computation time ≈ 5 seconds for $N = 8192$.

Monte Carlo and deterministic algorithm and find that there is good agreement between the two. For high swapping success probability p_{swap} the deterministic algorithm can compute probability distributions up to thousands of nodes, as illustrated in fig. 5.2. For small p_{swap}, we have found that the number of repeater segments $N = 2^n$ that we can simulate is limited in practice. This is a consequence of the fact that t_{trunc} grows fast in N for small p_{swap} if we want the guarantee that 99% of the probability distribution is captured (see eq. (5.40)), and the polynomial scaling in t_{trunc} of the algorithm's runtime.

Secondly, we consider the average fidelity of the NESTED-SWAP-ONLY and d-NESTED-WITH-DISTILL protocols. We investigate the NESTED-SWAP-ONLY protocol with a small number of segments ($N = 1, 2, 4$), see fig. 5.4. We observe that fidelity stabilises as the waiting time increases, and it stabilises at values for which the state remains entangled in spite of the absence of distillation. Again, the deterministic and Monte Carlo algorithms show good agreement. Adding the calculation of fidelity increases the time complexity of the deterministic algorithm, which reduced the maximum number of segments that we could simulate. We found that the Monte Carlo algorithm is able to simulate a larger number of segments, as its computational complexity is unchanged when also tracking the fidelity.

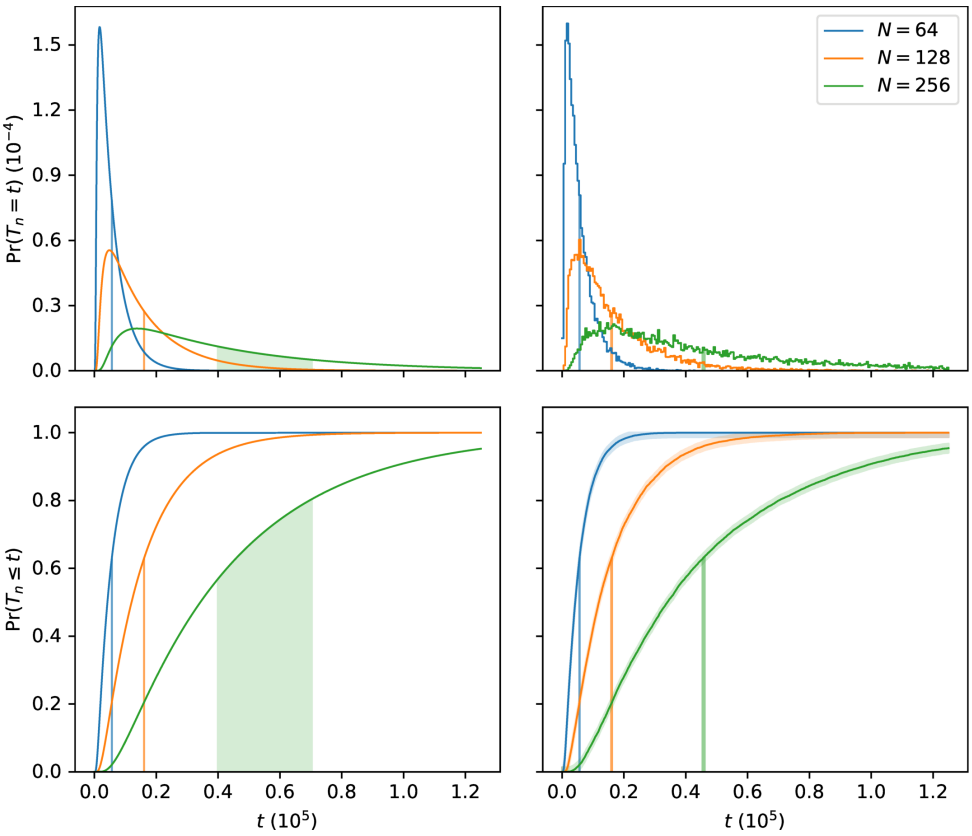


Figure 5.3: The probability distributions of waiting time for repeater chains with varying number of segments N for the NESTED-SWAP-ONLY protocol, computed using the deterministic algorithm (left plots) and generated from 15,000 samples from the Monte Carlo simulation (right plots). Vertical shaded areas indicate bounds on the mean in the left plots (see sec. 5.4.1) and the sample mean ± one standard error in the right plot. In the bottom right plot the shaded bands are confidence bands for a probability $z = 0.01$ that the actual distribution lies outside of these bands, obtained from the Dvoretzky-Kiefer-Wolfowitz inequality [42] discussed in sec. 5.3.1. The repeater chain parameters are $p_{\text{gen}} = 0.1$, $p_{\text{swap}} = 0.5$. The computation time for the deterministic algorithm ≈ 9 minutes, while for the Monte Carlo algorithm ≈ 30 minutes.

Third, we consider the d-NESTED-WITH-DISTILL protocol. In fig. 5.5, we study the effects of distillation in a repeater chain of 4 segments comparing one distillation round ($d = 1$) against no distillation rounds ($d = 0$) for two different memory coherence times. We first observe the increase in the waiting times caused by the generation of the additional links necessary for distillation. An increase in waiting time is accompanied by an increase in memory decoherence, which implies that the degree to which distillation is beneficial depends on the memory coherence time. The values for the coherence time we chose allow to show both types of behaviour.

Finally, we incorporate the communication time for the entanglement swap into our model following sec. 5.3.1. Fig. 5.6 shows how the output probability distributions change when we include this communication time. We confirm that, as stated by Brask

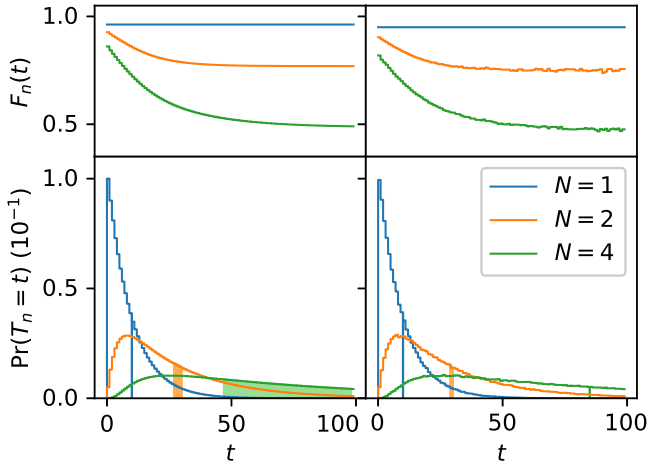


Figure 5.4: The average fidelity of links delivered at time t by an N-segment NESTED-SWAP-ONLY repeater chain (top row), and the corresponding probability distributions (bottom row), from both the deterministic algorithm (left column) and the Monte Carlo algorithm (right column) using 250,000 samples. For the deterministic figures the vertical shaded areas indicate numerical bounds on the mean (see sec. 5.4.1), and for the Monte Carlo figures these indicate the sample mean ± one standard error. The repeater chain parameters for NESTED-SWAP-ONLY protocol are $p_{\text{gen}} = 0.1$, $p_{\text{swap}} = 0.5$, $T_{\text{coh}} = 50$ time steps and the fidelity of the elementary links equals $F_0 = 0.95$, which corresponds to Werner parameter $w_0 = (4 \cdot 0.95 - 1)/3 \approx 0.93$ following eq. (5.2). The computation time for the deterministic algorithm ≈ 15 minutes, while for the Monte Carlo algorithm ≈ 20 seconds.

and Sørensen [6], omitting this communication time gives a good approximation for small p_{gen}, but not for larger p_{gen}.

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

$$\Pr(X \leq t) = \exp\left(-(1+\xi s)^{-1/\xi}\right) \tag{5.41}$$

where X is a random variable following the GEV distribution, $s = (t-\mu)/\sigma$, and $\xi > 0$, $\sigma > 0$ and $\mu \in \mathbb{R}$ are the free parameters [47]. Fig. 5.8 shows a typical result of such a fit. We find that the fit seems rather close to the computed distribution, although the difference in the means indicates that the fit should only be used to make approximate statements.

A good fit could provide a speedup for the deterministic algorithm, since the algorithm computes the distribution at each level from the distribution at the previous level. To be precise, the algorithm's runtime can be reduced by starting the computation at the fitted distribution instead of computing the distribution at level n, and subsequently using this distribution to have the algorithm compute the distribution at level $n+1$. In order to ensure that the distribution at the final level $> n$ still approximates the real distribution, careful analysis of the acquired error of the distribution at higher levels is required. We leave such error accumulation analysis, based on a distribution that forms a

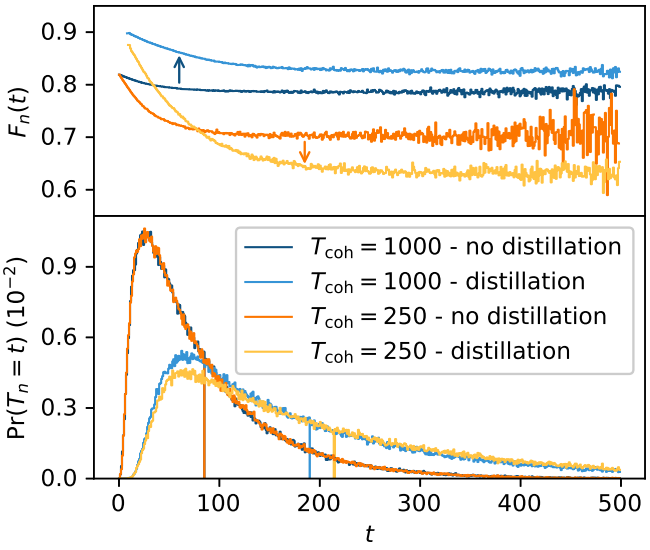


Figure 5.5: Comparison between the d-NESTED-WITH-DISTILL protocol with a single distillation round on every level ($d = 1$) and the NESTED-SWAP-ONLY protocol (no distillation) for a repeater chain with $N = 4$ segments, for longer and shorter memory coherence times T_{coh}. The additional repeater chain parameters are $p_{\text{gen}} = 0.1$, $p_{\text{swap}} = 0.5$, and $F_0 = 0.95$, which corresponds to $w_0 \approx 0.93$. While the goal of distillation is to improve the fidelity of delivered links, when the coherence time is too short compared to the time needed to deliver a link, adding distillation actually decreases the fidelity (orange arrow). For longer coherence times adding distillation does improve the fidelity (blue arrow). In both cases the waiting time increases because entanglement distillation requires more links to be generated. For the NESTED-SWAP-ONLY protocol, the waiting time is independent of the memory coherence time (in contrast to fidelity), which can be observed from the identical waiting times in the bottom plot. Each curve has been generated from 250,000 Monte Carlo algorithm samples. Computation time ≈ 15 seconds without distillation, ≈ 100 seconds with distillation.

phenomenological fit, for future work.

5.6. DISCUSSION

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

Both algorithms run in time polynomial in the number of nodes, which is faster than the exponential runtime of previous algorithms. The workhorse behind the improved complexity is a formal recursive definition of the waiting time and the state produced by the chain. We developed an open source proof-of-principle implementation in [46],

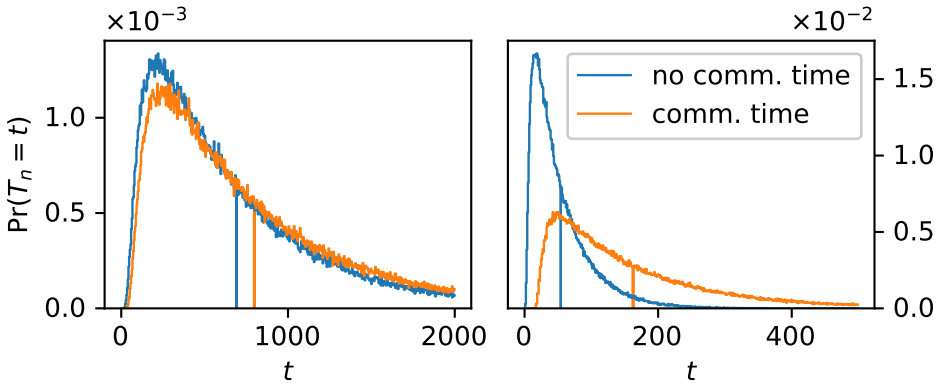


Figure 5.6: Waiting time distributions with and without communication time for entanglement swapping for the NESTED-SWAP-ONLY protocol, with entanglement generation success probabilities $p_{\text{gen}} = 0.1$ (left) and $p_{\text{gen}} = 0.9$ (right), generated from 250,000 samples from the Monte Carlo algorithm. The vertical bars indicate the mean ± one standard error. As stated by Brask and Sørensen [6] omitting this communication time gives a good approximation when p_{gen} is small. The repeater chain has $N = 16$ segments and $p_{\text{swap}} = 0.5$. Computation time ≈ 5 minutes per curve.

5

which allows to analyse repeater chains with several thousands of segments for some parameter regimes.

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

We have been able to adapt our algorithms to several protocols for repeater chains. More concretely, we have studied the NESTED-SWAP-ONLY protocol and two generalisations. The first one includes distillation d-NESTED-WITH-DISTILL, the second one takes into account the communication time in the swap operations. We believe that the tools we have developed here can be extended to several other protocols without losing the polynomial runtime. Some examples which we leave for further work include tracking the full density matrix, variations of d-NESTED-WITH-DISTILL with unequal spacing of the nodes or with a number of segments different than a power of two, and the investigation of more general network topologies. Inspired by hardware, it would also be interesting to model decaying memory efficiency and nodes that can not generate entanglement concurrently with both adjacent neighbours.

In summary, we have proposed two efficient algorithms to characterise the behaviour of repeater chain protocols. We expect our algorithms to find use in the study and analysis of future quantum networks. Moreover, the existence of protocols capable of efficiently characterising the state produced opens the door to real-time decision

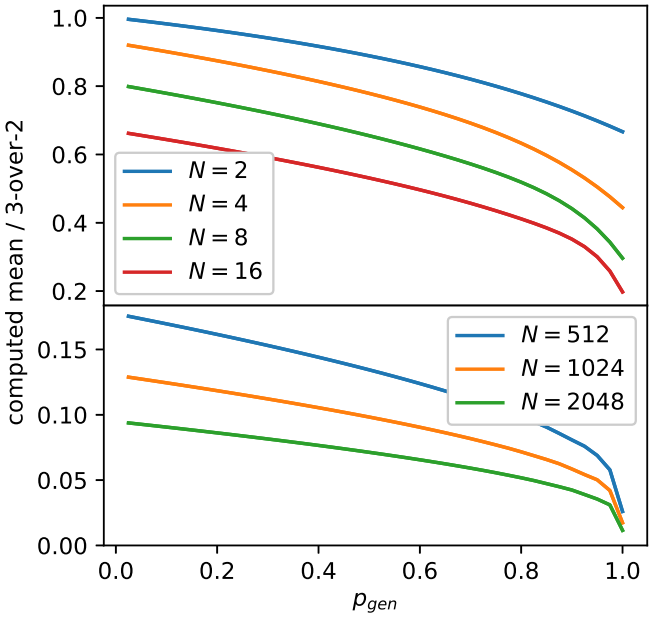


Figure 5.7: Ratio between the bound on the mean computed by the deterministic algorithm and the 3-over-2 approximation eq. (5.37) as a function of entanglement generation success probability p_{gen} with deterministic swapping ($p_{\text{swap}} = 1$). For each number of segments N, the figures show two lines: one for the upper and lower bound on the mean (see sec. 5.4.1). The fact that for each N only a single thick line rather than two lines can be seen indicates that the bounds on the mean almost coincide. The top figure recovers work by Shchukin et al. [35, fig. 7(a)], whose exponential-time algorithm based on Markov chains is able to compute the mean exactly while our algorithms can get arbitrarily tight bounds on the mean (deterministic algorithm) or approximate the mean with arbitrary precision (Monte Carlo algorithm) at the benefit of polynomial runtime. The runtime improvement over the Markov-chain approach allows us to extend the results of Shchukin et al. to more than 2000 segments (bottom figure). Each curve was generated by running the deterministic algorithm for 40 different values of p_{gen} $(0.025, 0.05, \ldots, 1)$ and the truncation time was set to $t_{\text{trunc}} = 1000$. Computation time for each curve $\lesssim 2$ seconds.

taking at the nodes based on this knowledge.

5.7. APPENDIX

5.7.1. DISTILLATION OF WERNER STATES

In this appendix, we find the state after successful entanglement distillation on two Werner states. Performing entanglement distillation on two Werner states with Bell-state fidelities F_A and F_B yields a state with Bell-state fidelity [24]

$$\frac{\left(F_A F_B + \frac{1}{9}\bar{F}_A\bar{F}_B\right)}{p_{\text{dist}}} \tag{5.42}$$

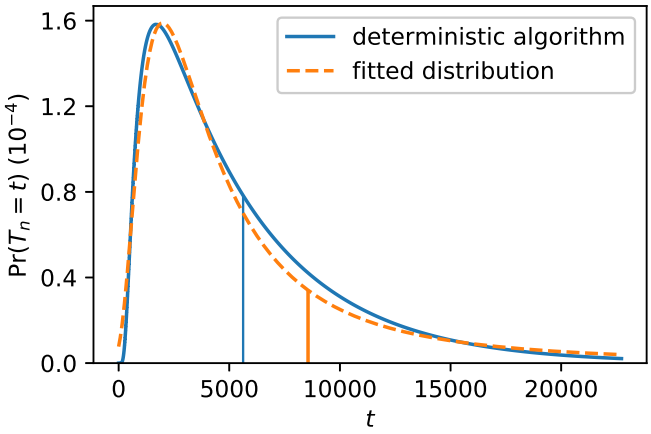


Figure 5.8: Waiting time distribution (64 segments, $p_{\text{gen}} = 0.1$, $p_{\text{swap}} = 0.5$), computed with the deterministic algorithm (sec. 5.3.2), and a fit to the same distribution using the generalised extreme value (GEV) distribution (see eq. (5.41); fitting parameters: $\xi \approx -0.5997, \mu \approx 3092.7, \sigma \approx 2694.9$). Although the two distributions seem to coincide fairly well by eye, the difference between the means (vertical lines) is relatively large (a factor ≈ 1.5). For different number of segments and success probabilities p_{gen} and p_{swap}, the difference between fitted and computed distribution is similar.

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

$$F_A F_B + \frac{1}{3} F_A \bar{F}_B + \frac{1}{3} \bar{F}_A F_B + \frac{5}{9} \bar{F}_A \bar{F}_B \tag{5.43}$$

where we have denoted $\bar{F} = 1 - F$. Although the output state is not a Werner state, it is always possible to transform it into a Werner state with the same Bell-state fidelity by local operations. We rewrite eqs. (5.42) and (5.43) as function of the Werner parameters w_A and w_B of the input states rather than their fidelities F_A and F_B using eq. (5.2), which yields eqs. (5.5) and (5.6).

5.7.2. COMPUTATION OF T_n^{upper}

In this appendix, we first prove proposition 1 and subsequently show how a modified version of the deterministic algorithm from section 5.3.2 computes the empirical mean $E\left[T_n^{\text{upper}}, t_{\text{trunc}}\right]$ from eq. (5.38).

PROOF OF PROPOSITION 1

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

$$T_0^{\text{upper}} = T_0 \tag{5.44}$$

$$M_{n+1}^{\text{upper}} = \left(T_n^{\text{upper}}\right)^{(A)} + \left(T_n^{\text{upper}}\right)^{(B)} \tag{5.45}$$

$$T_{n+1}^{\text{upper}} = \sum_{k=1}^{K_n} \left(M_n^{\text{upper}}\right)^{(k)} \tag{5.46}$$

where T_0 is defined in section 5.2.1 and K_n is geometrically distributed with parameter p_{gen} for all n.

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

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

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

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

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

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

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

where we use the notation $X^{(.)}$ to denote an i.i.d. copy of X, following sec. 5.1.3.

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

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

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

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

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

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

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

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

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

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

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

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

Proposition 2. *It holds that* $T_n \leq_{\text{st}} T_n^{\text{upper}}$ *for all* $n \geq 0$.

Proof. We use induction on n. The base case $n = 0$ is immediate since $T_0^{\text{upper}} = T_0$ (eq. (5.45)). Now suppose that $T_n \leq_{\text{st}} T_n^{\text{upper}}$ for some $n \geq 0$. It follows directly from lemma 1(a) that $M_n \leq_{\text{st}} M_n^{\text{upper}}$, where M_n is given in eq. (5.9) and M_n^{upper} in eq. (5.45). Using lemma 1(e), we find that the dominance $M_n \leq_{\text{st}} M_n^{\text{upper}}$ implies $T_{n+1} \leq_{\text{st}} T_{n+1}^{\text{upper}}$ where T_{n+1} and T_{n+1}^{upper} are defined in eqs. (5.8) and (5.46), respectively. This concludes our proof. □

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

Lemma 2.

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

Proof. We use induction on n. Since T_0^{upper} equals T_0, which is geometrically distributed with parameter p_{gen}, we have $E\left[T_0^{\text{upper}}\right] = 1/p_{\text{gen}}$. For the induction step, first note that by linearity of the mean, we have

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

The last step is given by Wald's identity [48], which states that the mean of a compound sum equals the product of the mean of the summand and the random variable that is the summation upper bound:

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

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

The following lemma, which states that stochastic domination implies domination of the empirical mean from eq. (5.38) provides the last step in proving proposition 1.

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

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

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

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

$$E[X] - E[X, t_{\text{trunc}}] = \sum_{t=t_{\text{trunc}}+1}^{\infty} \Pr(X \geq t). \tag{5.47}$$

The upper bound follows from eq. (5.47) and the definition of stochastic dominance: $\Pr(X \geq t) \leq \Pr(Y \geq t)$ for all $t \in \{0, 1, \ldots\}$. □

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

COMPUTING THE EMPIRICAL MEAN OF T_n^{upper}

Here, we outline how the deterministic algorithm computes $E\left[T_n^{\text{upper}}, t_{\text{trunc}}\right]$, which is needed for determining a bound on the mean of T_n using proposition 1. First note that T_n^{upper} and T_n are identical except for the difference between M_n in eq. (5.9), which equals the maximum of two copies of the waiting time, and M_n^{upper} in eq. (5.45), which equals their sum. We modify the algorithm to account for this difference by replacing the computation of the probability distribution of M_n in eq. (5.30) by the convolution

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

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

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

REFERENCES

[1] K. Azuma, K. Tamaki, and H.-K. Lo, *All-photonic quantum repeaters,* Nature Communications **6**, 6787 (2015).

[2] N. K. Bernardes, L. Praxmeyer, and P. van Loock, *Rate analysis for a hybrid quantum repeater,* Phys. Rev. A **83**, 012323 (2011).

[3] J. Borregaard, P. Kómár, E. M. Kessler, M. D. Lukin, and A. S. Sørensen, *Long-distance entanglement distribution using individual atoms in optical cavities,* Physical Review A **92**, 012307 (2015).

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

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

[6] J. B. Brask and A. S. Sørensen, *Memory imperfections in atomic-ensemble-based quantum repeaters,* Phys. Rev. A **78**, 012350 (2008).

[7] H.-J. Briegel, W. Dür, J. I. Cirac, and P. Zoller, *Quantum repeaters: The role of imperfect local operations in quantum communication,* Phys. Rev. Lett. **81**, 5932 (1998).

[8] D. Buterakos, E. Barnes, and S. E. Economou, *Deterministic generation of all-photonic quantum repeaters from solid-state emitters,* Physical Review X **7**, 041023 (2017).

[9] O. A. Collins, S. D. Jenkins, A. Kuzmich, and T. A. B. Kennedy, *Multiplexed memory-insensitive quantum repeaters,* Phys. Rev. Lett. **98**, 060502 (2007).

[10] L.-M. Duan, M. D. Lukin, J. I. Cirac, and P. Zoller, *Long-distance quantum communication with atomic ensembles and linear optics,* Nature **414**, 413 EP (2001), article.

[11] A. G. Fowler, D. S. Wang, C. D. Hill, T. D. Ladd, R. Van Meter, and L. C. Hollenberg, *Surface code quantum communication,* Physical Review Letters **104**, 180503 (2010).

[12] S. Guha, H. Krovi, C. A. Fuchs, Z. Dutton, J. A. Slater, C. Simon, and W. Tittel, *Rate-loss analysis of an efficient quantum repeater architecture,* Phys. Rev. A **92**, 022357 (2015).

[13] L. Jiang, J. M. Taylor, and M. D. Lukin, *Fast and robust approach to long-distance quantum communication with atomic ensembles,* Phys. Rev. A **76**, 012301 (2007).

[14] L. Jiang, J. M. Taylor, K. Nemoto, W. J. Munro, R. Van Meter, and M. D. Lukin, *Quantum repeater with encoding,* Physical Review A **79**, 032325 (2009).

[15] W. Munro, K. Harrison, A. Stephens, S. Devitt, and K. Nemoto, *From quantum multiplexing to high-performance quantum networking,* Nature Photonics **4**, 792 (2010).

[16] S. Muralidharan, J. Kim, N. Lütkenhaus, M. D. Lukin, and L. Jiang, *Ultrafast and fault-tolerant quantum communication across long distances,* Phys. Rev. Lett. **112**, 250501 (2014).

[17] S. Muralidharan, L. Li, J. Kim, N. Lütkenhaus, M. D. Lukin, and L. Jiang, *Optimal architectures for long distance quantum communication,* Scientific Reports **6**, 20463 EP (2016), article.

[18] S. Muralidharan, C.-L. Zou, L. Li, J. Wen, and L. Jiang, *Overcoming erasure errors with multilevel systems,* New Journal of Physics **19**, 013026 (2017).

[19] M. Pant, H. Krovi, D. Englund, and S. Guha, *Rate-distance tradeoff and resource costs for all-optical quantum repeaters,* Physical Review A **95**, 012304 (2017).

[20] S. Santra, L. Jiang, and V. S. Malinovsky, *Quantum repeater architecture with hierarchically optimized memory buffer times,* Quantum Science and Technology **4**, 025010 (2019).

[21] C. Simon, H. de Riedmatten, M. Afzelius, N. Sangouard, H. Zbinden, and N. Gisin, *Quantum repeaters with photon pair sources and multimode memories,* Phys. Rev. Lett. **98**, 190503 (2007).

[22] R. Van Meter, T. D. Ladd, W. Munro, and K. Nemoto, *System design for a long-line quantum repeater,* IEEE/ACM Transactions on Networking (TON) **17**, 1002 (2009).

[23] W. J. Munro, K. Azuma, K. Tamaki, and K. Nemoto, *Inside quantum repeaters,* IEEE Journal of Selected Topics in Quantum Electronics **21**, 78 (2015).

[24] W. Dür and H. Briegel, *Entanglement purification and quantum error correction,* Reports on Progress in Physics **70**, 1381 (2007).

[25] N. Gisin, *Hidden quantum nonlocality revealed by local filters,* Physics Letters A **210**, 151 (1996).

[26] C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, and W. K. Wootters, *Purification of noisy entanglement and faithful teleportation via noisy channels,* Phys. Rev. Lett. **76**, 722 (1996).

[27] D. Deutsch, A. Ekert, R. Jozsa, C. Macchiavello, S. Popescu, and A. Sanpera, *Quantum privacy amplification and the security of quantum cryptography over noisy channels,* Phys. Rev. Lett. **77**, 2818 (1996).

[28] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, *Mixed-state entanglement and quantum error correction,* Physical Review A **54**, 3824 (1996).

[29] E. T. Campbell and S. C. Benjamin, *Measurement-based entanglement under conditions of extreme photon loss,* Physical Review Letters **101**, 130502 (2008).

[30] F. Rozpędek, T. Schiet, D. Elkouss, A. C. Doherty, S. Wehner, *et al.*, *Optimizing practical entanglement distillation,* Physical Review A **97**, 062333 (2018).

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

[32] S. Abruzzo, S. Bratzik, N. K. Bernardes, H. Kampermann, P. van Loock, and D. Bruß, *Quantum repeaters and quantum key distribution: Analysis of secret-key rates,* Phys. Rev. A **87**, 052315 (2013).

[33] S. Khatri, C. T. Matyas, A. U. Siddiqui, and J. P. Dowling, *Practical figures of merit and thresholds for entanglement distribution in quantum networks,* Phys. Rev. Research **1**, 023032 (2019).

[34] N. Sangouard, C. Simon, H. de Riedmatten, and N. Gisin, *Quantum repeaters based on atomic ensembles and linear optics,* Rev. Mod. Phys. **83**, 33 (2011).

[35] P. v. L. E. Shchukin, F. Schmidt, *On the waiting time in quantum repeaters with probabilistic entanglement swapping,* arXiv:1710.06214 (2017).

[36] S. E. Vinay and P. Kok, *Statistical analysis of quantum-entangled-network generation,* Phys. Rev. A **99**, 042313 (2019).

[37] F. Rozpędek, K. Goodenough, J. Ribeiro, N. Kalb, V. C. Vivoli, A. Reiserer, R. Hanson, S. Wehner, and D. Elkouss, *Parameter regimes for a single sequential quantum repeater,* Quantum Science and Technology **3**, 034002 (2018).

[38] F. Rozpędek, R. Yehia, K. Goodenough, M. Ruf, P. C. Humphreys, R. Hanson, S. Wehner, and D. Elkouss, *Near-term quantum-repeater experiments with nitrogen-vacancy centers: Overcoming the limitations of direct transmission,* Phys. Rev. A **99**, 052330 (2019).

[39] J. Cirac, A. Ekert, S. Huelga, and C. Macchiavello, *Distributed quantum computation over noisy channels,* Physical Review A **59**, 4249 (1999).

[40] R. F. Werner, *Quantum states with Einstein-Podolsky-Rosen correlations admitting a hidden-variable model,* Phys. Rev. A **40**, 4277 (1989).

[41] W. Feller, *An introduction to probability theory and its applications*, Vol. I (John Wiley and Sons, New York, 1957) p. 461.

[42] A. Dvoretzky, J. Kiefer, and J. Wolfowitz, *Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator,* Ann. Math. Statist. **27**, 642 (1956).

[43] J. W. Cooley and J. W. Tukey, *An algorithm for the machine calculation of complex fourier series,* Mathematics of Computation **19**, 297 (1965).

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

[45] M. A. Nielsen, *Optical quantum computation using cluster states,* Phys. Rev. Lett. **93**, 040503 (2004).

[46] git, *Waiting time and fidelity in quantum repeater chains,* `https://github.com/sebastiaanbrand/waiting-time-quantum-repeater-chains` (2019).

[47] M. Charras-Garrido and P. Lezaud, *Extreme value analysis: an introduction,* Journal de la Société Française de Statistique **154**, pp (2013).

[48] A. Wald, *Sequential Analysis* (Courier Corporation, Dover, New York, 1947).

6

Efficient Optimisation of Cut-offs in Quantum Repeater Chains

In this chapter, we investigate how repeater protocols can be improved by adding a cut-off, for instance, a maximum storage time for entanglement after which it is discarded. We will first develop an improved version of the algorithm of Chapter 5 for computing the probability distribution of the waiting time and fidelity of entanglement produced by repeater chain protocols. The algorithm from this chapter is faster and moreover can handle protocols which include cut-offs. Next, we use the algorithm to optimise cut-offs in order to maximise secret-key rate between the end nodes of the repeater chain. We find that the use of the optimal cut-off extends the parameter regime for which secret key can be generated and moreover significantly increases the secret-key rate for a large range of parameters.

This chapter is a modified version of the publication: B. Li, T. Coopmans and D. Elkouss, *Efficient optimization of cut-offs in quantum repeater chains*, IEEE Transactions on Quantum Engineering (2021) .

Most quantum repeater schemes require quantum memories [1, 2], which makes them suffer from memory noise if the memories are imperfect. To see how, we note that in many protocols an entangled pair is generated that needs to wait in a quantum memory until the generation of an additional pair. During this waiting time the first pair decoheres, reducing the quality of the final entanglement produced. At the cost of a lower rate, this effect can be mitigated by imposing a cut-off condition. For instance, a maximum storage time for entanglement after which it is discarded [3].

Cut-offs have been considered for entanglement generation in different contexts [3–14]. Notably, they play a key role for generating entanglement already in multi-pair experiments between adjacent nodes [5]. They also promise to be helpful in near-term quantum repeater experiments [6, 7, 11]. In the multi-repeater case, it is possible to obtain analytical expressions for the waiting time for general families of protocols [12, 13], though in general it appears challenging to extend those methods to characterise the quality of the states produced. Santra et al. [8] analytically optimised the distillable entanglement for a restricted class of quantum repeater schemes.

In this chapter, we first characterise the performance of a very general class of repeater schemes including cut-offs, probabilistic swapping, distillation and memory decoherence. We sidestep the challenge of analytical characterisation by computing the probability distribution of the waiting time and fidelity of the first generated entangled pair between the repeater's end nodes. For this, we improve the closed-form expressions from Chapter 5 to get faster algorithm runtimes and extend the expressions to repeater schemes which involve distillation and cut-offs. The runtime of the algorithm which evaluates these expressions is polynomial in the pre-specified size of the computed probability distribution's support.

In the second part of the chapter, we optimise the choices of the cut-off to maximise the secret-key rate. We study different cut-off strategies and find that the use of the optimal cut-off extends the parameter regime for which secret key can be generated and moreover significantly increases the secret-key rate for a large range of parameters. We also analyse the dependence of the optimal cut-off on different properties of the hardware and find that memory quality highly influences the effectiveness of the cut-off, whereas the influence is small for success probability of entanglement swapping. In addition, our numerical simulations show that for symmetric repeater protocols with evenly spaced nodes, a nonuniform cut-off (different cut-off time in different parts of the repeater chain) does not yield a significant improvement in end-to-end node secret key rate compared to a uniform cut-off.

This chapter is organised as follows. In section 6.1, we describe the class of repeater schemes under study and elaborate on the hardware model used in our simulations. Section 6.2 presents the closed-form expressions and their evaluation algorithms for the waiting time distribution and output quantum states of repeater schemes which include cut-offs. The second part of the chapter, on optimisation of the cut-off, consists of section 6.3, where we provide details on the optimisation procedure, and the results of the numerical optimisation as presented in section 6.4. Section 6.5 ends the chapter with a conclusion.

6.1. PRELIMINARIES

The algorithm we describe in this chapter is applicable to all `tree-shaped`-type quantum repeater protocols, which are constructed from four building blocks or PROTOCOL-UNITS: GENERATE, SWAP, DISTILL, CUT-OFF. We refer to sec. 3.3 in Chapter 3 for an explanation of these building blocks. See Fig. 6.1(a) for a visualisation of the building blocks and Fig. 6.1(b) for an example `tree-shaped`-type protocol, composed of these building blocks.

Note that the class the algorithm from this section is applicable to, is an extension of the class studied in Chapter 5 with the addition of cut-offs. The CUT-OFF building block takes two links as input (not necessarily between the same nodes). It accepts or rejects the two input links depending on a success condition. In case of success, it leaves the two input links untouched and outputs them again. In case of failure, both input links are discarded. In this chapter, we study three different success conditions. In the first two, DIF-TIME-CUT-OFF and MAX-TIME-CUT-OFF, 'success' is declared if respectively the difference or the maximum of the input links' production times does not exceed some prespecified cut-off threshold. In the third strategy, FIDELITY-CUT-OFF, the input states are passed on only if they are both of sufficient quality. This success condition translates to a cut-off on the individual input states' fidelity with a maximally-entangled state (see section 6.1.1).

In the remainder of this section, we first summarise the hardware model we use, which is identical to the one used in Chapter 5 and finish with a brief note on the limitations of our use of cut-offs.

6.1.1. MODEL

We here describe how we model each of the four PROTOCOL-UNITS described in section 6.1, which is identical to the modelling in Chapter 5, except for the newly introduced CUT-OFF unit. For each PROTOCOL-UNIT, we describe the success condition as well as the quantum state that it outputs.

First, we model the fresh entanglement generation (GEN) using schemes which generate links in heralded attempts of duration $L_{\text{internode}}/c$, where $L_{\text{internode}}$ is the internode distance and c is the speed of light in the used transmission medium, e.g. glass fibre [1]. We assume that each attempt is independent and succeeds with constant probability $0 < p_{\text{gen}} \leq 1$. For simplicity, we assume that the nodes are equally spaced with internode distance L_0, so that each attempt in elementary link generation takes duration $\Delta t_0 = L_0/c$, which will be the time unit in our numerical simulation.

We model the elementary link as a Werner state $\rho(w)$ with constant Werner parameter $w = w_0$ [15]:

$$\rho(w) = w\left|\Phi^+\right\rangle\!\left\langle\Phi^+\right| + (1-w)\frac{\mathbb{1}_4}{4} \tag{6.1}$$

where the Bell state

$$|\Phi^+\rangle = (|00\rangle + |11\rangle)/\sqrt{2} \tag{6.2}$$

is a maximally-entangled two-qubit state and

$$\mathbb{1}_4/4 = (|0\rangle\langle 0| + |1\rangle\langle 1|) \otimes (|0\rangle\langle 0| + |1\rangle\langle 1|)/4$$

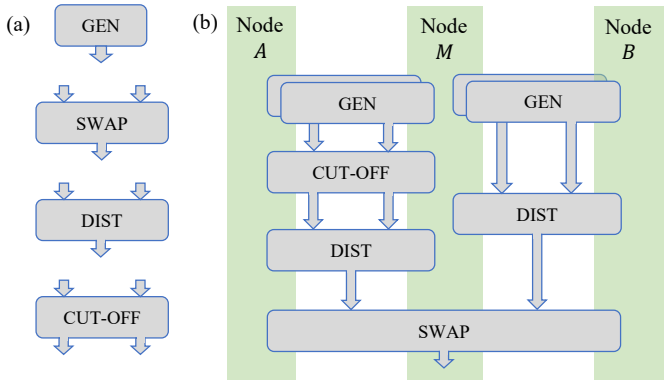


Figure 6.1: In this chapter, we consider the `tree-shaped`-type repeater protocols as introduced in Chapter 3, composed of four different types of PROTOCOL-UNITs. **(a)** The four PROTOCOL-UNITs: elementary-link generation between adjacent nodes (GEN), entanglement swapping for connecting two short-distance links in a single long-distance one (SWAP), entanglement distillation for converting two low-quality links in a single high-quality link (DIST) and discarding two links (CUT-OFF), for example if their generation times differ by more than a pre-specified cut-off time. The repeater chain protocols we consider in this chapter are composed of combinations of the four PROTOCOL-UNITs, provided that each CUT-OFF is succeeded by a SWAP or DIST. The in-/outgoing arrows of each PROTOCOL-UNIT indicate the number of entangled links that the block consumes/produces. **(b)** An example of a composite `tree-shaped`-type protocol on three nodes (end nodes A and B and single repeater M). At the start of the protocol, two fresh elementary links are generated (GEN) in parallel between adjacent nodes A and M and subsequently selected through a CUT-OFF block. The first two links that survive the cut-off are then distilled (DIST) into a single link of higher quality. Asynchronously, the nodes M and B generate (GEN) pairs of links until the distillation (DIST) succeeds. Once distillation on both sides of node M has succeeded, the resulting links $A \leftrightarrow M$ and $M \leftrightarrow B$ are converted via a SWAP into a single entangled link between the end nodes A and B.

is the maximally-mixed state on two qubits. We refer to the parameter w with $0 \leq w \leq 1$ as the Werner parameter. Since a Werner state is completely determined by its Werner parameter, we use the Werner parameter to indicate the quantum state.

Equivalently to the Werner parameter, we will also express the state's quality using the fidelity, which for general density matrices ρ and σ is defined as

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

The fidelity between a Werner state $\rho(w)$ and $\left|\Phi^+\right\rangle\!\left\langle\Phi^+\right|$ equals

$$F = \frac{1+3w}{4}.$$

For the other PROTOCOL-UNITs, the success conditions are summarised in Table 6.1. In short: we model entanglement swapping (SWAP) as succeeding with a constant probability p_{swap}. For entanglement distillation (DIST), we use the BBPSSW protocol [16] which we adapt by bringing the output state back into Werner form. The latter operation does not change the output state's fidelity with the target state $|\Phi^+\rangle$. The success probability p_{dist} of distillation is a function of the input states' Werner parameters (see

Chapter 5 for details). The cut-off (CUT-OFF) success condition depends deterministically on the waiting time or the fidelity of the input links.

The states that any PROTOCOL-UNIT outputs are Werner states at any time of the execution of the protocol. Indeed, a successful entanglement swap or distillation attempt maps Werner states to Werner states (see Chapter 5 for a brief explanation). Also, the CUT-OFF leaves the input states untouched in case of success, thereby outputting Werner states if it got those as input. For each PROTOCOL-UNIT, the Werner parameters of the output links w_{out} are a function of those of the input links and are given in Table 6.1.

In addition to the fact that the PROTOCOL-UNITs change the quantum states they handle, the local quantum memories that are used to store the links are imperfect. In our model, a link with initial Werner parameter w, which lives in memory for time Δt until it is retrieved, decoheres to Werner parameter

$$w_{\text{decayed}} = w \cdot e^{-\Delta t / t_{\text{coh}}}. \tag{6.3}$$

where t_{coh} is the joint coherence time of the two involved memories.

For simplicity, we ignore the time needed for classical communication between the nodes in this chapter as well as the time to perform the local operations. The algorithm we provide can be easily extended to include these features, following the extension described in Chapter 5.

In summary, for a given composite protocol (including the cut-off condition τ or w_{cut} for each CUT-OFF block), the simulation of the entanglement distribution process is determined by 4 hardware parameters: the success probability of elementary link generation p_{gen}, the swap success probability p_{swap}, the Werner parameter of the elementary link w_0 and the memory coherence time t_{coh}.

6.1.2. WAITING TIME AND PRODUCED END-TO-END STATE IN REPEATER SCHEMES USING PROBABILISTIC COMPONENTS

In this chapter, we study the time until the first entangled pair of qubits is generated between the end nodes of the repeater chain (called 'waiting time' from here on) and the state's quality, expressed as its Werner parameter (recall that the end-to-end state is a Werner state, see section 6.1.1). Because the repeater chain protocols we study in this chapter are composed of probabilistic components, both the waiting time and the end-to-end state's Werner parameter are random variables. For an illustration of the random behaviour of the waiting time, see Fig. 6.2. We characterise the quality by the averaged Werner parameters of all states generated at the same time step t. The algorithm we present in this chapter computes the probability distribution $\Pr(T = t)$ of the waiting time T and the average Werner parameter $W(t)$ of the end-to-end state which is delivered at time t.

We finish this section by noting that by considering the average Werner parameter, we ignore the 'history' of a link, resulting in a suboptimal estimation of the fidelity of the states. To see this, consider for example the three-node protocol of Fig. 6.1(b). In this protocol, the following two series of events lead to an output entangled pair between nodes A and B at time $t = 10$: (i) all GEN blocks fail at each timestep $t < 10$ but succeed at time $t = 10$, after which all other PROTOCOL-UNITs also succeed immediately, (ii)

Table 6.1: Overview of success probability and the output Werner parameter for each PROTOCOL-UNIT.

PROTOCOL-UNIT	success probability p	Werner parameter w_{out}
generation (GEN)	p_{gen} (constant)	w_0
entanglement swapping (SWAP)	p_{swap} (constant)	$w'_A \cdot w'_B$
entanglement distillation (DIST)	$p_{\text{dist}} = \frac{1 + w'_A w'_B}{2}$	$\frac{w'_A + w'_B + 4 w'_A w'_B}{6 p_{\text{dist}}}$
DIF-TIME-CUT-OFF	$p_{\text{cut}} = \begin{cases} 1 & \text{if } \lvert t_{\text{A}} - t_{\text{B}} \rvert \leq \tau \\ 0 & \text{otherwise} \end{cases}$	$w'_A,\ w'_B$
FIDELITY-CUT-OFF	$p_{\text{cut}} = \begin{cases} 1 & \text{if } w'_A \geq w_{\text{cut}} \text{ and } w'_B \geq w_{\text{cut}} \\ 0 & \text{otherwise} \end{cases}$	$w'_A,\ w'_B$
MAX-TIME-CUT-OFF	$p_{\text{cut}} = \begin{cases} 1 & \text{if} \max(t_A, t_B) \leq \tau \\ 0 & \text{otherwise} \end{cases}$	$w'_A,\ w'_B$

where $(t_{\text{A}}, w_{\text{A}})$ and $(t_{\text{B}}, w_{\text{B}})$ are the waiting time and Werner parameter of the links A and B provided as input to the PROTOCOL-UNIT. Parameters τ and w_{cut} are the cut-off thresholds on time and Werner parameter, respectively. The primed notation denotes Werner parameter with decay in (6.3) applied to the link that waits until the other is finished: $w'_X = w_X \cdot e^{-|t_{\text{A}} - t_{\text{B}}|/t_{\text{coh}}}$ if $t_X = \min(t_{\text{A}}, t_{\text{B}})$ and $w'_X = w_X$ otherwise, for $X \in \{A, B\}$. For an explanation of the different PROTOCOL-UNITs, see section 6.1.

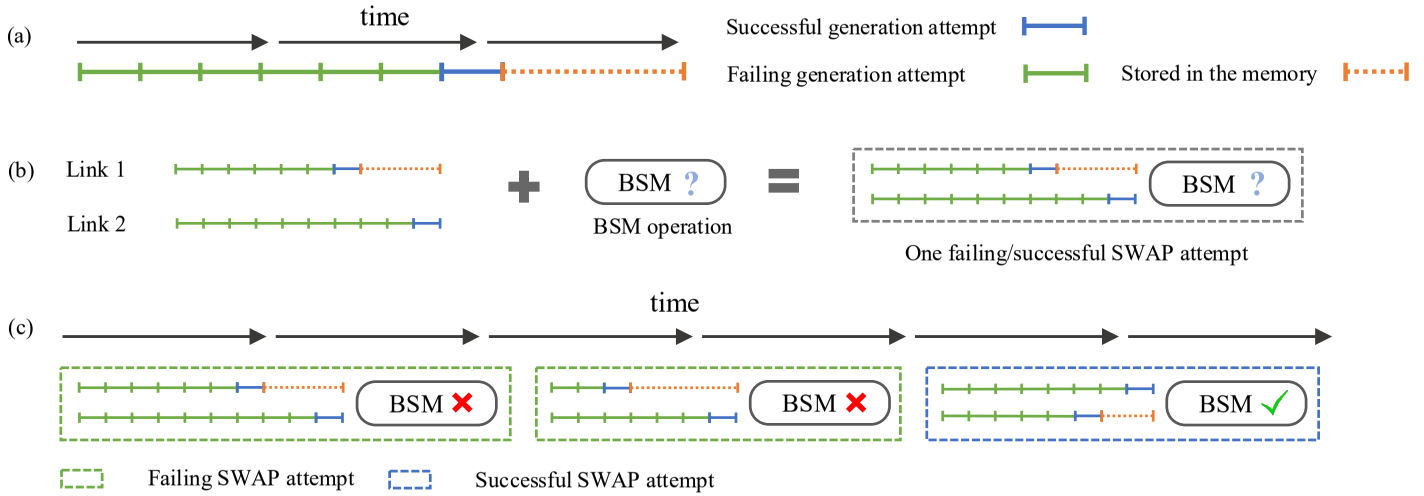


Figure 6.2: Visualisation of the waiting time until end-to-end entanglement is delivered for a 3-node repeater chain. The repeater scheme consists of the generation of two elementary links, followed by an entanglement swap on the two links. **(a)** A single link is generated in fixed-duration attempts, which succeed probabilistically and thus may fail (green line segment), after which generation is re-attempted until success (blue line segment). After that, the link is stored until it is consumed (dotted orange line segment). **(b)** A run of the 3-node protocol until the first swap attempt, which consists of first preparing two input links in parallel, followed by a Bell state measurement (BSM). The link that is generated earlier than the other needs to wait in the memory (link 1 in the figure, the 'waiting' is indicated by the dotted orange line). While waiting, the earlier link's quality decreases due to decoherence. The total waiting time before the BSM equals the maximum of the generation times of the two links. The BSM operation can fail, in which case the two links are lost and need to be regenerated. **(c)** A full run of the 3-node protocol, consisting of failed entanglement swaps (green dashed box) on fresh links until the first successful swap (blue dashed box). The total waiting time is the sum of the waiting times for the parallel generation of each pair of elementary links, up to and including the first successful swap.

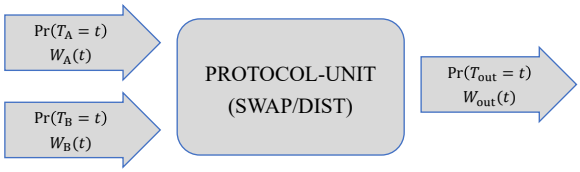


Figure 6.3: The workflow of the algorithm for one PROTOCOL-UNIT (SWAP or DIST). It takes the waiting time distribution and Werner parameter of the two input links and computes those of the output.

the PROTOCOL-UNITs between A and M all succeed at time $t = 1$, while the GEN blocks between M and B succeed at time $t = 10$, followed by all other remaining PROTOCOL-UNITs also succeeding at time $t = 10$. In case (i), no entanglement has waited in memory, whereas in case (ii), the produced link between A and M has waited 10 timesteps and decohered in that time. By keeping track of the timestamps at which the several PROTOCOL-UNITs succeeded, one could distinguish these two scenarios. Since the resulting fidelity estimation computation is rather complex and in this chapter, we focus on quantifying the effect of a cut-off, we leave such advanced fidelity estimation for future work.

6.2. COMPUTING THE WAITING TIME DISTRIBUTION AND THE OUTPUT WERNER PARAMETER

In this section, we present closed-form expressions of the waiting time probability distribution and Werner parameter of the output links for each PROTOCOL-UNIT, as function of waiting time distribution and Werner parameter of its input links. Expressions for a composite protocol are obtained by iterative application over the PROTOCOL-UNITs that the protocol consists of. These expressions naturally lead to an algorithm for their evaluation, which we also present in this section.

Closed-form expressions for GEN and SWAP were already obtained Chapter 5, where we explicitly mentioned that the approach does not generalise straightforwardly to DIST. Here, we include DIST and even CUT-OFF, provided the latter is succeeded by SWAP or DIST. The novel idea is to use separate expressions for the waiting time probability distribution of a successful and failed attempt. We then express the total waiting time distribution and the Werner parameter as those of the successful attempt averaged by the occurrence probability of all possible sequences of failed attempts, where the weighted average is efficiently computed using convolution. As an additional benefit, the evaluation algorithm for SWAP is faster than the one presented in Chapter 5.

In the following, we first derive general closed-form expressions for the waiting time distribution and Werner parameter of one PROTOCOL-UNIT in section 6.2.1. We then give specific expressions for each PROTOCOL-UNIT individually in sections 6.2.2 to 6.2.5. In the last section (section 6.2.6), we show how these expressions can be converted into an efficient algorithm. We also explain how to modify the closed-form expressions using the discrete Fourier transform, motivated by its use in [17, 18]. These modified expressions lead to an even faster algorithm for computing the waiting time and Werner parameter,

which we provide in Appendix 6.6.2. We denote the random variables of the waiting time and average Werner parameter as T and $W(t)$, with subscript A and B for the input links and 'out' for the output link (see Fig. 6.3).

6.2.1. GENERAL CLOSED-FORM EXPRESSIONS FOR WAITING TIME AND PRODUCED STATES FOR ALL PROTOCOL-UNITS

RANDOM VARIABLE EXPRESSION FOR THE WAITING TIME OF PROTOCOL-UNITS

We start by presenting an expression for the random variable T_{out}. To study the waiting time distribution, we divide the total waiting time into the waiting time for each attempt. An attempt can fail or succeed and it repeats until the first successful attempt occurs (see Fig. 6.2). The total waiting time T_{out} is given by

$$T_{\text{out}} = \sum_{i=1}^{K} M^{(i)} \tag{6.4}$$

where $M^{(i)}$ are i.i.d. random variables characterising the waiting time of each attempt and therefore each is a function of the waiting time of two input links T_{A}, T_{B}. For example, for SWAP, we have $M = \max(T_{\text{A}}, T_{\text{B}})$, *i.e.* we need to wait until both links are ready to perform the operation. K is the number of attempts we need until the first successful attempt occurs, which is also a random variable.

The success or failure of one attempt is characterised by a probability p. The success probability p of one attempt is independent of that of others and is given by $p = p(t_{\text{A}}, t_{\text{B}}, w_{\text{A}}, w_{\text{B}})$ (Table 6.1). We reduce the Werner parameter dependence to time dependence by plugging in $w_A = W_{\text{A}}(t_A)$ and $w_B = W_{\text{B}}(t_{\text{B}})$. Hence, we write $p(t_{\text{A}}, t_{\text{B}})$ in the rest of this section.

The time dependence of p implies that, in general, K is correlated to $M^{(j)}$. To make this correlation between K and M in (6.4) explicit, we introduce a random variable Y. Y denotes the binary random variable describing success (1) or failure (0) of a single attempt, subjected to the success probability $p(t_{\text{A}}, t_{\text{B}})$. The time-dependent success probability can be understood as the success probability with given waiting time t_{A}, t_{B} of the input links:

$$p(t_{\text{A}}, t_{\text{B}}) = \Pr(Y = 1 | T_{\text{A}} = t_{\text{A}}, T_{\text{B}} = t_{\text{B}}).$$

We then rewrite (6.4) with a sum over all possible number of attempts weighted by its occurrence probability[3]:

$$T_{\text{out}} = \sum_{k=1}^{\infty} \left\{ \left(Y^{(k)} \prod_{j=1}^{k-1} \left(1 - Y^{(j)}\right) \right) \cdot \sum_{i=1}^{k} M^{(i)} \right\}. \tag{6.5}$$

The expression in round brackets evaluates to 1 precisely if $Y^{(k)} = 1$ and $Y^{(j)} = 0$ for all $j < k$, and to 0 in all other cases. This factor thus makes that only the sum $\sum_{i=1}^{k} M^{(k)}$ is taken for which k is the first successful attempt. Notice that $Y^{(j)}$ and $M^{(i)}$ are correlated for all $i = j$ because they describe the same attempts. In the next section, we go further to compute the probability distribution of T_{out}.

A CLOSED-FORM EXPRESSION FOR THE WAITING TIME DISTRIBUTION

In the following, we give an expression of the waiting time distribution $\Pr(T_{\text{out}} = t)$ for one PROTOCOL-UNIT.

We consider the generation time of a successful or failed attempt separately and use the joint distribution of M and Y. We define the joint distribution that one attempt succeeds/fails and takes time t as

$$\begin{aligned} P_{\text{s}}(t) &:= \Pr(M = t, Y = 1) \\ &= \sum_{t_{\text{A}}, t_{\text{B}}: \max(t_{\text{A}}, t_{\text{B}}) = t} \Pr(T_{\text{A}} = t_{\text{A}}, T_{\text{B}} = t_{\text{B}}) \cdot p(t_{\text{A}}, t_{\text{B}}), \end{aligned} \tag{6.6}$$

$$\begin{aligned} P_{\text{f}}(t) &:= \Pr(M = t, Y = 0) \\ &= \sum_{t_{\text{A}}, t_{\text{B}}: \max(t_{\text{A}}, t_{\text{B}}) = t} \Pr(T_{\text{A}} = t_{\text{A}}, T_{\text{B}} = t_{\text{B}}) \cdot [1 - p](t_{\text{A}}, t_{\text{B}}). \end{aligned} \tag{6.7}$$

In the above equation, we iterate over all possible combinations of the input links' generation time t_{A}, t_{B} that leads to a waiting time t for this attempt.

With the definition (6.6) and (6.7), the sum of the waiting time for all attempts can be obtained by

$$\boxed{\Pr(T_{\text{out}} = t) = \sum_{k=1}^{\infty} \left[\left(\overset{k-1}{\underset{j=1}{\ast}} P_{\text{f}}^{(j)} \right) * P_{\text{s}} \right](t)} \tag{6.8}$$

where $*$ is the notation for convolution and the sum over k considers all the possible numbers of attempts. The notation $\overset{k-1}{\underset{j=1}{\ast}} P_{\text{f}}^{(j)}$ represents the convolution of $k-1$ independent functions P_{f}. In the above equation, the discrete linear convolution is defined by

$$[f_1 * f_2](t) = \sum_{t'=0}^{t} f_1(t - t') \cdot f_2(t'). \tag{6.9}$$

If f_1, f_2 describe two probability distributions of two random variables, their convolution is the distribution of the sum of those two random variables. However, neither P_{f} or P_{s} characterises a random variable since they are joint distributions including Y. That is to say, P_{s} and P_{f} do not sum up to 1. Instead, we have

$$\sum_t P_{\text{f}}(t) + \sum_t P_{\text{s}}(t) = 1.$$

Therefore, the convolution here cannot be simply interpreted as a sum of two random variables. Instead, it is the summed waiting time conditioned on the success/failure of each attempt.

As we will show in sec. 6.2.6, eq. (6.8) is sufficient for the derivation of the main algorithm for computing the probability distribution of T_{out} we present in this chapter. The algorithm's runtime is partially determined by the sum and the convolution in the summand in eq. (6.8). Fortunately, these can be eliminated by the use of the discrete Fourier

transform, resulting in a faster alternative algorithm. Below, we use the Fourier transform to derive an equivalent expression to eq. (6.8). The alternative algorithm is given in Appendix 6.6.2

Since the discrete Fourier transform acts on a finite sequence of numbers, we first truncate the probability distribution at a fixed time L, *i.e.* we obtain the finite sequence $\{\Pr(T_{\text{out}} = t) | t = 0, 1, 2, \ldots, L\}$. If $\vec{x} := x_0, x_1, \ldots, x_{L-1}$ is a sequence of complex numbers, then its Fourier transform $\mathscr{F}(\vec{x})$ is the sequence $y_0, y_1, \ldots, y_{L-1}$ given by

$$y_j = \sum_{k=0}^{L-1} x_k \cdot \exp\left(-2\pi \mathrm{i} \cdot j \cdot k / L\right) \tag{6.10}$$

where i is the complex unit. The Fourier transform is a linear map and moreover it converts convolutions into element-wise multiplication, *i.e.* $\mathscr{F}(\vec{x} * \vec{x'}) = \mathscr{F}(\vec{x}) \cdot \mathscr{F}(\vec{x'})$. As a consequence, taking the Fourier transform of both sides of eq. (6.8) yields

$$\mathscr{F}\left[\Pr(T_{\text{out}} = t)\right] = \sum_{k=1}^{\infty} \left[\left(\prod_{j=1}^{k-1} \mathscr{F}(P_{\text{f}})^{(j)} \right) \cdot \mathscr{F}[P_{\text{s}}] \right](t).$$

Because, $P_{\text{f}}^{(j)}$ are identical distribution for all j, we use the identity $\sum_{k=1}^{\infty} x^{(k-1)} = 1/(1-x)$ to obtain

$$\Pr(T_{\text{out}} = t) = \mathscr{F}^{-1}\left[\frac{\mathscr{F}[P_{\text{s}}]}{1 - \mathscr{F}[P_{\text{f}}]} \right](t). \tag{6.11}$$

6

A CLOSED-FORM EXPRESSION FOR THE WERNER PARAMETER

Here, we derive the expression for the Werner parameter $W_{\text{out}}(t)$.

To arrive at $W_{\text{out}}(t)$, we first compute the average Werner parameter of the output link of one attempt, given that it succeeds and finishes at time t:

$$W_{\text{s}}(t) = \frac{\sum_{t_{\text{A}}, t_{\text{B}}: \max(t_{\text{A}}, t_{\text{B}}) = t} \Pr(T_{\text{A}} = t_{\text{A}}, T_{\text{B}} = t_{\text{B}}) \cdot [p \cdot w_{\text{out}}](t_{\text{A}}, t_{\text{B}})}{P_{\text{s}}(t)}. \tag{6.12}$$

Here, w_{out} is the Werner parameter of the output link of a successful attempt and p the success probability (Table 6.1). We again simplify the notation with $w_{\text{out}}(t_{\text{A}}, t_{\text{B}}) = w_{\text{out}}(t_{\text{A}}, t_{\text{B}}, W_{\text{A}}(t_{\text{A}}), W_{\text{B}}(t_{\text{B}}))$.

Next, we take a weighted average of W_{s}' over all possible sequences of failed attempts, followed by a single successful attempt:

$$\boxed{W_{\text{out}}(t) = \frac{\sum_{k=1}^{\infty} \left[\left(\overset{k-1}{\underset{j=1}{\ast}} P_{\text{f}} \right) * (P_{\text{s}} \cdot W_{\text{s}}) \right](t)}{\Pr(T_{\text{out}} = t)}}. \tag{6.13}$$

where $\overset{k-1}{\underset{j=1}{\ast}} P_{\text{f}}^{(j)}$ computes the waiting time distribution of $k-1$ failed attempts and the additional convolution is the weighted average.

For eq. (6.8), which is an expression for the probability distribution of T_{out}, we obtained a more compact equivalent, eq. (6.11), by moving to Fourier space. By an analogous derivation, we can get a more compact expression for W_{out} than eq. (6.13):

$$W_{\text{out}}(t) = \mathscr{F}^{-1}\left[\frac{\mathscr{F}[P_{\text{s}} \cdot W_{\text{s}}]}{1 - \mathscr{F}[P_{\text{f}}]}\right] \frac{1}{\Pr(T_{\text{out}} = t)}(t). \tag{6.14}$$

6.2.2. SPECIFIC CASE: GEN

We give here the expression for PROTOCOL-UNIT GEN. Since GEN does not have input links, the output does not rely on the expression introduced in the section 6.2.1. Because one attempt in GEN takes one time step and the success probability p_{gen} is a constant, the waiting time can be described by a geometric distribution

$$\Pr(T_{\text{out}} = t) = p_{\text{gen}}(1 - p_{\text{gen}})^{t-1}.$$

The output state is a Werner state with Werner parameter w_0 as described in section 6.1.1.

6.2.3. SPECIFIC CASE: SWAP

For entanglement swap, since p_{swap} is constant, Y is not correlated with M. As a result, P_{s} and P_{f} differ only by a constant coefficient (see (6.6) and (6.7)). Therefore, we can factor the constant out and get

$$\Pr(T_{\text{out}} = t) = \sum_{k=1}^{\infty} p_{\text{swap}}(1 - p_{\text{swap}})^{k-1} \left[\overset{k}{\underset{j=1}{\ast}} m\right]$$

where

$$m(t) \coloneqq Pr(M = t) = \sum_{t_{\text{A}}, t_{\text{B}}: \max(t_{\text{A}}, t_{\text{B}}) = t} \Pr(T_{\text{A}} = t_{\text{A}}, T_{\text{B}} = t_{\text{B}}).$$

This is exactly the geometric compound distribution obtained in Chapter 5.

For the Werner parameter, we can directly use (6.13) and obtain

$$W_{\text{out}} = \sum_{k=1}^{\infty} p_{\text{swap}}(1 - p_{\text{swap}})^{k-1} \left[\left(\overset{k-1}{\underset{j=1}{\ast}} m\right) * \left(m \cdot W'_{\text{s}}\right)\right]. \tag{6.15}$$

Compared to the expression in Chapter 5, this expression replaces the iteration over all pair of possible input Werner parameters for each k by convolution.

Both expressions above can also be written in Fourier space by substituting $P_{\text{s}} = p_{\text{swap}} m(t)$ and $P_{\text{f}} = (1 - p_{\text{swap}}) m(t)$ in (6.11) and (6.14).

6.2.4. SPECIFIC CASE: DIST

For entanglement distillation, the success probability depends on the Werner parameters. As discussed in section 6.2.1, we can compute T_{out} and W_{out} because we iterate over all possible combinations of t_{A} and t_{B} and we use $W(t)$ to reduce the dependence on Werner parameters to the dependence on the waiting time. The calculation goes as follows. First, we compute P_{f} and P_{s} using $p(t_A, t_B) = p_{\text{dist}}(W(t_A), W(t_B))$ (Table 6.1). Then, we plug in P_{f} and P_{s} in (6.8) to compute T_{out}. Finally, W_{out} can be calculated similarly using Table 6.1, (6.12) and (6.13).

6.2.5. SPECIFIC CASE: CUT-OFF

CUT-OFF selects the input links and accepts them if the cut-off condition described in section 6.1.1 is fulfilled. We consider only the case where CUT-OFF is followed by SWAP or DIST, so that the two blocks together output a single entangled link.

THE WAITING TIME DISTRIBUTION

We define a new binary variable Y_{cut} representing whether the cut-off condition is fulfilled. The corresponding success probability is described by p_{cut} in Table 6.1. In addition, we also define the waiting time of one cut-off attempt as Z, in contrast to M for a swap or distillation attempt. For CUT-OFF, we need to distinguish the waiting time of a successful and a failed attempt. In the case of success, we always have $Z_{\text{s}} = \max(T_{\text{A}}, T_{\text{B}})$, *i.e.* we wait until two links are produced. However, in the case of failure, the waiting time is different for different cut-off strategies. With the notation $Z_{\text{f}} = t_{\text{fail}}(T_{\text{A}}, T_{\text{B}})$, we have the following: for DIF-TIME-CUT-OFF, $t_{\text{fail}}(T_{\text{A}}, T_{\text{B}}) = \min(T_{\text{A}}, T_{\text{B}}) + \tau$, because there is no need to wait for the second link longer than the cut-off threshold. For MAX-TIME-CUT-OFF, $t_{\text{fail}}(T_{\text{A}}, T_{\text{B}})$ is the constant τ, *i.e.* the maximal allowed waiting time. For FIDELITY-CUT-OFF, it is $t_{\text{fail}}(T_{\text{A}}, T_{\text{B}}) = \max(T_{\text{A}}, T_{\text{B}})$.

Similar as the nested structure shown in Fig. 6.2, a swap or distillation attempt is now composed of several cut-off attempts. We can write its waiting time M as

$$M = \sum_k \left\{ \left[Y_{\text{cut}}^{(k)} \prod_{j=1}^{k-1} \left(1 - Y_{\text{cut}}^{(j)}\right) \right] \cdot \left[Z_{\text{s}}^{(k)} + \sum_{i=1}^{k-1} \left(Z_{\text{f}}^{(i)}\right) \right] \right\}$$

This expression will replace $M = \max(T_{\text{A}}, T_{\text{B}})$ used in (6.5). For $\tau = \infty$ or $w_{\text{cut}} = 0$, *i.e.* no cut-off, Y_{cut} is always 1. Therefore, $k = 1$ is the only surviving term and the two expressions coincide.

To calculate the waiting time distribution, we need three joint distributions: P'_{f} for unsuccessful input link preparation because of the cut-off, $P'_{\text{s,f}}$ for successful preparation but unsuccessful swap/distillation and $P'_{\text{s,s}}$ for both successful:

$$\begin{aligned} P'_{\text{f}}(t) &= \Pr(M = t, Y_{\text{cut}} = 0) \\ &= \sum_{t_{\text{A}}, t_{\text{B}}: t_{\text{fail}}(t_{\text{A}}, t_{\text{B}}) = t} \Pr(T_{\text{A}} = t_{\text{A}}, T_{\text{B}} = t_{\text{B}}) \cdot [1 - p_{\text{cut}}](T_{\text{A}}, T_{\text{B}}) \end{aligned}$$

$$\begin{aligned} P'_{\text{s,f}}(t) &= \Pr(M = t, Y_{\text{cut}} = 1, Y = 0) \\ &= \sum_{t_{\text{A}}, t_{\text{B}}: \max(t_{\text{A}}, t_{\text{B}}) = t} \Pr(T_{\text{A}} = t_{\text{A}}, T_{\text{B}} = t_{\text{B}}) \cdot [p_{\text{cut}} \cdot (1 - p)](t_{\text{A}}, t_{\text{B}}) \end{aligned}$$

$$\begin{aligned} P'_{\text{s,s}}(t) &= \Pr(M = t, Y_{\text{cut}} = 1, Y = 1) \\ &= \sum_{t_{\text{A}}, t_{\text{B}}: \max(t_{\text{A}}, t_{\text{B}}) = t} \Pr(T_{\text{A}} = t_{\text{A}}, T_{\text{B}} = t_{\text{B}}) \cdot [p_{\text{cut}} \cdot p](t_{\text{A}}, t_{\text{B}}). \end{aligned}$$

The prime notation indicates that they describe the waiting time of one attempt in CUT-OFF, in contrast to one attempt in swap or distillation.

For one attempt in swap/distillation with time-out, we then get similarly to (6.8)

$$P_{\mathrm{s}}(t) = \Pr(M = t, Y = 1) = \sum_k \left[\left(\overset{k-1}{\underset{j=1}{\circledast}} P_{\mathrm{f}}^{\prime(j)} \right) * P'_{\mathrm{s,s}} \right](t)$$

$$P_{\mathrm{f}}(t) = \Pr(M = t, Y = 0) = \sum_k \left[\left(\overset{k-1}{\underset{j=1}{\circledast}} P_{\mathrm{f}}^{\prime(j)} \right) * P'_{\mathrm{s,f}} \right](t)$$

as well as the expressions in Fourier space analogous to (6.11)

$$P_{\mathrm{s}}(t) = \Pr(M = t, Y = 1) = \mathscr{F}^{-1}\left[\frac{\mathscr{F}[P'_{\mathrm{s,s}}]}{1 - \mathscr{F}[P'_{\mathrm{f}}]} \right],$$

$$P_{\mathrm{f}}(t) = \Pr(M = t, Y = 0) = \mathscr{F}^{-1}\left[\frac{\mathscr{F}[P'_{\mathrm{s,f}}]}{1 - \mathscr{F}[P'_{\mathrm{f}}]} \right].$$

The total waiting time then follows by substituting the expressions for P_{f} and P_{s} above in (6.8) or (6.11).

For entanglement swap, *i.e.* constant success probability p_{swap}, simplification can be made for this calculation. In this special case, $P'_{\mathrm{s,f}}$ and $P'_{\mathrm{s,s}}$ differ only by a constant and the same holds for P_{s} and P_{f}.

6

THE WERNER PARAMETER

For the Werner parameter, we now need three steps.

We start from calculating the resulting Werner parameter of a swap or distillation for the very last preparation attempt where $Y_{\mathrm{cut}} = Y = 1$. It is denoted by W'_{s} and we only need to replace P_{s} by $P'_{\mathrm{s,s}}$ and $p \cdot w_{\mathrm{out}}$ by $p_{\mathrm{cut}} \cdot p \cdot w_{\mathrm{out}}$ in (6.12).

Next, we compute the Werner parameter $W_{\mathrm{s}}(t)$ as a function of time t that includes the failed cut-off attempts, in analogue to the derivation of eq. (6.13). $W_{\mathrm{s}}(t)$ is the Werner parameter that the pair of output links of CUT-OFF will produce, given that the swap or distillation operation following is successful:

$$W_{\mathrm{s}}(t) = \frac{\sum_{k=1}^{\infty} \left[\left(\overset{k-1}{\underset{j=1}{\circledast}} P'_{\mathrm{f}} \right) * (P'_{\mathrm{s,s}} \cdot W'_{\mathrm{s}}) \right](t)}{P_{\mathrm{s}}(t)}.$$

Finally, we consider the time consumed by failed attempts in SWAP or DIST and obtain

$$W_{\mathrm{out}}(t) = \frac{\sum_{k=1}^{\infty} \left[\left(\overset{k-1}{\underset{j=1}{\circledast}} P_{\mathrm{f}} \right) * (P_{\mathrm{s}} \cdot W_{\mathrm{s}}) \right](t)}{\Pr(T_{\mathrm{out}} = t)}.$$

Using the Fourier transform, the two expressions above become

$$W_{\mathrm{s}}(t) = \mathscr{F}^{-1}\left[\frac{\mathscr{F}[P'_{\mathrm{s,s}} \cdot W'_{\mathrm{s}}]}{1 - \mathscr{F}[P'_{\mathrm{f}}]} \right] \frac{1}{P_{\mathrm{s}}},$$

$$W_{\mathrm{out}}(t) = \mathscr{F}^{-1}\left[\frac{\mathscr{F}[P_{\mathrm{s}} \cdot W_{\mathrm{s}}]}{1 - \mathscr{F}[P_{\mathrm{f}}]} \right] \frac{1}{\Pr(T_{\mathrm{out}} = t)}.$$

6.2.6. CONVERTING THE CLOSED-FORM EXPRESSIONS INTO AN EFFICIENT ALGORITHM

In the sections above, we presented closed-form expressions for T_{out} and W_{out} for each of the four PROTOCOL-UNITs, as a function of waiting time distribution and Werner parameter of the input links. In order to convert these expressions into an algorithm, we take the same approach as in Chapter 5 and cap the infinite sum in (6.8) and (6.13) by a pre-specified truncation time t_{trunc}. This yields a correct $\Pr(T_{\text{out}} = t)$ and $W_{\text{out}}(t)$ for $t \in \{1, \ldots, t_{\text{trunc}}\}$ since in each of the expressions with an infinite sum above, $\Pr(T_{\text{out}} = t)$ and $W_{\text{out}}(t)$ are only dependent on waiting time and Werner parameter of input links produced at time $t' \leq t$.

We now show that the algorithm scales polynomially in terms of t_{trunc}. To analyse the complexity, we divide the algorithm into two parts: computing the distribution for one attempt, *i.e.* the iteration over all possible values of T_A, T_B ((6.6), (6.7) and (6.12)) and for the whole PROTOCOL-UNIT((6.8) and (6.13)).

The complexity for the first part is $\mathcal{O}(t_{\text{trunc}}^2)$ since it iterates over two discrete random variables up to t_{trunc}. For the second part, because we need at least one time step in each attempt, *i.e.* $\Pr(T = 0) = 0$, only the first t_{trunc} convolutions will have non-zero contribution. We can perform the convolution iteratively for each k using at most t_{trunc} convolutions. The complexity of one convolution with fast Fourier transform (FFT) is $\mathcal{O}(t_{\text{trunc}} \log t_{\text{trunc}})$ [19]. Thus, the complexity of the second part scales as $\mathcal{O}(t_{\text{trunc}}^2 \log t_{\text{trunc}})$. The overall complexity, therefore, is $\mathcal{O}(t_{\text{trunc}}^2 \log t_{\text{trunc}})$.

In appendix 6.6.2, we show that with further simplification of (6.6) and (6.7) as well as expressions in Fourier space (equations (6.11) and (6.14)), the complexity can be reduced to $\mathcal{O}(t_{\text{trunc}} \log t_{\text{trunc}})$, with an exponentially vanishing error.

The preceding discussion shows that the algorithm is efficient as a function of the truncation time. However, for fixed truncation time, the probability mass captured by the algorithm decreases as the number of nodes increases. For protocols without cut-off, variations of the arguments in Chapter 5 would allow to prove that the algorithm introduced here is also efficient for fixed probability mass. Unfortunately, the arguments do not translate to protocols with cut-off. This is because for these protocols, the truncation time that covers a fixed probability mass can grow exponentially with the number of nodes, *i.e.* such an algorithm can not exist.

As an example, consider a nested protocol on 2^n repeater segments ($n = 0, 1, 2, \ldots$), which for $n = 1$ consists of a GEN block only, and for each additional level $n > 1$, each pair of adjacent links is connected by a CUT-OFF followed by a SWAP. We set $\tau = 0$ for each cut-off, *i.e.* all elementary links need to be generated at the same time and also all entanglement swaps should succeed at the first attempt for the links to survive all the cut-offs. Since 2^n elementary links need to be generated and the protocol consists of $2^n - 1$ swaps, the probability of successful end-to-end entanglement before time t equals $1 - (1 - p)^t$ with $p = p_{\text{gen}}^{N-1} \cdot p_{\text{swap}}^{N-2}$, *i.e.* decreases exponentially in the number of nodes $N = 2^n + 1$.

6.3. OPTIMISATION

In this section, we describe the details of our optimisation over cut-offs, including the figure of merit and optimisation method.

In our numerical study, we use the secret-key rate of the BB84 protocol [20] as a figure of merit to assess the performance of composite repeater protocols. We compute the secret-key rate R as the secret-key fraction divided by the average waiting time

$$R = \frac{r}{\bar{T}}. \tag{6.16}$$

The secret-key fraction r describes the amount of secret key that can be extracted from the generated entanglement and is given by [21, 22]

$$r(w) = \max\{0, 1 - h[e_X(w)] - h[e_Z(w)]\}$$

where $h(p) = -p\log_2(p) - (1-p)\log_2(1-p)$ is the binary entropy function and e_X (e_Z) is the quantum bit error rate in the X (Z) basis. Since the quantum states tracked by our algorithm are Werner states at any point in the execution of the composite repeater protocol (see section 6.1), the quantum bit error rate can be expressed as function of the end-to-end state's Werner parameter:

$$e_Z(w) = \langle 01|\rho(w)|01\rangle + \langle 10|\rho(w)|10\rangle = \frac{1-w}{2}$$

for a Werner state $\rho(w)$ defined in (6.1). The same result holds for e_X because of the symmetry of the Werner state. In section 6.6.3, we detail how we compute the secret-key rate with truncated waiting time distribution and Werner parameter obtained from the algorithm in section 6.2.6.

Since we have discrete time steps, we need an optimisation algorithm which is compatible with a discrete search space. We choose the differential evolution algorithm implemented in the SciPy-optimisation library of the Python programming language [23, 24].

6.4. NUMERICAL RESULTS

In this section, we optimise over repeater protocols with cut-offs in order to maximise the rate at which secret key can be extracted from the produced end-to-end entanglement. First, we use our algorithm from section 6.2 and the DIF-TIME-CUT-OFF strategy (section 6.1) to study the effect of the cut-off on the waiting time and fidelity and show that the use of a cut-off boosts secret-key rate. We then extend our study to two other cut-off strategies, MAX-TIME-CUT-OFF and FIDELITY-CUT-OFF, and compare their performance. For all three cut-off strategies, we observe that the resulting repeater protocols produce secret key at significantly higher rates than their no-cut-off alternatives. Finally, we focus on the DIF-TIME-CUT-OFF strategy and analyse the sensitivity of the optimal cut-off threshold with respect to the hardware parameters.

We investigate repeater protocols with 3 nesting levels where at each nesting level the range of entanglement is doubled by an entanglement swap. The protocol thus spans

$2^3 = 8$ segments ($8+1=9$ nodes). Each entanglement swap operation is preceded by a cut-off, *i.e.* the scheme is of the form

$$\text{GEN} \rightarrow (\rightarrow \text{CUT-OFF} \rightarrow \text{SWAP})^3. \tag{6.17}$$

The numerical results in this section were obtained using our open-source implementation [25] of the algorithm from section 6.2 on consumer-market hardware (Intel i7-8700 CPU). We validated correctness of the implementation by comparison with an extended version of the Monte Carlo algorithm from Chapter 5 (see Fig. 6.4 and appendix 6.6.1 for details).

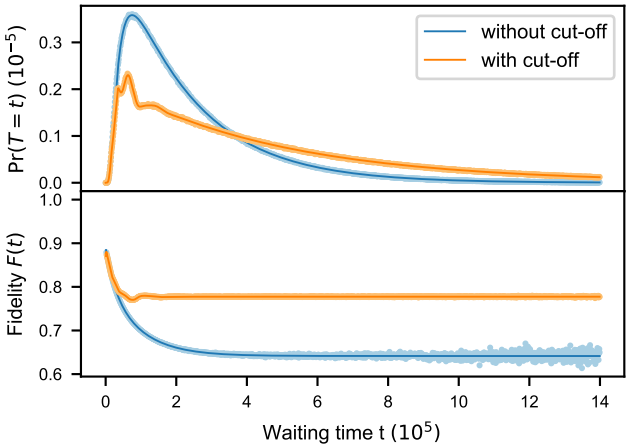


Figure 6.4: The probability distribution of the waiting time T and the average fidelity $F(t)$ of the end-to-end link for a protocol with and without a cut-off on entanglements' production time differences (solid lines) for a 9-node repeater protocol of the form as in (6.17) (unit of time is the attempt duration of elementary link generation, L_0/c). We observe that the fidelity increases for most times t while the probability that the link is produced at time t shifts to larger t, indicating a longer waiting time. The secret-key rates computed from the data are 0 (without cut-off) and $0.32 \cdot 10^{-7}$ (with cut-off). The parameters used are $p_{\text{gen}} = 10^{-4}$, $p_{\text{swap}} = 0.5$, $w_0 = 0.98$, $t_{\text{coh}} = 4 \cdot 10^5$ and the cut-offs for the three nesting levels are $\tau = (1.7, 3.2, 5.5) \cdot 10^4$ (in increasing order of number of segments spanned by the CUT-OFF block). Computation time ≈ 20 seconds for $3 \cdot 10^6$ time steps. We observe good agreement with a Monte Carlo algorithm (dots), which we use for validating the correctness of our implementation (see appendix 6.6.1 for details).

6.4.1. EFFECT OF DIF-TIME-CUT-OFF ON THE WAITING TIME AND FIDELITY

We start by investigating the DIF-TIME-CUT-OFF strategy, where links are discarded if their production times differ by more than a predetermined threshold τ. We compute waiting time and average fidelity for a particular choice of the cut-off threshold at each of the three levels and compare it with the protocol without cut-off (cut-off duration $\tau = \infty$ at each nesting level), see Fig. 6.4. We observe that the cut-off increases fidelity at the cost of longer waiting time, as one would intuitively expect. We further quantify the time-fidelity trade-off for a range of cut-offs in Fig. 6.5. For maximising the secret key rate, we observe a single optimal choice of the cut-off threshold τ.

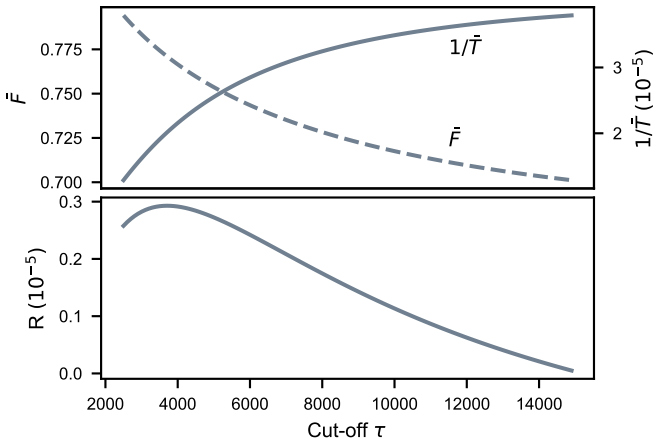


Figure 6.5: Influence of choice of cut-off on average waiting time, average fidelity and secret-key rate for repeater protocols of the form (6.17) where the cut-off strategy is DIF-TIME-CUT-OFF. **(Top)** Increasing the cut-off yields higher average generation rate (reciprocal of average waiting time $\bar{T}$) but lower average fidelity $\bar{F}$. **(Bottom)** The secret key rate R as a function of the cut-off time. The used parameters are $p_{\text{gen}} = 10^{-3}$, $p_{\text{swap}} = 0.5$, $w_0 = 0.98$ and $t_{\text{coh}} = 4 \cdot 10^4$. The chosen truncation time is $5 \cdot 10^5$. The cut-off time is chosen identical for all three swap levels. Unit of time is the attempt duration of elementary link generation.

6

6.4.2. EXTENSION TO OTHER CUT-OFF STRATEGIES

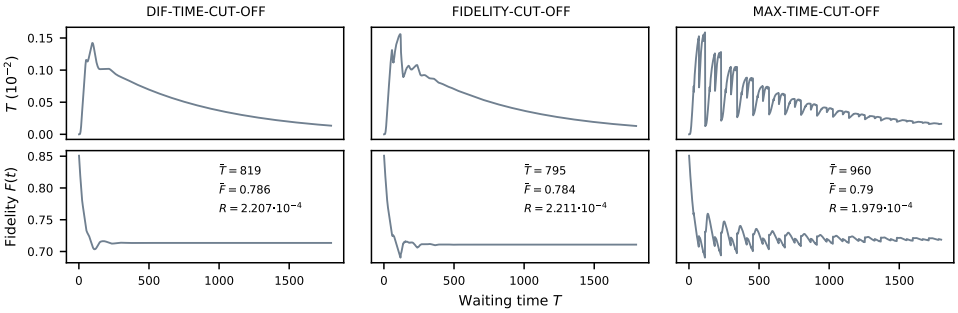


Figure 6.6: Comparison between three different cut-off strategies: cut-off on the difference of entanglements' production time (DIF-TIME-CUT-OFF), the fidelity (FIDELITY-CUT-OFF) and the total waiting time (MAX-TIME-CUT-OFF, see sec. 6.1 for definitions). For each strategy, we find the optimised cut-off threshold when applied to the 9-node repeater chain protocol from (6.17) with parameters: $p_{\text{gen}} = 0.1$, $p_{\text{swap}} = 0.4$, $w_0 = 0.98$, $t_{\text{coh}} = 600$. For each cut-off strategy, the plot shows the numerically found waiting time and fidelity distribution for the optimal protocol. We observe that the FIDELITY-CUT-OFF strategy yields the largest secret-key rate. However, the DIF-TIME-CUT-OFF strategy only performs slightly worse. We observed the same behaviour for all other parameter regimes we investigated.

We extend the analysis of the previous sub-section to two other cut-off strategies: a cut-off on the fidelity (FIDELITY-CUT-OFF) and on the total waiting time (MAX-TIME-CUT-OFF, see section 6.1 and Table 6.1 for definitions). To be precise, we choose the same 9-node protocol from (6.17) and use FIDELITY-CUT-OFF and

MAX-TIME-CUT-OFF as the CUT-OFF unit, respectively.

We observe that a single optimal cut-off threshold exists for both strategies, as we saw before already for the DIF-TIME-CUT-OFF strategy in Fig. 6.5. For each strategy, we optimise their cut-off parameters and plot the waiting time distribution and fidelity distribution in Fig. 6.6. As shown in the figure, although the FIDELITY-CUT-OFF yields the highest secret-key rate, the distribution and resulting secret-key rate of the DIF-TIME-CUT-OFF strategy are very close to those of the FIDELITY-CUT-OFF strategy. In contrast, the MAX-TIME-CUT-OFF strategy performs significantly worse in the achieved secret-key rate ($\approx 10\%$). We find similar behaviour also in other parameter regimes.

Since the DIF-TIME-CUT-OFF strategy is straightforward to implement in experiments while it performs only marginally worse than the best of the three strategies (FIDELITY-CUT-OFF), we focus on this strategy for further analysis.

6.4.3. PERFORMANCE OF THE OPTIMAL CUT-OFF FOR VARYING HARDWARE PARAMETERS

We proceed with optimising the cut-off in the DIF-TIME-CUT-OFF strategy to maximise the secret key rate for a range of parameters. The maximal secret-key rates for different repeater parameters are shown in Fig. 6.7(a-d). We observe that cut-offs extend the parameter regime for which secret key can be generated. To see how much one can gain in the secret key rate by using cut-offs, we choose two parameters t_{coh} and w_0 and plot the absolute increase in Fig. 6.8. We observe that the use of the optimal cut-off increases the secret key rate for the entire parameter range plotted and the improvement is largest close to the threshold parameters at which the no-cut-off protocol starts to produce nonzero secret key.

In addition, we compare uniform and non-uniform cut-offs, where 'uniform' means that we choose the same cut-off time for each nesting level. For the parameter regimes studied, we observe that non-uniform and uniform cut-off perform similarly, see Fig. 6.7(a-d).

Our next step is the sensitivity analysis of cut-off performance in the hardware parameters. For this, we first choose baseline values for the four hardware parameters and find the corresponding optimal cut-off τ_{baseline}. Given a target set of parameters that deviates slightly from the baseline values (optimal cut-off τ_{target}), we quantify the sensitivity by their relative performance difference

$$\frac{R(\tau_{\text{target}}) - R(\tau_{\text{baseline}})}{R(\tau_{\text{target}})} \tag{6.18}$$

where R is the secret-key rate achieved by the repeater protocol. If this relative difference is small, the performance of cut-off is insensitive to the parameter deviation.

In Fig. 6.7(e-h), we plot the relative performance difference for deviations in each of the four hardware parameters separately. We find that the performance of the baseline cut-off is influenced most by variation in coherence time, while it is largely insensitive to change in the swap success probability. For the coherence time and the remaining two parameters, the elementary link quality and the success probability of elementary link generation, we distinguish the case where the parameter is improved and the regime where the parameter is made worse. We observe that a worse parameter results in a

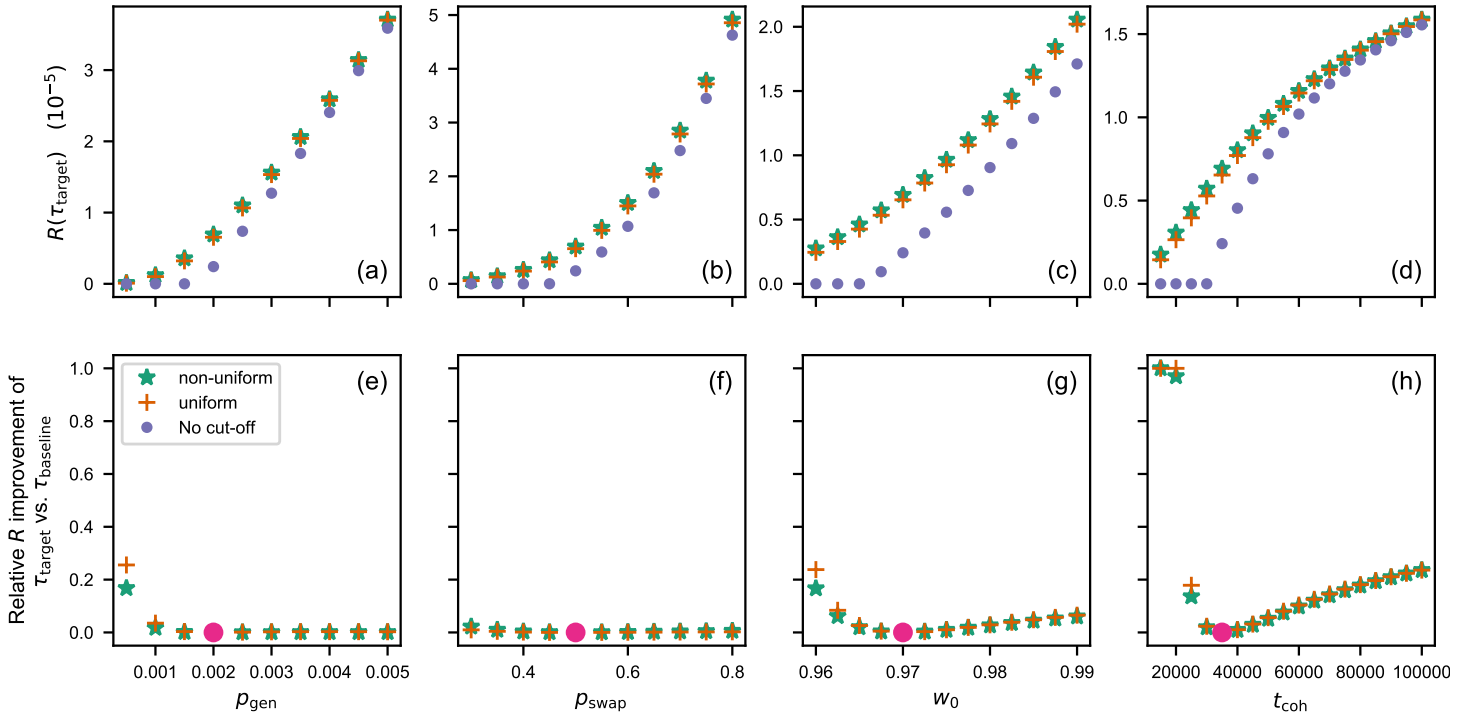


Figure 6.7: The effect of the optimal cut-off (cut-off on the difference in entanglements' production times) on secret-key rate for different hardware parameters, for the 9-node protocol as in (6.17). We choose a set of parameters as baseline parameters ($p_{\text{gen}} = 0.002$, $p_{\text{swap}} = 0.5$, $w_0 = 0.97$ and $t_{\text{coh}} = 35000$) and in each plot in the figure, we vary only one of the four parameters. The **top plots (a-d)** show the performance of the protocol with optimised cut-offs, where the optimisation is implicitly performed for each data point separately. The set of cut-offs we optimise over is either non-uniform (allow for different cut-offs at the three nesting levels of the protocols) or uniform (same cut-off at each level). We observe that the performance difference between uniform and non-uniform cut-offs is small or even negligible. The plots also indicate parameter regimes in which the protocol with the optimal cut-off generates key while its no-cut-off alternative does not (*i.e.* the no-cut-off has zero secret-key rate). The **bottom plots (e-h)** show relative performance improvement ((6.18)) of the optimal cut-off (τ_{target}) for a given data point, versus the optimal cut-off τ_{baseline} for the baseline parameters (see above). The plots show that cut-off performance is most sensitive to coherence time (t_{coh}), while it is least influenced by varying the success probability entanglement swapping (p_{swap}). For a detailed explanation see the main text. Note that the smaller the relative secret-key rate improvement (vertical axis), the closer the performance of τ_{baseline} is to the performance of the optimal τ_{target}, which is why in the plots the best-performing 'non-uniform' cut-off shows smaller relative improvement than the best-performing 'uniform' cut-off. The purple circles refer to the baseline parameters, for which the relative improvement is 0 by definition.

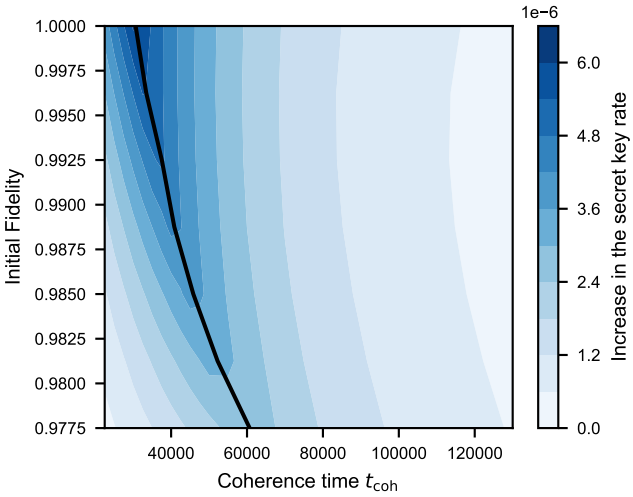


Figure 6.8: The absolute increase in secret key rate with the optimal cut-off compared to no cut-off as a function of memory coherence time and fidelity of the elementary links (= $(1+3w_0)/4$, see section 6.1), for the 9-node repeater protocols as in (6.17) where the used cut-off strategy is DIF-TIME-CUT-OFF. The black solid line separates the area where the no-cut-off protocol produces no secret key (left of the line) and where its secret-key rate is strictly larger than zero (right of the line). We observe that for the entire parameter range depicted in the figure, cut-offs increase the secret key rate and the absolute improvement is largest for parameters close to the key-producing threshold for the no-cut-off protocol (i.e. close to the black solid line). The plot consists of 126 data points on a grid and the used parameters are $p_{\text{gen}} = 0.001$ and $p_{\text{swap}} = 0.5$. Time unit is the duration of a single elementary link generation attempt.

significant performance difference with the optimal cutoff, while the performance difference is small when the parameter is improved.

We finish by investigating the most influential parameter, the coherence time, in Fig. 6.9. We observe that the optimal threshold depends approximately linearly on the memory coherence time, which could serve as a heuristic for choosing a performant cut-off.

6.5. CONCLUSION

In this chapter, we optimised the secret key rate over repeater protocols including cut-offs. Our main tool is an algorithm for computing the probability distribution of waiting time and fidelity of the first generated end-to-end link. The algorithm is applicable to a large class of quantum repeater schemes that can include cut-off strategies and distillation. Its runtime is polynomial in the support size of the probability distribution of waiting time.

Our simulations show that the use of the optimal cut-off lowers the hardware quality threshold at which secret key can be generated compared to the no-cut-off alternative. Furthermore, we observed an increase in secret-key rate for the entire regime studied for which the no-cut-off protocol produces nonzero key.

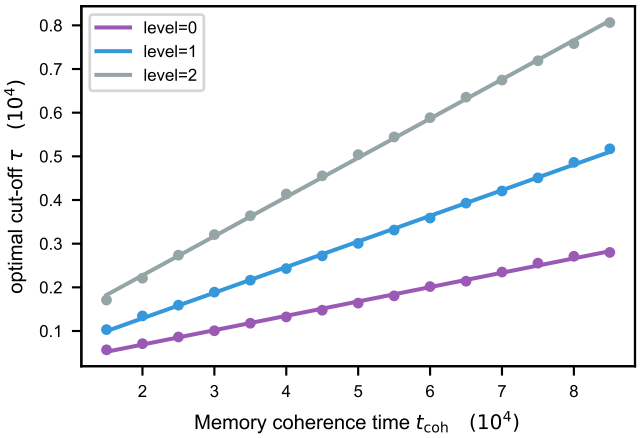


Figure 6.9: Optimal cut-off as a function of the memory coherence time in the nested 9-node repeater protocols from (6.17), where the cut-off strategy is (DIF-TIME-CUT-OFF). We observe that the numerically found optimal cut-off (dots) is a linear function of the coherence time. Solid lines are linear fits. The hardware parameters used are the same as those for Fig. 6.7 (d). When considering the same protocol on fewer nesting levels (3 and 5 nodes, respectively), we observe similar behaviour.

Regarding the choice of cut-off, we find that uniform cut-offs lead to a negligible reduction in the secret key rate compared to the optimal set of cut-offs which differ per nesting level. Moreover, the optimal uniform cut-off is highly sensitive to the quality of the memory, while it is barely influenced by the success probability of swapping. Such sensitivity could guide the heuristic cut-off optimisation of more complex protocols.

6.6. APPENDIX

6.6.1. VALIDATION AGAINST A MONTE CARLO ALGORITHM

In this section, we verify that our implementation of the deterministic algorithm presented in section 6.2 is correct by validation against the Monte Carlo sampling algorithm from Chapter 5. For all repeater schemes we ran (up to $2^{10}+1$ nodes for some parameters), we observed good agreement between the waiting time probability distribution and Werner parameter the algorithms computed, which is convincing evidence that our implementation is correct. Fig. 6.4 depicts the result of a typical run.

What follows is a brief description of the Monte Carlo algorithm from Chapter 5, including an extension to CUT-OFF. Each run of the Monte Carlo algorithm samples a tuple of waiting time and Werner parameter. It is defined recursively by having a dedicated function for each PROTOCOL-UNIT (described below) call the dedicated functions of the two PROTOCOL-UNITs that produce its two input links. The recursion follows the repeater protocol's tree structure (see Fig. 6.1), resulting in a sampling algorithm of waiting time and Werner parameter of the entire repeater protocol.

The dedicated functions for each of the four PROTOCOL-UNITs are as follows. If the protocol is only a GEN, the Monte Carlo algorithm samples the waiting time from the geometric distribution with parameter p_{gen} and the Werner parameter is the constant

w_0. For the other PROTOCOL-UNITs, each of which takes two links as input, the algorithm begins by initialising the total elapsed time $t = 0$. Then, it enters a loop which starts by calling the dedicated functions of the PROTOCOL-UNITs that produce the two input links, resulting in two samples (t_A, w_A) and (t_B, w_B). The algorithm randomly declares 'success' or 'failure' according to the success probability in Table 6.1. If it succeeds, the function breaks the loop and outputs $t+\max(t_A, t_B)$ and the resulting Werner parameter $w_{\text{out}}(t_A, w_A, t_B, w_B)$ (see Table 6.1). If it fails, the total elapsed time t is increased by the waiting time ($\max(t_A, t_B)$ for SWAP and DIST, $\min(t_A, t_B)+\tau$ for CUT-OFF) and the function goes back to the start of the loop.

6.6.2. ALTERNATIVE ALGORITHM AND ITS COMPLEXITY

In section 6.2.6, we presented an $\mathcal{O}(t_{\text{trunc}}^2 \log t_{\text{trunc}})$-algorithm for evaluating analytically-derived expressions for the waiting time distribution and average fidelity. Here, we outline how the algorithm can be modified to achieve a complexity reduction to $\mathcal{O}(t_{\text{trunc}} \log t_{\text{trunc}})$ for protocols composed of PROTOCOL-UNITs in Table 6.1 except for FIDELITY-CUT-OFF. Similar to the algorithm from the main text, the modified algorithm consists of two steps: first, evaluating the expressions regarding a single attempt (equations (6.6), (6.7) and (6.12)), followed by computing expressions regarding the whole PROTOCOL-UNIT (equations (6.11) and (6.14)). We show a complexity reduction for both.

For the first part, we show how to evaluate (6.6), (6.7) and (6.12) in time $\mathcal{O}(t_{\text{trunc}})$, improving on the $\mathcal{O}(t_{\text{trunc}}^2)$ runtime of the algorithm in the main text. Our insight here is that p and $p \cdot w_{\text{out}}$, for SWAP and DIST (see Table 6.1), can always be written in the form

$$\sum_i f^{(i)}(t_{\text{A}}) \cdot g^{(i)}(t_{\text{B}}) \tag{6.19}$$

where the $f^{(i)}$ and $g^{(i)}$ are arbitrary functions on the real numbers. For instance, given $t_{\text{A}} \geq t_{\text{B}}$, we can write the success probability of distillation p_{dist} with $f^{(1)}(t_{\text{A}}) = \frac{1}{2}$, $g^{(1)}(t_{\text{B}}) = 1$ and $f^{(2)}(t_{\text{A}}) = \frac{1}{2} p_{\text{swap}} w_{\text{A}}(t_A) \exp\left(-\frac{t_{\text{A}}}{t_{\text{coh}}}\right)$, $g^{(2)}(t_{\text{B}}) = w_{\text{B}}(t_B) \exp\left(\frac{t_{\text{B}}}{t_{\text{coh}}}\right)$. Consequently, each of (6.6), (6.7) and (6.12) can be written in the form

$$\sum_{t_{\text{A}}, t_{\text{B}}: \max(t_{\text{A}}, t_{\text{B}}) = t} \Pr(T_{\text{A}} = t_{\text{A}}, T_{\text{B}} = t_{\text{B}}) \cdot \sum_i f^{(i)}(t_{\text{A}}) g^{(i)}(t_{\text{B}}) \tag{6.20}$$

which can be rewritten by splitting up the sum in the regime $t_A \geq t_B$ and $t_B > t_A$:

$$\begin{aligned} & \sum_{t_{\text{B}}=0}^{t} \Pr(T_{\text{A}} = t, T_{\text{B}} = t_{\text{B}}) \cdot \sum_i f^{(i)}(t) g^{(i)}(t_{\text{B}}) \\ + & \sum_{t_{\text{A}}=0}^{t-1} \Pr(T_{\text{A}} = t_{\text{A}}, T_{\text{B}} = t) \cdot \sum_i f^{(i)}(t_{\text{A}}) g^{(i)}(t). \end{aligned} \tag{6.21}$$

The first term in (6.21) can be written as

$$\Pr(T_{\text{A}} = t) \cdot \sum_i f^{(i)}(t) \cdot G^{(i)}(t) \tag{6.22}$$

where we have defined

$$G^{(i)}(t) = \sum_{t_{\text{B}}=0}^{t} \Pr(T_{\text{B}} = t_{\text{B}}) g^{(i)}(t_{\text{B}}).$$

The expression for the second term in (6.21) can be found analogously. Computing (6.22) for all t is now performed by first computing $G^{(i)}(t)$ for all t, which requires linear time in t_{trunc}, and then evaluating (6.22) for fixed t in constant time. Therefore, the complexity for computing (6.22) and also for (6.20) for all t scales as $\mathcal{O}(t_{\text{trunc}})$.

This complexity holds also for protocols with DIF-TIME-CUT-OFF and MAX-TIME-CUT-OFF, as the cut-off condition appears only as an additional constraint on t_A and t_B in the sum of (6.21). For the third cut-off strategy we consider in this chapter, FIDELITY-CUT-OFF, the cut-off condition is not a function of time and therefore the above method does not work.

The second part regards the evaluation of (6.8) and (6.13) which is done exactly by the algorithm from the main text in time $\mathcal{O}(t_{\text{trunc}}^2 \log t_{\text{trunc}})$. Here, we give an $\mathcal{O}(t_{\text{trunc}} \log t_{\text{trunc}})$-algorithm which evaluates the equivalent expressions in Fourier space given in section 6.2.6 (equations (6.11) and (6.14)) with arbitrarily small error. We proceed in two steps. First, we show how to evaluate the expressions in Fourier space exactly in time $\mathcal{O}(t_{\text{trunc}}^2 \log t_{\text{trunc}}^2)$. Then, we show how to achieve a reduction to $\mathcal{O}(t_{\text{trunc}} \log t_{\text{trunc}})$ with an arbitrarily small error.

The expressions in Fourier space (equations (6.11) and (6.14)) hold for any t in case P_s, P_f and W_s are defined for all $t \geq 0$. However, in the implementation, we truncate the distribution and only have access to them for $0 \leq t < t_{\text{trunc}}$, each stored as an array of length t_{trunc}, and use the discrete Fourier transform defined in (6.10). The convolution defined in this way is a circular convolution:

$$[f_1 \tilde{*} f_2](t) = \sum_{t'=0}^{t} f_1(t-t') \cdot f_2(t') + \sum_{t'=t+1}^{L-1} f_1(L+t-t') \cdot f_2(t') \tag{6.23}$$

where L is the length of the array and $\tilde{*}$ denotes the circular convolution. The circular convolution introduces discrepancy compared to the linear convolution defined in (6.9) because $[f_1 \tilde{*} f_2](t) = [f_1 * f_2](t) + [f_1 * f_2](L+t)$. To avoid this, we pad the arrays of P_s, P_f and W_s with zeroes until a length of $L = t_{\text{trunc}}^2$, which is longer than the size of t_{trunc} times convolution of arrays of size t_{trunc} (see equivalent expressions (6.8) and (6.13), and the algorithm presented in section 6.2.6). That is, we set $P_s(t) = 0$ and $P_f(t) = 0$ for $t_{\text{trunc}} \leq t < L = t_{\text{trunc}}^2$. With this setup, the summand in the circular convolution is always 0 for $t' > t$ and it coincides with the linear one. The complexity of the obtained algorithm evaluating (6.11) and (6.14) is dominated by one Fourier transform and one inverse Fourier transform on an array of length $\mathcal{O}(t_{\text{trunc}}^2)$. Since a Fourier transform on an array of length L can be performed in time $\mathcal{O}(L \log L)$, the algorithm has a complexity of $\mathcal{O}(t_{\text{trunc}}^2 \log t_{\text{trunc}}^2)$.

We now show that we can reduce this complexity by zero-padding the arrays only until a length of $C\,t_{\text{trunc}}$ for some predefined constant C, yielding an exponentially small error

$$\epsilon = \max_t \left(\left| \Pr(T_{\text{out}} = t) - \Pr\left(T_{\text{approx}} = t\right) \right| \right)$$

in C of the distribution $\Pr\left(T_{\text{approx}} = t\right)$ obtained with circular convolution. The resulting algorithm has complexity of $\mathcal{O}(C\,t_{\text{trunc}} \log(C\,t_{\text{trunc}})) = \mathcal{O}(t_{\text{trunc}} \log t_{\text{trunc}})$.

The motivation behind this reduction is that $\Pr(T_{\text{out}} = t)$ is the sum of all possible sequences of failed attempts (see (6.8)) and is exponentially decreasing for large t. For a fixed number of attempts k, the probability results from a successful attempt after at least $k-1$ failed attempts. Therefore, it has an occurrence probability of at most $(1-p)^{k-1}$, where p is the success probability for a PROTOCOL-UNIT. To see this mathematically, we use the Young's convolution inequality [26] and obtain

$$\left\| \circledast_{j=1}^{k-1} P_{\text{f}}^{(j)} * P_{\text{s}} \right\| \leq \|P_{\text{f}}\|^{k-1} \|P_{\text{s}}\| \leq (1-p)^{k-1}$$

where the norm is defined by $\|f(t)\| = \sum_t f(t)$. In addition, note that

$$\left[\circledast_{j=1}^{k-1} P_{\text{f}}^{(j)} * P_{\text{s}} \right](t) = 0 \quad \text{for} \quad t \geq k t_{\text{trunc}}$$

because $P_{\text{f}}(t)$ and $P_{\text{s}}(t)$ are finite arrays of length t_{trunc}. Hence, for $t \geq K t_{\text{trunc}}$, we only need to consider the terms with $k \geq K+1$, *i.e.* cases with at least K failed attempts. As a result, we obtain a bound for the probability given in (6.8) for $t \geq K t_{\text{trunc}}$:

$$\Pr(T_{\text{out}} = t) \leq \sum_{k=K+1}^{\infty} (1-p)^{k-1} = \frac{(1-p)^K}{p}.$$

The above expression bounds the distribution with an exponentially decreasing probability with respect to the minimal number of failed attempts, which we now use to bound the error. Because of the circular convolution (6.23), if we only zero-pad to $C t_{\text{trunc}}$, the obtained distribution is given by

$$\Pr\big(T_{\text{approx}} = t\big) = \sum_{j=0}^{\infty} \Pr\big(T_{\text{out}} = t + jC t_{\text{trunc}}\big)$$

for $0 \leq t < C t_{\text{trunc}}$. That is, the probability for $t > C t_{\text{trunc}}$ ($j > 0$) will be added to the first $C t_{\text{trunc}}$ elements, introducing an error in the final result. This error is bounded by

$$\epsilon = \sum_{j=1}^{\infty} \frac{(1-p)^{jC}}{p} \leq \frac{(1-p)^C}{p^2},$$

which is exponentially small in C. The same bound can be given in analogue for the calculation of $W_{\text{out}}(t)$ defined in (6.13) by noticing that $W_{\text{s}}(t) \leq 1$.

The above bound is only for a single PROTOCOL-UNIT and does not account for the propagation of noise among different levels. However, in practice, as long as one chooses a C large enough so that the error on each array value is below the numerical accuracy, this improved algorithm gives the same result as the algorithm provided in the main text. In addition, the above bound is very loose. In our numerical study, we find that, if the truncation time t_{trunc} is chosen so that more than 99% distribution is covered, it suffices to triple the size of the array during the calculation, *i.e.* set $C = 3$.

Although in general there exists no efficient algorithm which captures a constant fraction of the probability mass for protocols including a cut-off (see section 6.2.6), we numerically find that the algorithm outlined above scales polynomially in the number of nodes in some parameter regimes, see Fig. 6.10.

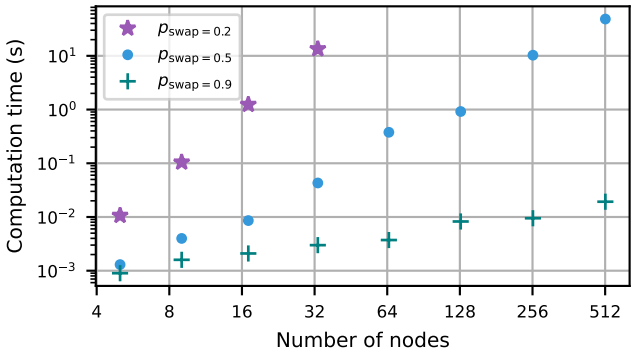


Figure 6.10: Computation time of the algorithm from appendix 6.6.2 as a function of the number of nodes in the repeater chain using consumer-market hardware (Intel i7-8700 CPU). We plot the computation time for three different p_{swap} and for protocols of the form GEN $\rightarrow$ ($\rightarrow$ CUT-OFF $\rightarrow$ SWAP)^n, similar to (6.17), where n is the nesting level and the number of nodes is $2^n + 1$. The truncation time is chosen such, that 99% of the probability mass is covered. Note that the plot's axes are both given in logarithmic scale; in such a log-log plot, a polynomial function is represented as a line. The used cut-off strategy is DIF-TIME-CUT-OFF and the other parameters used are: $p_{\text{gen}} = 0.1$, $w_0 = 1.0$, $t_{\text{coh}} = 500/p_{\text{swap}}^{n-1}$, $\tau = 42/p_{\text{swap}}^{n-1}$. In this plot, the number of truncation time steps goes up to about 10^6.

6

6.6.3. CALCULATION OF THE SECRET-KEY RATE

Here, we show how we calculate the secret-key rate with truncated waiting time distribution.

One could think of the secret-key rate, computed with finite truncation time $t_{\text{trunc}} < \infty$, as an approximation of the real secret-key rate or, alternatively, as the rate achieved by the following repeater protocol. The protocol starts with the two parties at the end nodes agree on a truncation time t_{trunc}. If up to $t = t_{\text{trunc}}$ the end-to-end link has not been delivered, the protocol terminates and restarts from GEN. Therefore, the number of protocol executions follows the geometric distribution with success probability $p_{\text{tr}} = \Pr(T \leq t_{\text{trunc}})$. The waiting time for a failed protocol is t_{trunc} while for a successful one it follows the waiting time distribution $\Pr(T = t)$ for $t < t_{\text{trunc}}$. The average total waiting time is then the sum of the time consumed in failed and successful executions:

$$\bar{T} = t_{\text{trunc}} \cdot \left(\sum_{k=1}^{\infty} k \cdot p_{\text{tr}} (1 - p_{\text{tr}})^k \right) + \frac{\sum_{t=1}^{t_{\text{trunc}}} t \cdot \Pr(T = t)}{\Pr(T \leq t_{\text{trunc}})}.$$

Accordingly, the average Werner parameter is an average over the successful execution

$$\bar{W} = \frac{\sum_{t=1}^{t_{\text{trunc}}} W(t) \cdot \Pr(T = t)}{\Pr(T \leq t_{\text{trunc}})}.$$

With the above equations, we calculate the secret-key rate defined in (6.16). In this chapter, we choose heuristically a t_{trunc} such that $\Pr(T \leq t_{\text{trunc}}) \geq 99\%$. With this choice, the difference in the secret key rate between protocols with finite and infinite t_{trunc} is negligibly small.

REFERENCES

[1] W. J. Munro, K. Azuma, K. Tamaki, and K. Nemoto, *Inside quantum repeaters,* IEEE Journal of Selected Topics in Quantum Electronics **21**, 78 (2015).

[2] S. Muralidharan, L. Li, J. Kim, N. Lütkenhaus, M. D. Lukin, and L. Jiang, *Optimal architectures for long distance quantum communication,* Scientific Reports **6**, 20463 EP (2016), article.

[3] O. A. Collins, S. D. Jenkins, A. Kuzmich, and T. A. B. Kennedy, *Multiplexed memory-insensitive quantum repeaters,* Phys. Rev. Lett. **98**, 060502 (2007).

[4] L. Praxmeyer, *Reposition time in probabilistic imperfect memories,* arXiv preprint arXiv:1309.3407 (2013), arXiv:1309.3407 .

[5] N. Kalb, A. A. Reiserer, P. C. Humphreys, J. J. W. Bakermans, S. J. Kamerling, N. H. Nickerson, S. C. Benjamin, D. J. Twitchen, M. Markham, and R. Hanson, *Entanglement distillation between solid-state quantum network nodes,* Science **356**, 928 (2017).

[6] F. Rozpędek, R. Yehia, K. Goodenough, M. Ruf, P. C. Humphreys, R. Hanson, S. Wehner, and D. Elkouss, *Near-term quantum-repeater experiments with nitrogen-vacancy centers: Overcoming the limitations of direct transmission,* Phys. Rev. A **99**, 052330 (2019).

[7] F. Rozpędek, K. Goodenough, J. Ribeiro, N. Kalb, V. C. Vivoli, A. Reiserer, R. Hanson, S. Wehner, and D. Elkouss, *Parameter regimes for a single sequential quantum repeater,* Quantum Science and Technology (2018).

[8] S. Santra, L. Jiang, and V. S. Malinovsky, *Quantum repeater architecture with hierarchically optimized memory buffer times,* Quantum Science and Technology **4**, 025010 (2019).

[9] K. Chakraborty, F. Rozpędek, A. Dahlberg, and S. Wehner, *Distributed routing in a quantum internet,* arXiv:1907.11630 (2019), arXiv:1907.11630 .

[10] P. van Loock, W. Alt, C. Becher, O. Benson, H. Boche, C. Deppe, J. Eschner, S. Höfling, D. Meschede, P. Michler, F. Schmidt, and H. Weinfurter, *Extending quantum links: Modules for fiber- and memory-based quantum repeaters,* arXiv:1912.10123 (2019), arXiv:1912.10123 .

[11] F. Schmidt and P. van Loock, *Memory-assisted long-distance phase-matching quantum key distribution,* Phys. Rev. A **102**, 042614 (2020).

[12] S. Khatri, C. T. Matyas, A. U. Siddiqui, and J. P. Dowling, *Practical figures of merit and thresholds for entanglement distribution in quantum networks,* Phys. Rev. Research **1**, 023032 (2019).

[13] E. Shchukin, F. Schmidt, and P. van Loock, *Waiting time in quantum repeaters with probabilistic entanglement swapping,* Phys. Rev. A **100**, 032322 (2019).

[14] Y. Wu, J. Liu, and C. Simon, *Near-term performance of quantum repeaters with imperfect ensemble-based quantum memories,* Phys. Rev. A **101**, 042301 (2020).

[15] R. F. Werner, *Quantum states with Einstein-Podolsky-Rosen correlations admitting a hidden-variable model,* Phys. Rev. A **40**, 4277 (1989).

[16] C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, and W. K. Wootters, *Purification of noisy entanglement and faithful teleportation via noisy channels,* Phys. Rev. Lett. **76**, 722 (1996).

[17] V. V. Kuzmin, D. V. Vasilyev, N. Sangouard, W. Dür, and C. A. Muschik, *Scalable repeater architectures for multi-party states,* npj Quantum Information **5**, 115 (2019).

[18] V. V. Kuzmin and D. V. Vasilyev, *Diagrammatic technique for simulation of large-scale quantum repeater networks with dissipating quantum memories,* Physical Review A **103**, 032618 (2021).

[19] J. W. Cooley and J. W. Tukey, *An algorithm for the machine calculation of complex fourier series,* Mathematics of Computation **19**, 297 (1965).

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

[21] P. W. Shor and J. Preskill, *Simple proof of security of the BB84 quantum key distribution protocol,* Phys. Rev. Lett. **85**, 441 (2000).

[22] H.-K. Lo, H. F. Chau, and M. Ardehali, *Efficient quantum key distribution scheme and a proof of its unconditional security,* Journal of Cryptology **18**, 133 (2005).

[23] R. Storn and K. Price, *Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces,* Journal of global optimization **11**, 341 (1997).

[24] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, *et al.*, *SciPy 1.0: fundamental algorithms for scientific computing in Python,* Nature methods , 1 (2020).

[25] git, *Optimization of cut-offs for repeater chains,* `https://github.com/BoxiLi/repeater-cut-off-optimization` (2019).

[26] V. I. Bogachev, *Measure theory*, Vol. 1 (Springer Science & Business Media, 2007).

7

IMPROVED ANALYTICAL BOUNDS ON DELIVERY TIMES OF LONG-DISTANCE ENTANGLEMENT

In this chapter, we provide improved analytical bounds on the mean and quantiles of the completion time of all `tree-shaped`*-type long-distance entanglement delivery schemes (see Chapter 3) in case the success probability of the individual components is bounded by a constant from below. A canonical example of such a protocol is the* NESTED-SWAP-ONLY *scheme which was introduced in Chapter 3: a symmetric nested quantum repeater scheme which consists of heralded entanglement generation and entanglement swaps. For this scheme specifically, our results imply that a common approximation to the mean entanglement distribution time, the 3-over-2 formula, is in essence an upper bound to the real time. Another example we treat is a quantum switch, which distributes multipartite entanglement. Our results rely on a novel connection with reliability theory.*

This chapter has been accepted, with minor changes, for publication in *Physical Review A*:
T. Coopmans, S. Brand and D. Elkouss, *Improved analytical bounds on delivery times of long-distance entanglement*. A preprint can be found on: arXiv:2103.11454 (2021).

Knowledge of the time that quantum repeater schemes take to deliver entanglement is highly relevant, for several reasons. Most evidently, the delivery rate should be sufficiently high for the application. Secure communication over video, for example, requires transmission rates of at least hundreds of kbits per second [1]. Furthermore, for the repeater proposals which make use of quantum memories and do not rely on error correcting codes, i.e. the ones that are closest to experimental reach, the delivery time influences the quality of the produced entanglement. The reason for this is that in these schemes, an entangled pair that is generated often needs to wait for another pair before the scheme can continue, and decoheres in memory while waiting. In addition, some memory types suffer from effects which are effectively time-dependent, such as noise which is induced each time the quantum processor attempts to generate remote entanglement [2], while for others the probability of extracting the state degrades over time [3]. Thus, the quality of the produced entanglement is a function of the time its generation takes. This implies that knowledge of the delivery time is crucial for assessing the viability of schemes for long-distance entanglement distribution using near-term hardware.

Analysis of the delivery time is generally challenging for the entanglement-distribution schemes that are closest to experimental reach because they consist of probabilistic components. The completion time of a such a scheme is not a single number but instead a random variable, which for many schemes has a complex structure due to the feedback loops and restarts. Although numerically, progress has recently been made in determining the completion time for increasingly larger networks (see Chapters 5 and 6, and also [4–9]), numerical approaches provide only limited intuition and moreover are demanding in computation time when performing large-scale optimisation over many network designs and hardware parameters. For this reason, analytical results are more convenient.

Unfortunately, due to the complexity of the problem, even the average completion time is known exactly only in limited cases: for quantum repeater chains consisting of at most four repeater nodes [5, 10] and a star network with a single node in the centre and an arbitrary number of leaves [11]. For larger networks, analytical results only include approximations or loose bounds on the mean entanglement delivery time [12]. The approximations are based on the assumption that the success probabilities of some of the network components are very small [13–16] or close to 1 [12, 17, 18]. Neither approximations are ideal, since some success probabilities can be boosted by techniques such as multiplexing, while others are bounded well below 1 for some setups[19]. Indeed, numerics have shown for some of the approximations that they become increasingly bad as the size of the network grows [5, 6]. Another scenario in which the completion time probability distribution is brought back to a known form includes the discarding of entanglement [20, 21]. See Chapter 4 for a review of the completion time analysis for entanglement distribution schemes.

A canonical use case which has found particularly much application is a symmetric nested repeater scheme NESTED-SWAP-ONLY[22, 23], introduced in Chapter 3, where at each nesting level two entangled pairs of qubits, spanning an equal number of nodes, are connected. Consequently, the entanglement span doubles at each nesting level. For this scheme, it was empirically known [24] that for small success probabilities of connecting the pairs, the average time to in-parallel create both required initial pairs at each nesting

level is roughly 3/2 times the average time for a single pair. This results in an approximation to the average completion time of the repeater scheme which is known as the 3-over-2 formula (we already gave a brief derivation of this approximation in Chapter 4). It has been frequently used since [13, 17, 24, 25, 25, 26, 26–39]. Analytically finding the exact factor, for an arbitrary number of nesting levels and for any value of the success probabilities, has been an open problem for more than ten years [13].

In this chapter, we provide analytical bounds on the completion time which not only improve significantly upon existing bounds, but also show *how good* some of the previous approximations are because the bounds become exact in the small probability limit. To be precise, we give analytical bounds on the mean and quantiles of the completion time random variable for entanglement-distributing protocols which are constructed of probabilistic components whose success probability can be bounded by a constant from below. This includes feedback loops in which failure of one component requires restart of other components, as long as no two components wait for the same other component to finish. Regarding the symmetric nested repeater protocol, our bounds imply that the 3-over-2 approximation is, in essence, an upper bound to the mean completion time, rigorously rendering analyses based on this approximation pessimistic. Other protocols we can treat include nested repeater chains with distillation and multipartite-entanglement generation schemes [8, 11, 40], among others.

This chapter is organised as follows. First, in Sec. 7.1 we describe the class of protocols our bounds apply to and introduce concepts from reliability theory we will use in the bounds' derivation. Sec. 7.2 contains our main results: analytical bounds on the mean completion time of such protocols and the tail of its probability distribution. Next, we obtain improved bounds with respect to existing work by applying these results to two use cases: a nested quantum repeater chain (Sec. 7.3) and a quantum switch in a star network (Sec. 7.4). We prove the main results in Sec. 7.5 and finish with a discussion in Sec. 7.6.

7.1. PRELIMINARIES

7.1.1. PROTOCOLS

In this chapter, we consider `tree-shaped`-type protocols and their generalisations for distributing multipartite entanglement. (These kind of protocols were introduced in Chapter 3 and more formally defined in sec. 7.1 of Chapter 6). We will divide the building blocks that they are composed of in two categories: GENERATE and RESTART-UNTIL-SUCCESS. We treat them individually.

First, recall from Chapter 3 that by GENERATE we refer to heralded generation of fresh entanglement. In our model, entanglement generation is performed in discrete attempts of fixed duration, each of which succeeds with a given constant probability p_{gen} [31]. The success is heralded, i.e. the nodes are aware which attempts fail and which succeed. The duration of a single attempt equals L/c, where L is the distance between the nodes and c is the speed of light in the transmission medium. We use L/c as the unit of time. As a consequence, the completion time of entanglement generation is a discrete random

variable following the geometric distribution:

$$\Pr\big(T_{\text{gen}} = t\big) = \begin{cases} p_{\text{gen}}(1-p_{\text{gen}})^{t-1} \text{ if } t \geq 1 \text{ is an integer} \\ 0 \text{ otherwise.} \end{cases} . \tag{7.1}$$

We will denote the mean of this distribution by $\mu_{\text{gen}} = 1/p_{\text{gen}}$.

We will also consider the exponential distribution, which is the continuous analogue of the geometric distribution and is defined as follows: if X follows the exponential distribution with parameter $\lambda > 0$, then

$$\Pr(X > x) = e^{-\lambda x} \tag{7.2}$$

for any real number $x \geq 0$. For small p_{gen}, the completion time of entanglement generation is sometimes approximated by an exponential random variable $T_{\text{gen}}^{\text{approx}}$ with the same mean, which is achieved by setting $\lambda = 1/\mu_{\text{gen}}$.

Next, we use the term RESTART-UNTIL-SUCCESS for an operation which takes entanglement as input, performs a probabilistic operation onto it, and demands the regeneration of the input entanglement in the case of failure. Its success probability can be a function of properties of the input entanglement, such as its quality or its delivery time, but it may also be a constant. By SWAP-UNTIL-SUCCESS and DISTILL-UNTIL-SUCCESS, we refer to instantiations of RESTART-UNTIL-SUCCESS where the probabilistic operation is entanglement swapping and entanglement distillation, respectively (see Chapter 3 for an introduction to entanglement swaps and distillation).

We model the entanglement swap success with probability $0 < p_{\text{swap}} \leq 1$, which is a constant that is independent of the states upon which the swap acts. We model fusion, the generalisation of the entanglement swap which converts more than 2 input links to a multipartite entangled state, in similar fashion. The success probability of distillation depends on the states of the two links, and is lower bounded by $\frac{1}{2}$ for the schemes considered here. We assume that the durations of the entanglement swap, fusion, and distillation operations are negligible.

7.1.2. PROBABILITY THEORY AND THE NBU PROPERTY

In this chapter, we will make extensive use of a class of probability distributions called new-better-than-used (NBU), which have been studied in the context of reliability theory and life distributions [41]. In order to mathematically define new-better-than-used, we first revisit some notions from probability theory. All random variables in this chapter that are continuous have the positive reals as domain, i.e. a continuous random variable X with $\Pr(X < 0) = 0$. The cumulative distribution function (CDF) of random variable X is $x \mapsto \Pr(X \leq x)$, and the co-CDF is $x \mapsto \Pr(X > x)$. This co-CDF is also referred to as the survival function or the *reliability*, since it states the probability that X will survive at least up to time x. The residual life distribution of X is given by the conditional probability $\Pr\big(X > x + y | X > y\big)$ and describes the time that X will survive at least up another interval x given that it has already survived time y. We now say that a real-valued random variable X is new-better-than-used (NBU) or that it has the NBU property if its residual life distribution is upper bounded by the original reliability, i.e.

$$\forall x, y \geq 0 : \qquad \Pr\big(X > x + y | X > y\big) \leq \Pr(X > x). \tag{7.3}$$

Intuitively, new-better-than-used random variables describe ageing over time. As an example, consider the lifetime of a car: the probability that an old car (one that is already y years old) will survive another x years is smaller than the probability that a brand new car will reach the age of x years.

For clarity, we separately state the definition of NBU, where we use an expression equivalent to eq. (7.3) for convenience of our proofs later on.

Definition 1. *A real-valued random variable X with $\Pr(X<0)=0$, is called new-better-than-used (NBU) if*

$$\forall x,y\geq 0: \qquad \Pr\big(X>x+y\big)\leq \Pr(X>x)\cdot\Pr\big(X>y\big).$$

It is called new-worse-than-used (NWU) if the reverse inequality holds.

We give two examples of NBU distributions.

Example 1. *A delta-peak distribution $\Pr(X=x_0)=1$ for some fixed $x_0\geq 0$ is NBU, since*

$$\Pr(X>x)\Pr\big(X>y\big)=\begin{cases}1 & \text{if } x<x_0 \text{ and } y<x_0\\ 0 & \text{otherwise}\end{cases}$$

while

$$\Pr\big(X>x+y\big)=\begin{cases}1 & \text{if } x+y<x_0\\ 0 & \text{otherwise.}\end{cases}$$

Since $x+y<x_0$ implies $x<x_0$ and $y<x_0$ for any $x,y\geq 0$, we see that $\Pr\big(X>x+y\big)\leq \Pr(X>x)\Pr\big(X>y\big)$ and thus X is NBU.

Example 2. *The exponential distribution, defined in eq. (7.2), satisfies $\Pr\big(X>x+y\big)=\Pr(X>x)\Pr\big(X>y\big)$ for all $x,y\geq 0$ and is therefore both NBU and NWU.*

Lastly, we will use the notion of stochastic dominance.

Definition 2. *Let X and Y be two random variables with common domain D, a subset of the real numbers. We say that X stochastically dominates Y and write $X\geq_{\text{st}} Y$ if*

$$\Pr(X>z)\geq\Pr(Y>z)$$

for all $z\in D$.

In particular, we will use the following lemma, which states that stochastic dominance of one random variable over the other implies an ordering of their means.

Lemma 4. *Let X and Y be two random variables with domain $[0,\infty)$. If $X\geq_{\text{st}} Y$, then $E[X]\geq E[Y]$.*

Proof. The lemma directly follows from the definition of stochastic dominance, together with the fact that the mean of X can be written as an integral over the co-CDF,

$$E[X]=\int_0^\infty \Pr(X>x)\,dx,$$

and similarly for Y. □

7.2. MAIN RESULTS

In this section, we give our main results in Prop. 3 and 4: bounds on the completion time distribution for protocols composed of elementary-link generation (GENERATE) and RESTART-UNTIL-SUCCESS operations. The proofs to the main results can be found in Sec. 7.5.

Our results bound continuous completion times, whereas the completion time of elementary-link generation is the discrete random variable T_{gen} (see Sec. 7.1). Therefore, before starting our main result we first remark that T_{gen} is stochastically dominated by a continuous NBU random variable we denote as $T_{\text{gen}}^{\text{upper}}$.

Lemma 5. *The completion time T_{gen} of elementary-link generation is stochastically dominated (Def. 2) by the continuous random variable $T_{\text{gen}}^{\text{upper}} = 1 + T_{\text{exp}}$ where T_{exp} is exponentially distributed with parameter $\frac{-1}{\log(1-p_{\text{gen}})}$. That is,*

$$\begin{aligned} \Pr\left(T_{\text{gen}} > t\right) &\leq \Pr\left(T_{\text{gen}}^{\text{upper}} > t\right) \\ &= \begin{cases} 1 & \text{if } 0 \leq t \leq 1 \\ \exp\left((t-1)/\log\left(1-p_{\text{gen}}\right)\right) & \text{if } t \geq 1 \end{cases} \end{aligned}$$

The mean of T_{gen} is upper bounded by the mean of $T_{\text{gen}}^{\text{upper}}$ which is given by

$$\mu_{\text{gen}}^{\text{upper}} = 1 - \frac{1}{\log\left(1-p_{\text{gen}}\right)} = \frac{1}{p_{\text{gen}}} + \frac{1}{2} + O(p_{\text{gen}}) \tag{7.4}$$

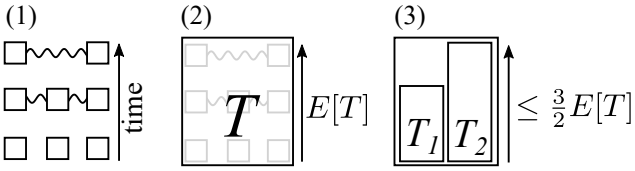
where $O(p_{\text{gen}})$ contains terms that scale with p_{gen} or powers of it. The means of T_{gen} and $T_{\text{gen}}^{\text{upper}}$ differ only slightly, both in difference and in ratio:

$$0 \leq \mu_{\text{gen}}^{\text{upper}} - \mu_{\text{gen}} \leq \frac{1}{2} \text{ and } 1 \leq \frac{\mu_{\text{gen}}^{\text{upper}}}{\mu_{\text{gen}}} \leq 1 + \frac{p_{\text{gen}}}{2} \tag{7.5}$$

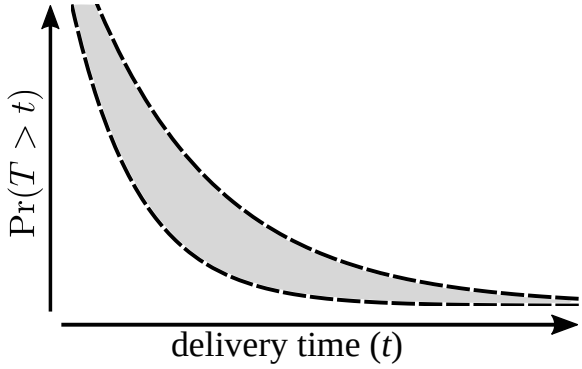
for any $p_{\text{gen}} \in [0,1]$. Moreover, $T_{\text{gen}}^{\text{upper}}$ is NBU.

As consequence of Lemma 5, we may assume that the duration of elementary-link generation is described by $T_{\text{gen}}^{\text{upper}}$ if we are looking for upper bounds on a protocol's completion time. Indeed, an upper bound on the co-CDF or the mean of the resulting completion time will automatically also become an upper bound on the real completion time (see Def. 2 and Lemma 4).

Now let us state our bounds on continuous completion times. For legibility, we first state a special case of our main result: the scenario where a SWAP-UNTIL-SUCCESS operation with constant success probability is performed on two quantum states. We assume that the time it takes until a state is produced is a random variable, and that this random variable is the same for both input states; that is, their completion times are independent and identically distributed.



(a) Consider an entanglement distribution process (1), whose completion time is a random variable T and has mean $E[T]$ (2). If T is NBU, completing two such independent and identically distributed processes in parallel has a mean time $\leq \frac{3}{2} \cdot E[T]$ (3).



(b) The probability distribution of the delivery time of entanglement distribution processes can be bounded by exponentially-fast decaying lower and upper bounds.

Figure 7.1: Visual overview of this chapter's bounds on the completion time of entanglement distribution protocols. The first result (7.1a) is a bound on the mean completion time of two parallel entanglement distribution processes, given that these processes possess the NBU property (Def. 1). Our second result (7.1b) is a two-sided bound on the probability distribution of the completion time of such processes.

Completion time of swapping: two states & IID

Proposition 3. *Consider the time T_{output} of a* SWAP-UNTIL-SUCCESS *protocol with constant success probability p, acting on two quantum states, produced with identically-distributed independent completion times T_{input}. If T_{input} is a continuous random variable and it is NBU (Def. 1), then:*

(a) T_{output} *is NBU;*

(b) the mean of T_{output} *is upper bounded as*

$$E[T_{\text{output}}] \leq \frac{3E[T_{\text{input}}]}{2p};$$

(c) for all t, the probability that T_{output} *takes longer than t timesteps decays exponentially fast:*

$$\Pr\big(T_{\text{output}} > t\big) \leq \exp\left(p - \frac{2pt}{3E[T_{\text{input}}]}\right)$$

while it is lower bounded as

$$\Pr\big(T_{\text{output}} > t\big) \geq \exp\left(\frac{-2pt}{3E[T_{\text{input}}]} \cdot \frac{1}{1-p}\right).$$

(d) in the limit $p \to 0$, the normalised completion time $T_{\text{output}}/E\left[T_{\text{output}}\right]$ approaches the exponential distribution with mean 1, and thus $E[T_{\text{output}}] \cdot 2p/(3E[T_{\text{input}}]) \to 1$.

7

The bounds from Prop. 3 are visually depicted in Fig. 7.1.

Although Prop. 3 regards a SWAP-UNTIL-SUCCESS protocol, it also finds application to DISTILL-UNTIL-SUCCESS, which has nonconstant success probability:

Remark 1. *Consider Prop. 3 where* SWAP-UNTIL-SUCCESS *is replaced by* DISTILL-UNTIL-SUCCESS. *Note:*

(a) Prop 3(a)-(c) still hold in case the quantum states produced with completion times T_{input} do not decohere over time, because then the distillation success probability p is a constant, independent of the production times of the input states;

The success probability of distillation is general lower bounded by $1/2$, resulting in

(b) $E[T_{\text{output}}] \leq 3E[T_{\text{input}}]$.

Since the upper bound in Prop 3(c) is monotonically decreasing in p in the regime $t \geq 3E[T_{\text{input}}]/2$, we may replace p by its lower bound $1/2$ to obtain:

(c) for $t \geq 3E[T_{\text{input}}]/2$, we have

$$\Pr\big(T_{\text{output}} > t\big) \leq \exp\left(\frac{1}{2} - \frac{t}{3E[T_{\text{input}}]}\right).$$

Prop. 3 is a special case of a more general version of Prop. 4 for RESTART-UNTIL-SUCCESS protocols that act on two or more quantum states whose completion times are independent, but not necessarily identically distributed.

General case: completion time of RESTART-UNTIL-SUCCESS protocol

Proposition 4. *Consider the time T_{output} of a* RESTART-UNTIL-SUCCESS *protocol with constant success probability p, acting on $n \geq 2$ quantum states, produced with independent completion times $T_1, \ldots, T_n$, which need not be identically distributed. Suppose that each of T_{output} and $T_1, \ldots, T_n$ is a continuous random variable. Denote $m = E[\max(T_1, \ldots, T_n)]$. If all $T_1, \ldots, T_n$ are NBU (Def. 1), then:*

(a) T_{output} is NBU;

(b) the mean of T_{output} equals $E[T_{\text{output}}] = m/p$;

(c) for all t, the probability that T_{output} takes longer than t timesteps is exponentially bounded from above as

$$\Pr\big(T_{\text{output}} > t\big) \leq \exp\left(p - \frac{p \cdot t}{m}\right).$$

while it is bounded from below by

$$\Pr\big(T_{\text{output}} > t\big) \geq \exp\left(\frac{-p \cdot t}{m} \cdot \frac{1}{1-p}\right).$$

(d) in the limit $p \to 0$, the normalised completion time $T_{\text{output}}/E\left[T_{\text{output}}\right]$ approaches the exponential distribution with mean 1, and thus $E[T_{\text{output}}] \cdot p/m \to 1$.

(e) We have

$$\max_{1 \leq j \leq n} E[T_j] \leq m \leq \sum_{j=1}^{n} E[T_j].$$

(f) In case all T_j are identically distributed with mean $E[T]$, then a tighter bound than (e) exists:

$$1 \leq \frac{m}{E[T]} \leq n - 1 + \frac{1}{n}.$$

We finish this section by generalising Remark 1.

Remark 2. *Consider a* RESTART-UNTIL-SUCCESS *protocol whose success probability is lower bounded by a constant c. Then the upper bounds in Prop. 4(e) and (f) still hold, while Prop. 4(b) and (c) can respectively be replaced by $E[T_{\text{output}}] \leq m/c$ and $\Pr\big(T_{\text{output}} > t\big) \leq \exp\big(c - \frac{ct}{m}\big)$ for $t \geq m$.*

In the next sections, we give two use cases for the bounds derived in this section: a quantum repeater chain scheme and a quantum switch protocol.

7.3. FIRST APPLICATION: THE NESTED-SWAP-ONLY QUANTUM REPEATER CHAIN

In this section, we apply our bounds on the completion time of entanglement distribution protocols to the extensively-studied NESTED-SWAP-ONLY protocol, a nested repeater chain protocol [22, 23] which was introduced in Chapter 3. For completeness, we briefly explain the protocol for the case where the number of segments is 2^n for some integer $n \geq 0$ (i.e. the chain consists of $2^n + 1$ nodes). See also Fig. 7.2. If $n = 0$, then the network consists of two end nodes only (no repeaters), which use heralded entanglement generation (see Sec. 7.1) to generate a single elementary link. If $n > 0$, then the chain has a middle node (since the number of segments is even). In parallel, a 2^{n-1}-hop-spanning link is produced on the left side of the middle node, as well as a link on its right side. As soon as both links have been prepared, the middle node performs an entanglement swap to convert the two links into a single 2^n-hop-spanning link. This scheme can also be extended with one or multiple rounds of entanglement distillation at each nesting level, in a nested fashion [22].

The exact completion time distribution of the nested repeater scheme has so far not been analytically found beyond the single-repeater case. The problem was first fully explained by Sangouard et al. [13], although it was already partially described in earlier work [24–26]. Sangouard et al., remarked that while the completion time of elementary-link generation at the bottom level follows a well-known distribution (the geometric distribution, Sec. 7.1), this is no longer the case for higher levels.

To circumvent this issue, many have resorted to approximating the probability distribution at each level with an exponential distribution, combined with the small-probability assumptions $p_{\text{swap}} \ll 1$ and $p_{\text{gen}} \ll 1$. We recall from Chapter 4 that this approximation leads to an expression for the mean entanglement delivery time as follows. At each nesting level, the protocol can only continue if both input states to the entanglement swap have been produced. Mathematically, this is expressed as the maximum of the delivery time of the two links. The mean of the maximum of two independent and identically distributed (i.i.d.) exponential random variables with mean μ is $\frac{3}{2} \cdot \mu$. Next, if the swap success probability is p_{swap}, then on average $1/p_{\text{swap}}$ attempts are needed until success. Thus, for each nesting level, the mean entanglement delivery time should be multiplied by a factor $3/(2p_{\text{swap}})$, resulting into an expression for the mean delivery time known as the *3-over-2-approximation*:

$$\left(\frac{3}{2p_{\text{swap}}}\right)^n \cdot \frac{1}{p_{\text{gen}}}. \tag{7.6}$$

The 3-over-2 approximation was first used by Jiang et al.[24], who mentioned that the factor 3/2 agreed well with simulations in the small-probability regime. Since then, the approximation has been frequently used [13, 17, 25–39].

However, the quality of this approximation is not known exactly and has only been very loosely bounded, as follows. As noted by Sangouard et al. [13], the mean of the maximum of two nonnegative i.i.d random variables with mean μ is lower bounded by μ and upper bounded by 2μ. These bounds correspond to the scenario where one waits only for a single link to be ready, or for both links to be prepared sequentially, respectively.

Consequently,

$$\left(\frac{1}{p_{\text{swap}}}\right)^n \cdot \frac{1}{p_{\text{gen}}} \leq E[T] \leq \left(\frac{2}{p_{\text{swap}}}\right)^n \cdot \frac{1}{p_{\text{gen}}}. \tag{7.7}$$

Now we use Markov's inequality, $\Pr(T \geq t) \leq E[T]/t$, which can be rephrased

$$\Pr(T > t) \leq E[T] \cdot \frac{1}{t+1}, \tag{7.8}$$

since T only takes integral values. Substituting $E[T]$ by its upper bound from eq. (7.7) leads to

$$\Pr(T > t) \leq \left(\frac{2}{p_{\text{swap}}}\right)^n \cdot \frac{1}{p_{\text{gen}}} \cdot \frac{1}{t+1}. \tag{7.9}$$

Both the mean bound from eq. (7.7) and the tail bound from eq. (7.9) are quite loose bounds, see Fig. 7.3 and 7.4. Only recently, it was shown analytically by Kuzmin and Vasilyev that the factor 3/2 from eq. (7.6) is exact in the limit of vanishing swap success probability, and moreover that the delivery time probability distribution after an entanglement swap in this limit is indeed an exponential distribution [14].

Our bounds from Sec. 7.2 allow us to go beyond these results. In particular, we show the following. First, we analytically show that the 3-over-2 approximation is, in essence, an *upper bound* to the mean completion time. This implies that the 3-over-2 approximation is pessimistic, confirming numerical simulations [5, 17]. Next, we derive two-sided bounds on the tail of the probability distribution of the repeater chain's completion time. Both the mean bound and the tail bounds coincide in the limit of vanishing success probabilities. We give the bounds below and plot them in Fig. 7.3 (mean bounds) and Fig. 7.4 (tail bounds).

Proposition 5. *Consider the completion time T_n of an equally-spaced, symmetric nested repeater scheme (no distillation) on 2^n segments, such as the example in Fig. 7.2 for $n = 2$. If $n > 0$, then:*

(a) the mean completion time is upper bounded as

$$E[T_n] \quad \leq \quad \left(\frac{3}{2p_{\text{swap}}}\right)^n \cdot \mu_0.$$

Here, μ_0 is the mean of any real-valued NBU random variable which stochastically dominates the completion time T_{gen} of elementary-link generation. In case the elementary-link generation is modelled as discrete attempts which succeed with probability p_{gen}, then we choose $T_{\text{gen}}^{\text{upper}}$ for this random variable (see Lemma 5), resulting in

$$\mu_0 = E[T_{\text{gen}}^{\text{upper}}] = 1 - \frac{1}{\log(1-p_{\text{gen}})}.$$

If instead the completion time of elementary-link generation is described by the exponentially-distributed random variable $T_{\text{gen}}^{\text{approx}}$ (see Sec. 7.1.1), which is NBU itself, then $\mu_0 = E[T_{\text{gen}}^{\text{approx}}] = 1/p_{\text{gen}}$. By Lemma 5, the two models' means only differ slightly: $0 \leq E[T_{\text{gen}}^{\text{upper}}] - E[T_{\text{gen}}^{\text{approx}}] \leq \frac{1}{2}$ and $1 \leq E[T_{\text{gen}}^{\text{upper}}]/E[T_{\text{gen}}^{\text{approx}}] \leq 1 + p_{\text{gen}}/2$.

(b) the mean completion time is lower bounded as

$$E[T_n] \geq \frac{1}{p_{\text{swap}}} \cdot \left(\frac{3 - 2p_{\text{swap}}}{p_{\text{swap}}(2 - p_{\text{swap}})} \right)^{n-1} \cdot \nu_0.$$

Here, ν_0 is the mean time until the latest of two parallel elementary-link generation processes has finished. In case elementary-link generation is modelled as discrete attempts which succeed with probability p_{gen}, then

$$\nu_0 = \frac{3 - 2p_{\text{gen}}}{p_{\text{gen}}(2 - p_{\text{gen}})}$$

while if its completion time is modelled by an exponential distribution, then $\nu_0 = 3/(2p_{\text{gen}})$.

(c) the co-CDF of T_n differs from the co-CDF of an exponential distribution by at most a factor $\exp(p_{\text{swap}})$ from above,

$$\Pr(T_n > t) \leq \exp(p_{\text{swap}}) \cdot \exp\left(-\frac{p_{\text{swap}} \cdot t}{m_{\text{upper}}} \right)$$

while it is lower bounded as

$$\Pr(T_n > t) \geq \exp\left(\frac{-p_{\text{swap}} \cdot t}{m_{\text{lower}}} \cdot \frac{1}{1 - p_{\text{swap}}} \right).$$

Here, we have denoted

$$m_{\text{upper}} = \frac{3}{2} \cdot \left(\frac{3}{2p_{\text{swap}}} \right)^{n-1} \cdot \mu_0$$

and

$$m_{\text{lower}} = \left(\frac{3 - 2p_{\text{swap}}}{p_{\text{swap}}(2 - p_{\text{swap}})} \right)^{n-1} \cdot \nu_0$$

where μ_0 and ν_0 are given in Prop. 5(a) and (b).

(d) in the limit where both $p_{\text{swap}} \to 0$ and $p_{\text{gen}} \to 0$, the normalised random variable $T_n / E[T_n]$ follows the exponential distribution with mean 1, and moreover

$$\lim_{p_{\text{swap}} \to 0, p_{\text{gen}} \to 0} E[T_n]/L_n = 1$$

with

$$L_n = \left(\frac{3}{2p_{\text{swap}}} \right)^n \cdot \frac{1}{p_{\text{gen}}}.$$

(e) If the completion time of elementary-link generation is described by the exponentially-distributed $T_{\text{gen}}^{\text{approx}}$, then T_n is NBU, while if it is modelled as discrete attempts, then T_n is stochastically dominated (Def. 2) by an NBU random variable which satisfies the bounds in items (a-c).

Most statements in Prop. 5 directly follow by applying Prop. 3 in Sec. 7.2 iteratively over the number of nesting levels. In particular, a useful feature following from Prop. 3(a) is that at each nesting level, the completion time possesses the NBU property (Def. 1). Consequently, the mean upper bound in Prop. 3(c), which is only applicable to NBU random variables, can be used at each nesting level. Only the lower bound in (b) and the expression for m_{lower} in (c) do not follow from Prop. 3. These can be found by noting that the maximum of two sums dominates a single sum whose length is the maximum of the original two sum lengths. We give the full proof in Appendix 7.7.2.

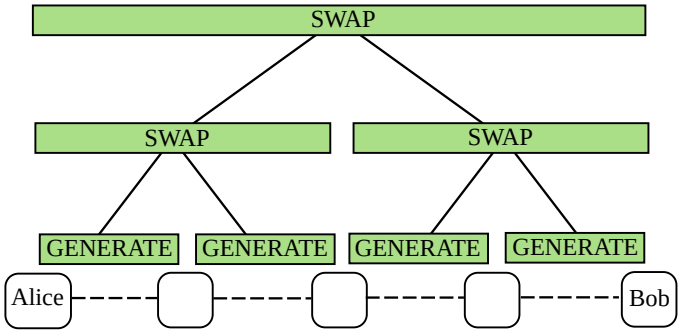


Figure 7.2: Schematic of the NESTED-SWAP-ONLY quantum repeater protocol on five nodes (figure is identical to fig. 3.3(d) from Chapter 3; we added it here so that the chapter is self-explanatory) . The figure depicts the protocol for delivering entanglement between remote parties Alice and Bob through three repeater nodes. At the start of the protocol, all nodes attempt to generate an elementary link with each of their neighbours in parallel. An entanglement swap is performed once the two leftmost links are ready, and similarly for the two rightmost links. Once both swaps have succeeded (failure requires regeneration of the involved links), the middle node performs an entanglement swap, which yields entanglement between Alice and Bob.

We finish this section by noting a stronger two-sided bound on the completion time T of an equally-spaced repeater chain than Prop. 5(a-b) in the case of deterministic swapping ($p_{\text{swap}} = 1$). The number of segments can be any integer $N \geq 2$. Since we assume that the entanglement swaps take no time (Sec. 7.1.1), the mean completion time for this scenario is

$$E[T] = E[\max(T_{\text{gen}}^{(1)}, T_{\text{gen}}^{(2)}, \ldots, T_{\text{gen}}^{(N)})]$$

where $T_{\text{gen}}^{(k)}$ is an independent and identically distributed copy of T_{gen} and describes the completion time of entanglement generation over the k^{th} segment. By replacing $T_{\text{gen}} \to T_{\text{gen}}^{\text{approx}}$, i.e. assuming that the completion time of entanglement generation follows the exponential distribution with mean $1/p_{\text{gen}}$, the following approximation to $E[T]$ has been derived [5, 15]:

$$E[T] \approx \frac{1}{p_{\text{gen}}} \cdot H_N \tag{7.10}$$

where

$$H_N := \sum_{k=1}^{N} \frac{1}{k} = \gamma + \log(N) + O\left(\frac{1}{N}\right) \tag{7.11}$$

is the N-th harmonic number and $\gamma \approx 0.5772$ is the Euler-Mascheroni constant. An alternative to eq. (7.10) is to replace $T_{\text{gen}} \to T_{\text{exp}}$, where T_{exp} is the exponentially-distributed

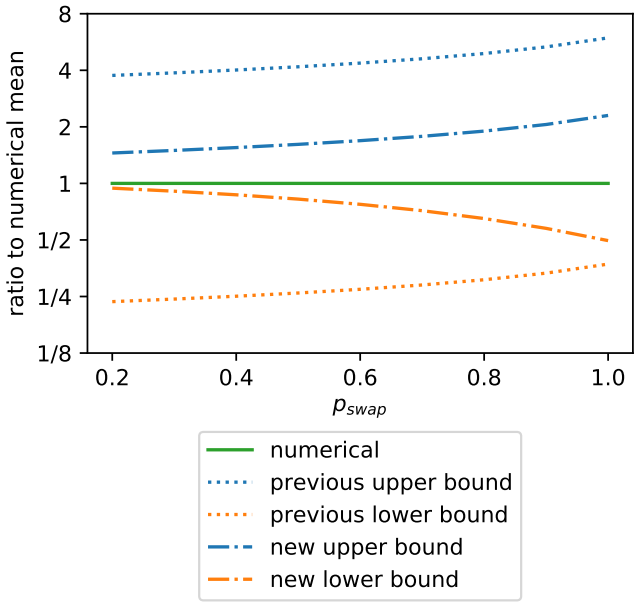


Figure 7.3: The ratio of different upper and lower bounds on the mean completion time of a nested repeater protocol, as compared to the numerically calculated mean with the deterministic algorithm from [6], for a repeater chain with 17 nodes ($p_{\text{gen}} = 0.5$, entanglement generation is performed in discrete attempts). The figure shows bounds known before this work (eq. (7.7)) and the tighter bounds from this work in Prop. 5(a) and (b).

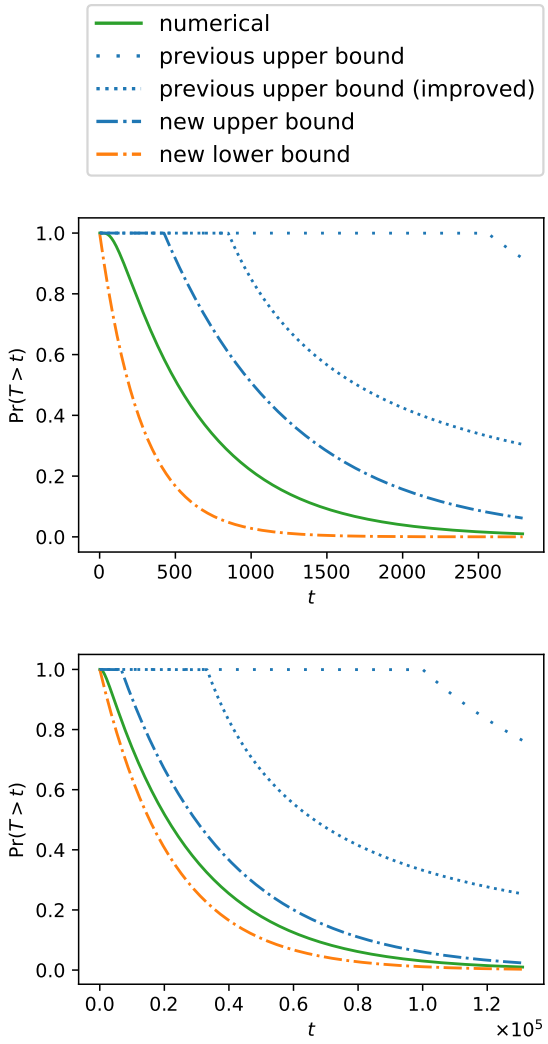


Figure 7.4: Probability distribution of the completion time T of a nested repeater protocol. The figures show the numerically computed distributions using the deterministic algorithm from [6], a polynomially-decaying bound known before this work which is derived from Markov's inequality and a bound on the mean completion time (eq. (7.9)), and two improvements on eq. (7.9) we achieve in this chapter: first, a simple improvement by using Markov's inequality and the improved bound on the mean completion time (Prop. 6(a)), followed by the exponentially-decaying two-sided tail bounds from Prop. 5(c). The plots show results for a repeater chain with 17 nodes ($p_{\text{gen}} = 0.1$) where entanglement generation is performed in discrete attempts. The swap success probability is $p_{\text{swap}} = 0.5$ (top), and $p_{\text{swap}} = 0.2$ (bottom).

random variable from Lemma 5, which results into

$$E[T] \approx \frac{-1}{\log(1-p_{\text{gen}})} \cdot H_N = \left(\frac{1}{p_{\text{gen}}} - \frac{1}{2} + O(p_{\text{gen}})\right) \cdot H_N. \tag{7.12}$$

We remark that eq. (7.10) and eq. (7.12) only differ slightly and that their ratio goes to 1 in the limit of $p_{\text{gen}} \to 0$. The quality of the second approximation, eq. (7.12), has been bounded in work by Eisenberg [42] and to our knowledge no-one has so far noted it in the context of completion times of quantum network protocols. We state it below.

Proposition 6. *[42] Suppose that entanglement swapping is deterministic ($p_{\text{swap}} = 1$). Let $E[T]$ denote the mean completion time of a repeater chain over N segments. Then $E[T]$ is bounded as*

$$a \cdot H_N \leq E[T] \leq 1 + a \cdot H_N$$

where H_N is the N-th harmonic number given in eq. (7.11) and

$$a = \mu_{\text{gen}}^{\text{upper}} - 1 = \frac{-1}{\log(1-p_{\text{gen}})} = \frac{1}{p_{\text{gen}}} - \frac{1}{2} + O(p_{\text{gen}}).$$

7.4. SECOND APPLICATION: A QUANTUM SWITCH

Here, we apply our results to a quantum switch. A quantum switch serves k user nodes. Each user is connected to the switch by an arm, which produces bipartite entanglement (a link) between switch and user. As soon as each user has produced a link with the switch, the switch performs a k-fuse operation, i.e. a probabilistic operation converting k bipartite links into a single k-partite entangled state on the user nodes.

Vardoyan et al., considered the scenario in which each user produces entanglement continuously with the switch and the switch fuses whenever it can [11]. They obtained analytical expressions for the rate at which the switch produces multipartite entanglement in the steady-state regime. Here, we consider the alternative protocol where the goal is to produce only a single k-partite state. We go beyond the model of Vardoyan et al., by replacing the arms, which connect the switch to the user, by an arbitrary entanglement-distribution network whose completion time is NBU. An example choice for such a network is the symmetric repeater chain from Sec.7.3, yielding the network topology as depicted in Fig. 7.5, Our tools allow us to achieve bounds on the completion time of the switch, as described in the following proposition.

Proposition 7. *Consider a k-armed quantum switch with fusion success probability p_{fuse}. Suppose that the completion times of the different arms are independent and identically distributed according to an NBU random variable S. Denote by T the time until the switch performs the first successful k-fuse attempt. Then:*

(a) T is NBU;

(b) The mean of T is bounded as

$$E[T] \leq \left(k - 1 + \frac{1}{k}\right) \cdot \frac{E[S]}{p_{\text{fuse}}}.$$

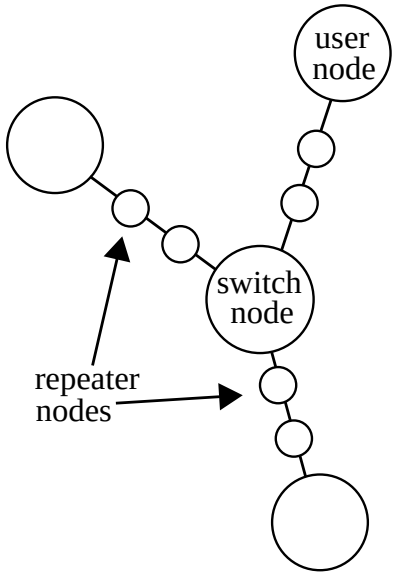


Figure 7.5: A quantum switch with 3 users, each connected to the switch by an identical repeater chain which produces links between user and switch. The switch produces 3-partite entangled states, shared between the users, by performing a probabilistic operation on 3 links, one with each user node, as soon as these 3 links are available.

(c) T's tail decays exponentially fast:

$$\Pr(T > t) \leq \exp\left(p_{\text{fuse}} - \frac{p_{\text{fuse}} \cdot t}{(k-1+1/k) \cdot E[S]}\right).$$

Prop. 7(a) follows directly from Prop. 4(a) (Sec. 7.2). Prop. 7(b) is a consequence of the expression for the mean completion time in Prop. 4(b) and the upper bound in Prop. 4(f), while Prop. 7(c) is an instantiation of the tail bound of Prop. 4(c) combined with the mean upper bound of Prop. 7(b).

7.5. PROOFS OF MAIN RESULTS

In this section, we prove our main results from Sec. 7.2. We provide proofs in the following order. First, a proof of Lemma 5. Then, we will prove Prop. 4. Since Prop. 3 is a special case of Prop. 4, we do not prove it separately.

7.5.1. PROOF OF LEMMA 5

Here, we prove the four parts of Lemma 5: (i) that T_{gen}, the completion time of heralded entanglement generation with probability p_{gen}, is stochastically dominated by $T_{\text{gen}}^{\text{upper}} = 1 + T_{\text{exp}}$, where T_{exp} is exponentially distributed with parameter $-1/\log(1-p_{\text{gen}})$. Next, (ii) that the mean of $T_{\text{gen}}^{\text{upper}}$ equals

$$1 - \frac{1}{\log(1-p_{\text{gen}})} = \frac{1}{p_{\text{gen}}} + \frac{1}{2} + O(p_{\text{gen}}).$$

Then, (iii) that $0 \leq E[T_{\text{gen}}^{\text{upper}}] - E[T_{\text{gen}}] \leq \frac{1}{2}$ and (iv) that $0 \leq E[T_{\text{gen}}^{\text{upper}}]/E[T_{\text{gen}}] \leq 1 + p_{\text{gen}}/2$. Fifth, (v) that $T_{\text{gen}}^{\text{upper}}$ is NBU.

Regarding (i), we use the definition of the geometric distribution in eq. (7.1), from which it follows that the survival function of T_{gen} is given by

$$\Pr\big(T_{\text{gen}} > t\big) = (1 - p_{\text{gen}})^{\lfloor t \rfloor}$$

for all $t \geq 1$, where $\lfloor t \rfloor$ denotes the floor of t: $\lfloor t \rfloor = t$ if t is an integer and it equals the largest integer strictly smaller than t otherwise. For $0 \leq t < 1$, we have $\Pr\big(T_{\text{gen}} > t\big) = 1 = \Pr\big(T_{\text{gen}}^{\text{upper}} > t\big)$, so the definition of stochastic dominance (Def. 2) is trivially satisfied on the interval $t \in [0, 1)$. We therefore only need to consider $t \geq 1$. Using the notation from Lemma 5, we now bound

$$\begin{aligned}
\Pr\big(T_{\text{gen}} > t\big) &= (1 - p_{\text{gen}})^{\lfloor t \rfloor} \\
&\leq (1 - p_{\text{gen}})^{t-1} \\
&= \exp\big[(t-1) \cdot \log\big(1 - p_{\text{gen}}\big)\big] \\
&\overset{*}{=} \Pr\big(T_{\text{exp}} > t - 1\big) \\
&= \Pr\big(1 + T_{\text{exp}} > t\big),
\end{aligned}$$

where in *, we have used the definition of the exponential distribution from eq. (7.2). For proving (ii), we recall that the mean of an exponential distribution with co-CDF $e^{-\lambda t}$ with parameter $\lambda > 0$ is $1/\lambda$, hence the mean of $T_{\text{gen}}^{\text{upper}}$ is

$$\begin{aligned}
E[T_{\text{gen}}^{\text{upper}}] &= E[1 + T_{\text{exp}}] \\
&= 1 + E[T_{\text{exp}}] \\
&= 1 - \frac{1}{\log\big(1 - p_{\text{gen}}\big)} \\
&= \frac{1}{p_{\text{gen}}} + \frac{1}{2} + O(p_{\text{gen}})
\end{aligned}$$

where in the last equation, we used the expansion of $1/\log(1+x)$ for $|x| < 1$ by Kowalenko [43]. We show (iii) by computing the derivative of $E[T_{\text{gen}}^{\text{upper}}] - E[T_{\text{gen}}]$ as function of p_{gen}, which equals

$$\frac{-1}{(1 - p_{\text{gen}}) \log^2(1 - p_{\text{gen}})} + \frac{1}{p_{\text{gen}}^2}. \tag{7.13}$$

It is not hard to see that eq. (7.13) is upper bounded by 0 for all $p_{\text{gen}} \in (0, 1)$: we start with the well-established inequality[44]

$$\log(x) \geq \frac{x - 1}{\sqrt{x}}$$

for $0 < x \leq 1$, which after the substitution $x \to 1 - p_{\text{gen}}$ becomes

$$\log\big(1 - p_{\text{gen}}\big) \geq \frac{-p_{\text{gen}}}{\sqrt{1 - p_{\text{gen}}}}. \tag{7.14}$$

Since both sides of eq. (7.14) are negative and the squaring function $x \mapsto x^2$ is monotonically decreasing for $x \leq 0$, squaring both sides requires the inequality sign to flip,

$$\log^2(1-p_{\text{gen}}) \leq \frac{p_{\text{gen}}^2}{1-p_{\text{gen}}}$$

and hence $(1-p_{\text{gen}})\log^2(1-p_{\text{gen}}) \leq p_{\text{gen}}^2$, implying that the derivative in eq. (7.13) is upper bounded by 0 for all $p_{\text{gen}} \in (0,1)$. Therefore, $E[T_{\text{gen}}^{\text{upper}}] - E[T_{\text{gen}}]$ is monotonically decreasing in that regime and achieves its optima at $p_{\text{gen}} \downarrow 0$ and $p_{\text{gen}} \uparrow 1$, which are $\frac{1}{2}$ and 0, respectively, yielding precisely the bound in (iii). For showing (iv), divide each side of $0 \leq E[T_{\text{gen}}^{\text{upper}}] - E[T_{\text{gen}}] \leq \frac{1}{2}$ by $E[T_{\text{gen}}]$ to obtain

$$0 \leq \frac{E[T_{\text{gen}}^{\text{upper}}]}{E[T_{\text{gen}}]} - 1 \leq \frac{1}{2E[T_{\text{gen}}]} = \frac{p_{\text{gen}}}{2}$$

from which (iv) directly follows. For proving (v), that $T_{\text{gen}}^{\text{upper}} = 1 + T_{\text{exp}}$ is an NBU random variable, we consider two cases with respect to the definition of NBU (Def. 1):

- both $x < 1$ and $y < 1$. Then

  $$\Pr\big(1 + T_{\text{exp}} > x\big) = \Pr\big(1 + T_{\text{exp}} > y\big) = 1$$

  so the definition of NBU trivially holds by the fact that $\Pr\big(1 + T_{\text{exp}} > x + y\big)$ cannot exceed 1;

- at least one of x or y is 1 or larger. Assume without loss of generality that $y \geq 1$. Then note that $\Pr\big(1 + T_{\text{exp}} > x + y\big)$ equals

  $$\begin{aligned} & \Pr\big(T_{\text{exp}} > x + (y-1)\big) \\ \leq\ & \Pr\big(T_{\text{exp}} > x\big)\Pr\big(T_{\text{exp}} > y - 1\big) \\ =\ & \Pr\big(T_{\text{exp}} > x\big)\Pr\big(1 + T_{\text{exp}} > y\big) \end{aligned}$$

  where the inequality holds by the fact that T_{exp} is itself NBU (see Example 2). The proof finishes by noting that $1 + T_{\text{exp}}$ stochastically dominates T_{exp}, i.e. $\Pr\big(1 + T_{\text{exp}} > y\big) \geq \Pr\big(T_{\text{exp}} > y\big)$.

7.5.2. PROOF OF PROPOSITION 4

Now, we prove Prop. 4, which automatically proves its special case Prop. 3. For our proof, we first give a formal definition of T_{output}, following Brand et al. [6]. The RESTART-UNTIL-SUCCESS acts on n quantum states, which first need to have been delivered. Thus, we define a fresh random variable to refer to the time until the last of n quantum states has been delivered:

$$M := \max(T_1, \ldots, T_n).$$

The restarts of the RESTART-UNTIL-SUCCESS protocol, according to a constant success probability p, result in the fact that T_{output} can be written as a *geometric sum* of copies of M:

$$T_{\text{output}} = \sum_{k=1}^{K} M^{(k)} \tag{7.15}$$

where $M^{(k)}$ is an i.i.d. copy of M and K is a geometrically distributed random variable with parameter p:

$$\Pr(K = k) = p(1-p)^{k-1}. \tag{7.16}$$

Eq. (7.15) reflects the fact that the RESTART-UNTIL-SUCCESS protocol needs to perform K attempts at success, each of which takes time given by a fresh instance of M (for a more thorough explanation, see [6]).

Now we will prove each of the statements (a-f) from Prop. 4. For statement (a), we need to show that T_{output} is NBU. This follows directly from the following two facts:

(i) NBU-ness is preserved under the maximum: if $T_1, \ldots, T_n$ are NBU random variables, then so is M;

(ii) NBU-ness is preserved under the geometric sum: if M is an NBU random variables, then so is $T_{\text{output}} = \sum_{k=1}^{K} M^{(k)}$.

We prove item (i) in Appendix 7.7.1, while item (ii) was proven by Brown, see Sec. 3.2 in [45] [1].

For proving statement (b), $E[T_{\text{output}}] = m/p$ with $m = E[M]$, we apply a well-known fact of randomised sums called Wald's Lemma [46] to eq. (7.15), which results in

$$E[T_{\text{output}}] = E[M] \cdot E[K]$$

and hence $E[T_{\text{output}}] = m \cdot \frac{1}{p}$.

Statement (c) describes a two-sided bound on the co-CDF of T_{output}:

$$\exp\left(\frac{-p \cdot t}{m} \cdot \frac{1}{1-p}\right) \leq \Pr\big(T_{\text{output}} > t\big) \leq \exp\left(p - \frac{p \cdot t}{m}\right).$$

These bounds follow from the following lemma from Brown, see eq.3.2.4 in [45]:

Lemma 6. *[45] Let X be a real-valued random variable with $\Pr(X < 0) = 0$. Define the geometric compound sum of i.i.d. copies of X as $Y := \sum_{k=1}^{K} X^{(k)}$, where K follows the geometric distribution with success probability p (eq. (7.16)). Moreover, define $Y_0 := \sum_{k=1}^{K_0} X^{(k)}$, where $K_0 = K - 1$. Then*

$$\Pr(Y > t) \leq \exp\big(p\big) \exp\left(-t/E[Y]\right)$$

while

$$\Pr(Y > t) \geq \exp\left(-t/E[Y_0]\right).$$

Now interpret $Y \to T_{\text{output}}$ and $X \to M$ in Lemma 6. The upper bound in statement (c) follows directly from Lemma 6 by the use of statement (b), which says that

[1] Let us clarify here that the work by Brown proves that the NBU property is preserved under the geometric sum if K is distributed according to eq. (7.16). However, the same paper also proves that if K is shifted by 1, i.e. $\Pr(K = k) = p(1-p)^k$, then the geometric sum is *always* NWU, irrespective of the summand random variable. However, we will not use the latter case here.

$E[T_{\text{output}}] = m/p$, while for the lower bound in statement (c) we use

$$\begin{aligned} E[Y_0] &= E[K_0] \cdot E[X] \\ &= E[K_0] \cdot E[M] \\ &= \left(\frac{1}{p} - 1\right) \cdot m \\ &= (1-p) \cdot \frac{m}{p}. \end{aligned}$$

Next, (d) states that $T_{\text{output}}/E[T_{\text{output}}]$ approaches the exponential distribution with mean 1. For proving this statement, we substitute $t \to t \cdot E[T_{\text{output}}] = tm/p$ in statement (c). The result is a bound on

$$\Pr\big(T_{\text{output}} > t \cdot E[T_{\text{output}}]\big) = \Pr\big(T_{\text{output}}/E[T_{\text{output}}] > t\big)$$

given by

$$\exp\left(-t \cdot \frac{1}{1-p}\right) \le \Pr\big(T_{\text{output}}/E[T_{\text{output}}] > t\big) \le \exp\big(p - t\big).$$

Letting $p \to 0$, the bounds on both sides coincide, and thus

$$\lim_{p \to 0} \Pr\big(T_{\text{output}}/E[T_{\text{output}}] > t\big) = \exp(-t)$$

which is precisely the co-CDF of the exponential distribution with parameter 1.

For showing the upper bound in statement (e),

$$m \le \sum_{j=1}^{n} E[T_j]$$

we use the fact that for all $j = 1, \ldots, n$, it holds that $T_j \ge 0$. The maximum of of nonnegative numbers is upper bounded by its sum, and thus

$$\begin{aligned} m &= E[\max(T_1, \ldots, T_n)] \\ &= \sum_{t_1, \ldots, t_n} \Pr(T_1 = t_1, \ldots, T_n = t_n) \max(t_1, \ldots, t_n) \\ &\le \sum_{t_1, \ldots, t_n} \Pr(T_1 = t_1, \ldots, T_n = t_n) (t_1 + \cdots + t_n) \\ &\overset{*}{=} \sum_{j=1}^{n} \sum_{t_j} \Pr\big(T_j = t_j\big) t_j \\ &= E\left[\sum_{j=1}^{n} T_j\right] \end{aligned}$$

where for $*$ we made use of the fact that all T_j are independent. The proof for the lower bound in statement (e), $\max_{1 \le j \le n} E[T_j] \le m$, is similar and relies on the fact that $\max(t_1, \ldots, t_n) \ge t_j$ for all $1 \le j \le n$, where $t_1, \ldots, t_n$ are nonnegative numbers. Last, (f) states that if all T_j are identically distributed with mean $E[T]$, then

$$1 \le \frac{m}{E[T]} \le n - 1 + \frac{1}{n}$$

where we recall that $m = E[\max(T_1,\ldots,T_n)]$. For proving this statement, we need the following lemma from Hu and Lin [47, Lemma 2.2.].

Lemma 7. *[47] If $X_1,\ldots,X_n$ are independent and identically distributed copies of an NBU random variable X on the domain $[0,\infty)$, then $E[\min(X_1,\ldots,X_n)] \geq E[X]/n$.*

Proof. The proof is based on two facts. First, note that

$$\Pr(\min(X_1,\ldots,X_n) > x) = \prod_{j=1}^{n} \Pr\big(X_j > x\big) = \Pr(X > x)^n.$$

Second, note that if X is NBU, then by repeated application of the definition of NBU (Def. 1), we find that

$$\Pr\left(X > \sum_{j=1}^{n} x_j\right) \leq \prod_{j=1}^{n} \Pr\big(X > x_j\big)$$

for any nonnegative numbers $x_j, 1 \leq j \leq n$. When choosing all x_j identical, say, to some constant nonnegative number x, this reduces to

$$\Pr(X > nx) \leq \Pr(X > x)^n.$$

Using these two facts, we can now prove the lemma:

$$\begin{aligned} E[\min(X_1,\ldots,X_n)] &= \int_0^\infty \Pr(X > x)^n dx \\ &\geq \int_0^\infty \Pr(X > nx) dx \\ &= \int_0^\infty \Pr(X/n > x) dx \\ &= E[X/n] \\ &= E[X]/n \end{aligned}$$

where we have used the fact that for any real-valued random variable X with $\Pr(X < 0) = 0$, the mean can be computed as $E[X] = \int_0^\infty \Pr(X > x) dx$. □

Statement (f) is proven by noting that for nonnegative numbers $t_1,\ldots,t_n$, it holds that $t_j \geq \min(t_1,\ldots,t_n)$ for all $j = 1,\ldots,n$, and therefore

$$t_1 + \ldots t_n \geq \max(t_1,\ldots,t_n) + (n-1)\cdot\min(t_1,\ldots,t_n).$$

Translating this to the T_j yields

$$\begin{aligned} E\left[\sum_{j=1}^{n} T_j\right] \quad \geq \quad & (n-1)\cdot E[\min(T_1,\ldots,T_n] \\ & + E[\max(T_1,\ldots,T_n)]. \end{aligned} \tag{7.17}$$

The left hand side of eq. (7.17) equals $n \cdot E[T]$ by the fact that the T_j are i.i.d., while the right hand side is lower bounded by $(n-1)/n \cdot E[T] + E[\max(T_1, \ldots, T_n)]$ by Lemma 7. Reshuffling yields

$$\begin{aligned} E[\max(T_1, \ldots, T_n) &\leq n \cdot E[T] - \frac{n-1}{n} E[T] \\ &= \left(n - 1 + \frac{1}{n}\right) E[T]. \end{aligned}$$

which is what we set out to prove.

7.6. DISCUSSION

The distribution of remote entanglement is a key element of many quantum network applications. In this chapter, we provided analytical bounds on both the mean and quantiles of entanglement delivery times for a large class of protocols. We applied these results to a nested quantum repeater chain scheme and to a quantum switch, and obtained bounds which are tighter than present in the literature.

In particular, we considered a frequently-used approximation to the mean entanglement-delivery time in the nested repeater chain scheme, known as the 3-over-2 formula. This approximation is derived by assuming that the delivery time follows an exponential distribution at each nesting level. It was not known in general how good this approximation is. Moreover, finding the exact mean delivery time has been an open problem for more than ten years [13]. We made a large step towards solving this question by showing that the co-CDF of the delivery time, i.e. the probability that entanglement is delivered after time t, is lower bounded by the co-CDF of an exponential distribution, and upper bounded by the co-CDF of an exponential distribution multiplied by a factor which is independent of t. In the limit of small success probabilities of the repeater's components, the bounds coincide. Second, we show that the 3-over-2 formula is, in essence, an upper bound to the mean delivery time, rendering old analyses building upon this approximation pessimistic.

Regarding future work, note that in many quantum internet scenarios, already-produced entanglement waits for the generation of other entanglement and in the meantime suffers from memory noise. We leave for future work converting our bounds on the delivery time to bounds on the amount of memory noise, and thus on the quality of the produced state.

In this chapter we only focused on the first remote entanglement that is delivered. Some protocols, however, might deliver entanglement while still holding residual entanglement, for example at lower levels in case of the nested repeater chain. In such a case, it is not optimal to restart the protocol for producing a second entangled pair of qubits, since that would require discarding the residual entanglement. Hence, another possibility for future work would be to extend our results to protocols which produce multiple entangled pairs without discarding existing entanglement in between.

Our bounds are partially based on a novel connection with reliability theory. We expect that reliability-theoretic tools will be useful in solving other open problems in quantum networks too.

7.7. APPENDIX

7.7.1. PROOF THAT THE NBU PROPERTY IS PRESERVED UNDER THE MAXIMUM

Here, we prove that the NBU property is preserved under the maximum of independent random variables.

Lemma 8. *Suppose $X_1, \ldots, X_n$ are independent random variables (not necessarily identically distributed). If all X_j are NBU random variables, then so is* $\max(X_1, \ldots, X_n)$.

We first prove the special case for $n = 2$, from which the statement for general n follows.

Lemma 9. *Let A and B be independent nonnegative real-valued random variables (not necessarily identically distributed). If both are NBU, then so is* $\max(A, B)$.

Proof. Let us denote $a_z := \Pr(A > z)$ and $b_z := \Pr(B > z)$ for $z \geq 0$. Assume that A and B possess the NBU property (Def. 1), so that

$$a_{x+y} \leq a_x a_y \text{ and } b_{x+y} \leq b_x b_y \quad \text{for all } x, y \geq 0. \tag{7.18}$$

We also write $m_z := \Pr(\max(A, B) \geq z)$ and compute

$$\begin{aligned} m_z &= \Pr(\max(A, B) > z) \\ &= 1 - \Pr(\max(A, B) \leq z) \\ &= 1 - \Pr(A \leq z) \Pr(B \leq z) \\ &= 1 - (1 - a_z)(1 - b_z) \qquad (7.19) \\ &= a_z + b_z - a_z b_z \\ &= a_z + b_z(1 - a_z). \qquad (7.20) \end{aligned}$$

We will prove that $\max(A, B)$ is NBU, which in our notation becomes $m_{x+y} \leq m_x m_y$ for all $x, y \geq 0$. To begin, we write out the expressions for both sides, i.e. for m_{x+y} and for $m_x m_y$. First, using eq. (7.19), we write out

$$m_{x+y} = 1 - (1 - a_{x+y})(1 - b_{x+y}). \tag{7.21}$$

Since m_{x+y} from eq. (7.21) is monotonically increasing in a_{x+y} and moreover $a_{x+y} \leq a_x a_y$ (eq. (7.18)), we obtain

$$m_{x+y} \leq 1 - (1 - a_x a_y)(1 - b_{x+y}). \tag{7.22}$$

We use the same insight again, but now for b_{x+y}: the right-hand side of eq. (7.22) is monotonically increasing in b_{x+y}, which combined with the fact that $b_{x+y} \leq b_x b_y$ (eq. (7.18)) yields

$$m_{x+y} \leq 1 - (1 - a_x a_y)(1 - b_x b_y) = a_x a_y + b_x b_y (1 - a_x a_y). \tag{7.23}$$

Next, by eq. (7.20) we have

$$\begin{aligned} m_x m_y &= (a_x + b_x(1 - a_x)) \cdot \big(a_y + b_y(1 - a_y)\big) \\ &= a_x a_y + a_x b_y (1 - a_y) + a_y b_x (1 - a_x) + b_x b_y (1 - a_x)(1 - a_y). \qquad (7.24) \end{aligned}$$

In order to prove that $m_{x+y} \leq m_x m_y$ we consider three cases.

- **Case $b_x = 0$.** In this case eq. (7.23) reduces to $m_{x+y} \leq a_x a_y$ and eq. (7.24) becomes

$$m_x m_y = a_x a_y + a_x b_y (1 - a_y). \tag{7.25}$$

  Since a_x, a_y, b_x and b_y are all cumulative probabilities, they take values in the interval $[0,1]$, and therefore the second term of eq. (7.25) is nonnegative, which yields $m_x m_y \geq a_x a_y \geq m_{x+y}$.

- **Case $b_y = 0$.** By the fact that both the right hand side of eq. (7.23) as well as the expression for $m_x m_y$ (eq. (7.24)) are invariant under exchanging b_x and b_y, this case is proven identically to the first case.

- **Case $b_x \neq 0$ and $b_y \neq 0$.** Using eq. (7.23) and eq. (7.24), we expand

$$\begin{aligned}
\frac{m_{x+y} - m_x m_y}{b_x b_y} &= \frac{a_x a_y}{b_x b_y} + \frac{b_x b_y}{b_x b_y}\left(1 - a_x a_y\right) - \frac{a_x a_y}{b_x b_y} - \frac{a_x b_y}{b_x b_y}\left(1 - a_y\right) \\
&\quad - \frac{a_y b_x}{b_x b_y}(1 - a_x) - \frac{b_x b_y}{b_x b_y}(1 - a_x) \cdot \left(1 - a_y\right) \\
&= 1 - a_x a_y - \frac{a_x}{b_x}\left(1 - a_y\right) - \frac{a_y}{b_y}(1 - a_x) - (1 - a_x) \cdot \left(1 - a_y\right)
\end{aligned}$$

  Using the fact that $b_x, b_y \leq 1$, we obtain

$$\frac{m_{x+y} - m_x m_y}{b_x b_y} \leq 1 - a_x a_y - a_x\left(1 - a_y\right) - a_y(1 - a_x) - (1 - a_x) \cdot \left(1 - a_y\right) = 0.$$

  Since b_x and b_y are positive numbers, it follows that $m_{x+y} - m_x m_y \leq 0$. This concludes our proof.

□

Let us now show how Lemma 8 follows from Lemma 9. Let $X_1, \ldots, X_n$ be n NBU independent random variables, for $n \geq 2$. We use induction on n. The case $n = 2$ is proven in Lemma 9. Now suppose Lemma 8 holds for $n = m$ for some $m \geq 2$. We show that Lemma 9 also holds for $n = m + 1$. For this, choose $A = \max(X_1, \ldots, X_m)$ and $B = X_{m+1}$. By assumption, B is NBU, and so is A by the induction hypothesis. Note that

$$\begin{aligned}
\max(X_1, \ldots, X_m, X_{m+1}) &= \max(\max(X_1, \ldots, X_m), X_{m+1}) \\
&= \max(A, B),
\end{aligned}$$

so it follows from Lemma 9 that $\max(X_1, \ldots, X_{m+1})$ is also NBU, which concludes the proof of Lemma 8.

7.7.2. PROOF OF THE LOWER BOUNDS IN PROPOSITION 5

Here, we prove the two lower bounds in Prop. 5: first, Prop. 5(b), followed by the lower bound on the quantiles from Prop. 5(c).

Throughout the appendix, we will use the notation $X^{(1)}, X^{(2)}, \ldots$ to denote independent and identically distributed copies of a random variable X. Before proving the

bounds on the mean and tail of T_n, let us formally define it. Regarding the base case $n = 0$, which describes elementary-link generation between adjacent nodes, we use either of two flavors: we either set $T_0 = T_{\text{gen}}$, i.e. T_0 follows the geometric distribution with parameter p_{gen}, or we set $T_0 = T_{\text{gen}}^{\text{approx}}$, i.e. T_0 follows the exponential distribution with parameter p_{gen}. For each statement about T_n in this appendix, either the statement will hold for both flavors, or it will be clear from the context which of the two flavors is used. Regardless of the choice for $n = 0$, we define T_n for $n > 0$ as

$$T_{n+1} = \sum_{k=1}^{K} M_n^{(k)} \tag{7.26}$$

where K is geometrically distributed with parameter p_{swap} and M_n is defined as

$$M_n = \max(T_n^{(1)}, T_n^{(2)}). \tag{7.27}$$

Eq. (7.26) was given in [6] and can be found by applying eq. (7.15) to each nesting level of the repeater protocol, where $M = M_n$ in eq. (7.15) describes the time until the last of two links, each spanning 2^n repeater segments, has been delivered.

PROOF OF PROPOSITION 5(B)

Here, we will prove the lower bound on the mean completion time T_n of the nested repeater protocol on n nesting levels. Informally stated, the insight is that

$$\max\left(\sum_{k=1}^{K^{(1)}} X^{(k)}, \sum_{k=1}^{K^{(2)}} X^{(k)}\right) \geq_{\text{st}} \sum_{k=1}^{\max(K^{(1)},K^{(2)})} X^{(k)} \quad \text{(informal)}$$

i.e. considering sums with independent and identically distributed summands, the maximum of two sums stochastically dominates the "longest" of the two. Since the definition of M_n in eq. (7.27) contains the maximum of two such sums, we use this idea to define a new random variable R_n as the "longest" of the two sums; by the insight above, R_n is stochastically dominated by M_n. Using Lemma 4, this stochastic domination can be converted to $E[M_n] \geq E[R_n]$, after which the bound on the mean of T_n as described in Prop. 5(b) follows by noting that $E[T_n] = E[M_n]/p_{\text{swap}}$.

We now give the formal proof, which we divide into three steps. First, we define R_n and compute its mean. Next, we show that $M_n \geq_{\text{st}} R_n$ for all $n > 0$, from which we infer a lower bound on the mean of T_n as third step.

For the first step, we define R_n:

$$\begin{aligned} R_0 &= \max(T_0^{(1)}, T_0^{(2)}), \\ R_{n+1} &= \sum_{j=1}^{N} R_n^{(j)} \quad \text{for } n \geq 0. \end{aligned}$$

Here, $N = \max\left(K^{(1)}, K^{(2)}\right)$ where $K^{(1)}$ and $K^{(2)}$ are both geometrically distributed with parameter p_{swap}. We emphasise that contrary to T_n, the random variable R_n does not correspond to the completion time of a protocol.

The mean of R_n is computed using the following two lemmas.

Lemma 10. *Let $X^{(1)}$ and $X^{(2)}$ be independent and identically distributed random variables with mean $1/p$ for some $0 < p \leq 1$. If both $X^{(1)}$ and $X^{(2)}$ follow a geometric distribution, then*

$$E[\max(X^{(1)}, X^{(2)})] = \frac{3-2p}{p(2-p)}$$

while if they follow an exponential distribution, then

$$E[\max(X^{(1)}, X^{(2)})] = \frac{3}{2p}.$$

Proof. We start with the case that X follows a geometric distribution. Note that $\min(X^{(1)}, X^{(2)})$ is geometrically distributed with parameter $1-(1-p)^2$:

$$\Pr\big(\min(X^{(1)}, X^{(2)}) > t\big) = \Pr\big(X^{(1)} > t\big)\Pr\big(X^{(2)} > t\big) = (1-p)^t \cdot (1-p)^t = (1-p)^{2t} = \big[1-\big(1-(1-p)^2\big)\big]^t$$

for $t = 0,1,2,\ldots$. Combined with the fact that $E[\max(X^{(1)}, X^{(2)})] = E[X^{(1)} + X^{(2)} - \min(X^{(1)}, X^{(2)})] = E[X^{(1)}] + E[X^{(2)}] - E[\min(X^{(1)}, X^{(2)})]$, we obtain

$$E[\max(X^{(1)}, X^{(2)})] = \frac{1}{p} + \frac{1}{p} - \frac{1}{1-(1-p)^2} = \frac{3-2p}{p(2-p)}$$

The case of the exponential distribution is analogous, with $\min(X^{(1)}, X^{(2)})$ following the exponential distribution with parameter $2p$. □

Lemma 11. *The mean of R_n is*

$$E[R_n] = \left(\frac{3-2p_{\text{swap}}}{p_{\text{swap}}(2-p_{\text{swap}})}\right)^n \cdot v_0 \tag{7.28}$$

where v_0 is defined as follows. If T_0, which describes elementary-link generation between adjacent nodes, follows the geometric distribution with parameter p_{gen}, then

$$v_0 = E[R_0] = E[\max(T_0^{(1)}, T_0^{(2)})] = \frac{3-2p_{\text{gen}}}{p_{\text{gen}}(2-p_{\text{gen}})} \tag{7.29}$$

while if T_0 follows the exponential distribution with parameter p_{gen}, then

$$v_0 = E[R_0] = E[\max(T_0^{(1)}, T_0^{(2)})] = \frac{3}{2p_{\text{gen}}}. \tag{7.30}$$

Proof. We use induction on n. The case $n = 0$ is treated in Lemma 10 where we set $p = p_{\text{gen}}$. For the induction case, we note that

$$E[R_{n+1}] = E\left[\sum_{j=1}^{N} R_n^{(j)}\right] = E[N] \cdot E[R_n]$$

by Wald's Lemma [46]. Since $N = \max(K^{(1)}, K^{(2)})$ and K is geometrically distributed with parameter p_{swap}, we again invoke Lemma 10 to obtain

$$E[N] = E[\max(K^{(1)}, K^{(2)})] = \frac{3-2p_{\text{swap}}}{p_{\text{swap}}(2-p_{\text{swap}})}.$$

This finishes the proof. □

As second step, we will show that M_n stochastically dominates R_n, for which we need the following two auxiliary lemmas and corollary.

Lemma 12. *Let P and Q be independent real-valued random variables, and P' and Q' i.i.d. copies of P and Q respectively. Then $P \geq_{\text{st}} Q$ implies $\max(P,P') \geq_{\text{st}} \max(Q,Q')$.*

Proof. By definition of $P \geq_{\text{st}} Q$, we have, for all real numbers z, that $\Pr(P > z) \geq \Pr(Q > z)$ and therefore $\Pr(P \leq z) \leq \Pr(Q \leq z)$. Consequently,

$$\Pr\big(\max(P,P') > z\big) = 1 - \Pr\big(\max(P,P') \leq z\big) = 1 - \Pr(P \leq z)^2 \geq 1 - \Pr(Q \leq z)^2 = \Pr\big(\max(Q,Q') > z\big)$$

for all real numbers z, so $\max(P,P') \geq_{\text{st}} \max(Q,Q')$. □

Lemma 13. *Let P and Q be independent, real-valued random variables with identical domain. Then $\max(P,Q) \geq_{\text{st}} Q$.*

Proof. For any real number z, we have

$$\Pr(\max(P,Q) > z) = 1 - \Pr(\max(P,Q) \leq z) = 1 - \Pr(P \leq z)\Pr(Q \leq z) \overset{*}{\geq} 1 - \Pr(Q \leq z) = \Pr(Q > z)$$

where the inequality * holds because $\Pr(P < z) \leq 1$. □

Corollary 1. *Let $A^{(1)}, A^{(2)}, A^{(3)}$ and $A^{(4)}$ be independent and identically distributed random variables with domain $\{1,2,3,\ldots\}$. Furthermore, let X, Y and Z be independent and identically distributed random variables with domain $[0,\infty)$. Then*

$$\max\left(\sum_{a=1}^{A^{(1)}} X^{(a)}, \sum_{b=1}^{A^{(2)}} Y^{(b)}\right) \geq_{\text{st}} \sum_{a=1}^{\max\left(A^{(3)},A^{(4)}\right)} Z^{(a)}. \tag{7.31}$$

Proof. We note that random sums occur on both sides of eq. (7.31), that is, sums whose number of terms is a random variable. We expand both sides of the inequality from the lemma as a weighted sum over instantiations of this random variable. For the left-hand-side, we obtain

$$\Pr\left(\max\left(\sum_{a=1}^{A^{(1)}} X^{(a)}, \sum_{b=1}^{A^{(2)}} Y^{(b)}\right) > y\right) = \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} \Pr\big(A^{(1)} = i\big) \cdot \Pr\big(A^{(2)} = j\big) \cdot C_{ij}^{y}$$

for $y \geq 0$, where we have defined

$$C_{ij}^{y} := \Pr\left(\max\left(\sum_{a=1}^{i} X^{(a)}, \sum_{b=1}^{j} Y^{(b)}\right) > y\right)$$

and for the right-hand-side we get

$$\Pr\left(\sum_{a=1}^{\max\left(A^{(3)},A^{(4)}\right)} Z^{(a)} > y\right) = \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} \Pr\big(A^{(3)} = i\big) \cdot \Pr\big(A^{(4)} = j\big) \cdot D_{ij}^{y}$$

with

$$D_{ij}^{y} := \Pr\left(\sum_{a=1}^{\max(i,j)} Z^{(a)} > y\right).$$

Given fixed i and j, we define random variables P and Q as follows:

- if $\max(i,j) = i > j$, then define $P = \sum_{b=1}^{j} Y^{(b)}$ and $Q = \sum_{a=1}^{i} X^{(a)}$;
- if $\max(i,j) = j$, then define $P = \sum_{a=1}^{i} X^{(a)}$ and $Q = \sum_{b=1}^{j} Y^{(b)}$;

In both cases, application of Lemma 13 that $\max(P,Q) \geq_{\text{st}} Q$ yields $C_{ij}^{y} \geq \Pr\left(\sum_{a=1}^{\max(i,j)} Y^{(a)} > y\right)$. Since Y and Z are i.i.d., we obtain $C_{ij}^{y} \geq D_{ij}^{y}$ for all $y \geq 0$ and for all i, j. This concludes the proof. □

Now we have the tools to show that M_n stochastically dominates R_n, as described in the following lemma.

Lemma 14. *For all $n \geq 0$, we have*

$$M_n \geq_{\text{st}} R_n$$

where $M_n = \max(T_n^{(1)}, T_n^{(2)})$ as defined in eq. (7.27).

Proof. We use induction on n. The base case $n = 0$ is an equality by definition of R_0. Now assume the statement from the lemma holds for $n = m$. We will show it also holds for $n = m+1$. First, we expand the definition of T_{m+1}:

$$T_{m+1} = \sum_{k=1}^{K} \max(T_m^{(1)}, T_m^{(2)})$$

Now apply the induction hypothesis:

$$T_{m+1} \geq_{\text{st}} \sum_{k=1}^{K} R_m^{(k)}.$$

Using Lemma 12 we obtain

$$\max(T_{m+1}^{(1)}, T_{m+1}^{(2)}) \geq_{\text{st}} \max\left(\sum_{j=1}^{K^{(1)}} R_m^{(i)}, \sum_{j=1}^{K^{(2)}} R_m^{(j)}\right).$$

Applying Corollary 1 to the previous equation yields

$$\max(T_{m+1}^{(1)}, T_{m+1}^{(2)}) \geq_{\text{st}} \sum_{k=1}^{\max(K^{(1)},K^{(2)})} R_m^{(k)}.$$

The left-hand side of the previous equation equals M_{m+1} by definition, while its right-hand side is R_{m+1}, again by definition. This concludes the proof. □

The third step is to derive the lower bound on the mean delivery time from Prop. 5. This follows directly from Lemma 14, as expressed in the following corollary.

Corollary 2. (Lower bound from Prop. 5) *For $n > 0$, it holds that*

$$E[T_n] \geq \frac{1}{p_{\text{swap}}} \cdot \left(\frac{3 - 2p_{\text{swap}}}{p_{\text{swap}}(2 - p_{\text{swap}})}\right)^{n-1} \cdot v_0$$

where v_0 is given in eq. (7.29) or eq. (7.30), depending on whether elementary-link generation is modelled following a geometric or exponential distribution, respectively.

Proof. By Wald's Lemma [46], it follows from the definition of T_n for $n > 0$ that $E[T_n] = E[K] \cdot E[M_{n-1}] = \frac{1}{p_{\text{swap}}} \cdot E[M_{n-1}]$. A lower bound on $E[M_n]$ follows from Lemma 4 and Lemma 14, resulting into

$$E[T_n] = \frac{1}{p_{\text{swap}}} \cdot E[M_{n-1}] \geq \frac{1}{p_{\text{swap}}} \cdot E[R_{n-1}].$$

The proof finishes by substituting $E[R_{n-1}]$ by the right-hand side of eq. (7.28). □

PROOF OF LOWER BOUND IN PROPOSITION 5(B)

Here, we provide the expression for m_{lower} in Prop. 5(c), which is a lower bound to the mean of the delivery time after both input links are ready, but before the entanglement swap. Formally, m_{lower} is a lower bound to the mean of M_{n-1} from eq. (7.27). Such a bound follows directly from Lemma 14 by the fact that $X \geq_{\text{st}} Y$ implies $E[X] \geq E[Y]$ (see Lemma 4):

$$m_{\text{lower}} = E[R_{n-1}]$$

and $E[R_{n-1}]$ is given in eq. (7.28).

REFERENCES

[1] M. Schmitt, J. Redi, P. Cesar, and D. Bulterman, *1Mbps is enough: Video quality and individual idiosyncrasies in multiparty HD video-conferencing*, in *2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX)* (2016) pp. 1–6.

[2] N. Kalb, P. C. Humphreys, J. J. Slim, and R. Hanson, *Dephasing mechanisms of diamond-based nuclear-spin memories for quantum networks*, Phys. Rev. A **97**, 062330 (2018).

[3] M. F. Askarani, T. Lutz, M. G. Puigibert, N. Sinclair, D. Oblak, and W. Tittel, *Persistent atomic frequency comb based on Zeeman sub-levels of an erbium-doped crystal waveguide*, J. Opt. Soc. Am. B **37**, 352 (2020).

[4] R. V. Meter, T. D. Ladd, W. J. Munro, and K. Nemoto, *System design for a long-line quantum repeater*, IEEE/ACM Transactions on Networking **17**, 1002 (2009).

[5] E. Shchukin, F. Schmidt, and P. van Loock, *Waiting time in quantum repeaters with probabilistic entanglement swapping*, Phys. Rev. A **100**, 032322 (2019).

[6] S. Brand, T. Coopmans, and D. Elkouss, *Efficient computation of the waiting time and fidelity in quantum repeater chains*, IEEE Journal on Selected Areas in Communications **38**, 619 (2020).

[7] B. Li, T. Coopmans, and D. Elkouss, *Efficient optimization of cutoffs in quantum repeater chains*, IEEE Transactions on Quantum Engineering **2**, 1 (2021).

[8] V. V. Kuzmin, D. V. Vasilyev, N. Sangouard, W. Dür, and C. A. Muschik, *Scalable repeater architectures for multi-party states*, npj Quantum Information **5**, 115 (2019).

[9] M. Caleffi, *Optimal routing for quantum networks,* IEEE Access **5**, 22299 (2017).

[10] S. E. Vinay and P. Kok, *Statistical analysis of quantum-entangled-network generation,* Phys. Rev. A **99**, 042313 (2019).

[11] G. Vardoyan, S. Guha, P. Nain, and D. Towsley, *On the stochastic analysis of a quantum entanglement switch,* SIGMETRICS Perform. Eval. Rev. **47**, 27 (2019).

[12] S. Khatri, C. T. Matyas, A. U. Siddiqui, and J. P. Dowling, *Practical figures of merit and thresholds for entanglement distribution in quantum networks,* Phys. Rev. Research **1**, 023032 (2019).

[13] N. Sangouard, C. Simon, H. de Riedmatten, and N. Gisin, *Quantum repeaters based on atomic ensembles and linear optics,* Rev. Mod. Phys. **83**, 33 (2011).

[14] V. V. Kuzmin and D. V. Vasilyev, *Diagrammatic technique for simulation of large-scale quantum repeater networks with dissipating quantum memories,* Physical Review A **103**, 032618 (2021).

[15] F. Schmidt and P. van Loock, *Memory-assisted long-distance phase-matching quantum key distribution,* Phys. Rev. A **102**, 042614 (2020).

[16] O. A. Collins, S. D. Jenkins, A. Kuzmich, and T. A. B. Kennedy, *Multiplexed memory-insensitive quantum repeaters,* Phys. Rev. Lett. **98**, 060502 (2007).

[17] N. K. Bernardes, L. Praxmeyer, and P. van Loock, *Rate analysis for a hybrid quantum repeater,* Phys. Rev. A **83**, 012323 (2011).

[18] L. Praxmeyer, *Reposition time in probabilistic imperfect memories,* arXiv:1309.3407 (2013), arXiv:1309.3407 .

[19] J. Calsamiglia and N. Lütkenhaus, *Maximum efficiency of a linear-optical bell-state analyzer,* Applied Physics B **72**, 67 (2001).

[20] S. Santra, L. Jiang, and V. S. Malinovsky, *Quantum repeater architecture with hierarchically optimized memory buffer times,* Quantum Science and Technology **4**, 025010 (2019).

[21] K. Chakraborty, F. Rozpędek, A. Dahlberg, and S. Wehner, *Distributed routing in a quantum internet,* arXiv:1907.11630 (2019), arXiv:1907.11630 .

[22] H.-J. Briegel, W. Dür, J. I. Cirac, and P. Zoller, *Quantum repeaters: The role of imperfect local operations in quantum communication,* Phys. Rev. Lett. **81**, 5932 (1998).

[23] L.-M. Duan, M. D. Lukin, J. I. Cirac, and P. Zoller, *Long-distance quantum communication with atomic ensembles and linear optics,* Nature **414**, 413 EP (2001), article.

[24] L. Jiang, J. M. Taylor, and M. D. Lukin, *Fast and robust approach to long-distance quantum communication with atomic ensembles,* Phys. Rev. A **76**, 012301 (2007).

[25] C. Simon, H. de Riedmatten, M. Afzelius, N. Sangouard, H. Zbinden, and N. Gisin, *Quantum repeaters with photon pair sources and multimode memories,* Phys. Rev. Lett. **98**, 190503 (2007).

[26] J. B. Brask and A. S. Sørensen, *Memory imperfections in atomic-ensemble-based quantum repeaters,* Phys. Rev. A **78**, 012350 (2008).

[27] N. Sangouard, C. Simon, J. c. v. Minář, H. Zbinden, H. de Riedmatten, and N. Gisin, *Long-distance entanglement distribution with single-photon sources,* Phys. Rev. A **76**, 050301 (2007).

[28] N. Sangouard, C. Simon, B. Zhao, Y.-A. Chen, H. de Riedmatten, J.-W. Pan, and N. Gisin, *Robust and efficient quantum repeaters with atomic ensembles and linear optics,* Phys. Rev. A **77**, 062301 (2008).

[29] N. Sangouard, R. Dubessy, and C. Simon, *Quantum repeaters based on single trapped ions,* Phys. Rev. A **79**, 042340 (2009).

[30] S. Abruzzo, S. Bratzik, N. K. Bernardes, H. Kampermann, P. van Loock, and D. Bruß, *Quantum repeaters and quantum key distribution: Analysis of secret-key rates,* Phys. Rev. A **87**, 052315 (2013).

[31] W. J. Munro, K. Azuma, K. Tamaki, and K. Nemoto, *Inside quantum repeaters,* IEEE Journal of Selected Topics in Quantum Electronics **21**, 78 (2015).

[32] K. Boone, J.-P. Bourgoin, E. Meyer-Scott, K. Heshami, T. Jennewein, and C. Simon, *Entanglement over global distances via quantum repeaters with satellite links,* Phys. Rev. A **91**, 052325 (2015).

[33] S. Muralidharan, L. Li, J. Kim, N. Lütkenhaus, M. D. Lukin, and L. Jiang, *Optimal architectures for long distance quantum communication,* Scientific Reports **6**, 20463 EP (2016), article.

[34] F. Kimiaee Asadi, N. Lauk, S. Wein, N. Sinclair, C. O'Brien, and C. Simon, *Quantum repeaters with individual rare-earth ions at telecommunication wavelengths,* Quantum **2**, 93 (2018).

[35] N. Lo Piparo, W. J. Munro, and K. Nemoto, *Quantum multiplexing,* Phys. Rev. A **99**, 022337 (2019).

[36] F. K. Asadi, S. C. Wein, and C. Simon, *Long-distance quantum communication with single ^{167}Er ions,* (2020), arXiv:2004.02998 .

[37] K. Sharman, F. K. Asadi, S. C. Wein, and C. Simon, *Quantum repeaters based on individual electron spins and nuclear-spin-ensemble memories in quantum dots,* arXiv:2010.13863 (2020).

[38] Y. Wu, J. Liu, and C. Simon, *Near-term performance of quantum repeaters with imperfect ensemble-based quantum memories,* Phys. Rev. A **101**, 042301 (2020).

[39] C. Liorni, H. Kampermann, and D. Bruss, *Quantum repeaters in space,* arXiv:2005.10146 (2020).

[40] N. H. Nickerson, Y. Li, and S. C. Benjamin, *Topological quantum computing with a very noisy network and local error rates approaching one percent,* Nature Communications **4** (2013), 10.1038/ncomms2773.

[41] A. W. Marshall and I. Olkin, *Nonparametric families: Origins in reliability theory,* in *Life Distributions: Structure of Nonparametric, Semiparametric, and Parametric Families* (Springer New York, New York, NY, 2007) pp. 137–193.

[42] B. Eisenberg, *On the expectation of the maximum of IID geometric random variables,* **78**, 135 (2008), 135.

[43] V. Kowalenko, *Properties and applications of the reciprocal logarithm numbers,* Acta Applicandae Mathematicae **109**, 413 (2008).

[44] F. Topsøe, *Some bounds for the logarithmic function,* Inequality theory and applications **4** (2007).

[45] M. Brown, *Error bounds for exponential approximations of geometric convolutions,* Ann. Probab. **18**, 1388 (1990).

[46] A. Wald, *Sequential Analysis* (Courier Corporation, Dover, New York, 1947).

[47] C.-Y. Hu and G. D. Lin, *Characterizations of the exponential distribution by stochastic ordering properties of the geometric compound,* Annals of the Institute of Statistical Mathematics **55**, 499 (2003).

II

SIMULATION OF DETAILED MODELS OF QUANTUM NETWORKS

8

NetSquid, a NETwork Simulator for QUantum Information using Discrete events

Abstract

In this chapter, we introduce NetSquid, a discrete-event based platform for simulating all aspects of quantum networks and modular quantum computing systems, ranging from the physical layer and its control plane up to the application level. Using a simulator such as NetSquid allows us to (more easily) study more complex scenarios than with the analytical and semi-analytical tools from the previous chapters. In particular, it allows us to investigate more detailed hardware models, to better determine the requirements for realizing quantum network protocols. We study two use cases to showcase NetSquid's power. First, a detailed physical layer simulations of repeater chains based on nitrogen vacancy centres in diamond. Next, we investigate the control plane of a quantum switch beyond its analytically known regime. We showcase NetSquid's ability to investigate large networks by simulating entanglement distribution over a chain of up to one thousand nodes.

A modified version of this chapter has published as: T.Coopmans*, R. Knegjens*, A. Dahlberg, D. Maier, L. Nijsten, J. de Oliveira Filho, M. Papendrecht, J. Rabbie, F. Rozpędek, M. Skrzypczyk, L. Wubben, W. de Jong, D. Podareanu, Ariana Torres-Knoop, D. Elkouss†, S. Wehner†, *NetSquid, a NETwork Simulator for QUantum Information using Discrete events*, Nature Communications Physics (2021), where * denoted equally-contributing authors and † denotes authors who jointly supervised the work.

8.1. INTRODUCTION

For bringing quantum networks and distributed quantum computing systems to the real world, many challenges must be overcome before they can fulfil their promise. The exact extent of these challenges remains generally unknown, and precise requirements to guide the construction of large-scale quantum networks are missing. At the physical layer, many proposals exist for quantum repeaters that can carry qubits over long distances (see e.g. [1–3] for an overview). Using analytical methods [4–25] and ad-hoc simulations [26–33] rough estimates for the requirements of such hardware proposals have been obtained. Yet, while greatly valuable to set minimal requirements, these studies still provide limited detail. Even for a small-scale quantum network, the intricate interplay between many communicating devices, and the resulting timing dependencies makes a precise analysis challenging. To go beyond, we would like a tool that can incorporate not only a detailed physical modelling, but also account for the implications of time-dependent behaviour.

Quantum networks cannot be built from quantum hardware alone; in order to scale they need a tightly integrated classical control plane, i.e. protocols responsible for orchestrating quantum network devices to bring entanglement to users. Fundamental differences between quantum and classical information demand an entirely new network stack in order to create entanglement, and run useful applications on future quantum networks [34–39]. The design of such a control stack is furthermore made challenging by numerous technological limitations of quantum devices. A good example is given by the limited lifetimes of quantum memories, due to which delays in the exchange of classical control messages have a direct impact on the performance of the network. To succeed, we hence need to understand how possible classical control strategies do perform on specific quantum hardware. Finally, to guide overall development, we need to understand the requirements of quantum network applications themselves. Apart from quantum key distribution (QKD) [40–44] and a few select applications [45–48], little is known about the requirements of quantum applications [49] on imperfect hardware.

Analytical tools offer only a limited solution for our needs. Statistical tools (see e.g. [50–53]) have been used to make predictions about certain aspects of large regular networks using simplified models, but are of limited use for more detailed studies. Information theory [54] can be used to benchmark implementations against the ideal performance. However, it gives no information about how well a specific proposal will perform. As a consequence, numerical methods are of great use to go beyond what is feasible using an analytical study. Ad-hoc simulations of quantum networks have indeed been used to provide further insights on specific aspects of quantum networks (see e.g. [26–33, 55–57]). However, while greatly informative, setting up ad-hoc simulations for each possible networking scenario to a level of detail that might allow the determination of more precise requirements is cumbersome, and does not straightforwardly lend itself to extensive explorations of new possibilities.

We would hence like a simulation platform that satisfies at least the following three features: First, accuracy: the tool should allow modelling the relevant physics. This includes the ability to model time-dependent noise and network behaviour. Second, modularity: it should allow stacking protocols and models together in order to construct complicated network simulations out of simple components. This includes the abil-

ity to investigate not only the physical layer hardware, but the entirety of the quantum network system including how different control protocols behave on a given hardware setup. Third, scalability: it should allow us to investigate large networks.

Evaluating the performance of large classical network systems, including their time-dependent behaviour is the essence of classical network analysis. Yet, even for classical networks, the predictive power of analytical methods is limited due to complex emergent behaviour arising from the interplay between many network devices. Consequently, a crucial tool in the design of such networks are network simulators, which form a tool to test new ideas, and many such simulators exist for purely classical networks [58–60]. However, such simulators do not allow the simulation of quantum behaviour.

In the quantum domain, many simulators are known for the simulation of quantum computers (see e.g. [61]). However, the task of simulating a quantum network differs greatly from simulating the execution of one monolithic quantum system. In the network, many devices are communicating with each other both quantumly and classically, leading to complex stochastic behaviour, and asynchronous and time-dependent events. From the perspective of building a simulation engine, there is also an important difference that allows for gains in the efficiency of the simulation. A simulator for a quantum computation is optimised to track large entangled states. In contrast, in a quantum network the state space grows and shrinks dynamically as qubits get measured or entangled, but for many protocols, at any moment in time the state space describing the quantum state of the network is small. We would thus like a simulator capable of exploiting this advantage.

In this chapter we introduce the quantum network simulator NetSquid: the NETwork Simulator for QUantum Information using Discrete events. NetSquid is a software tool (available as a package for Python and previously made freely available online [62]) for accurately simulating quantum networking and modular computing systems that are subject to physical non-idealities. It achieves this by integrating several key technologies: a discrete-event simulation engine, a specialised quantum computing library, a modular framework for modelling quantum hardware devices, and an asynchronous programming framework for describing quantum protocols. We showcase the utility of this tool for a range of applications by studying use cases: the analysis of a control plane protocol beyond its analytically accessible regime, predicting the performance of protocols on realistic near-term hardware, and benchmarking different quantum devices. These use cases, in combination with a scalability analysis, demonstrate that NetSquid achieves all three features set forth above. Furthermore, they show its potential as a general and versatile design tool for quantum networks, as well as for modular quantum computing architectures.

8.2. RESULTS AND DISCUSSION

8.2.1. NETSQUID IN A NUTSHELL

Simulating a quantum network with NetSquid is generally performed in three steps. Firstly, the network is modelled using a modular framework of components and physical models. Next, protocols are assigned to network nodes to describe the intended behaviour. Finally, the simulation is executed for a typically large number of independent

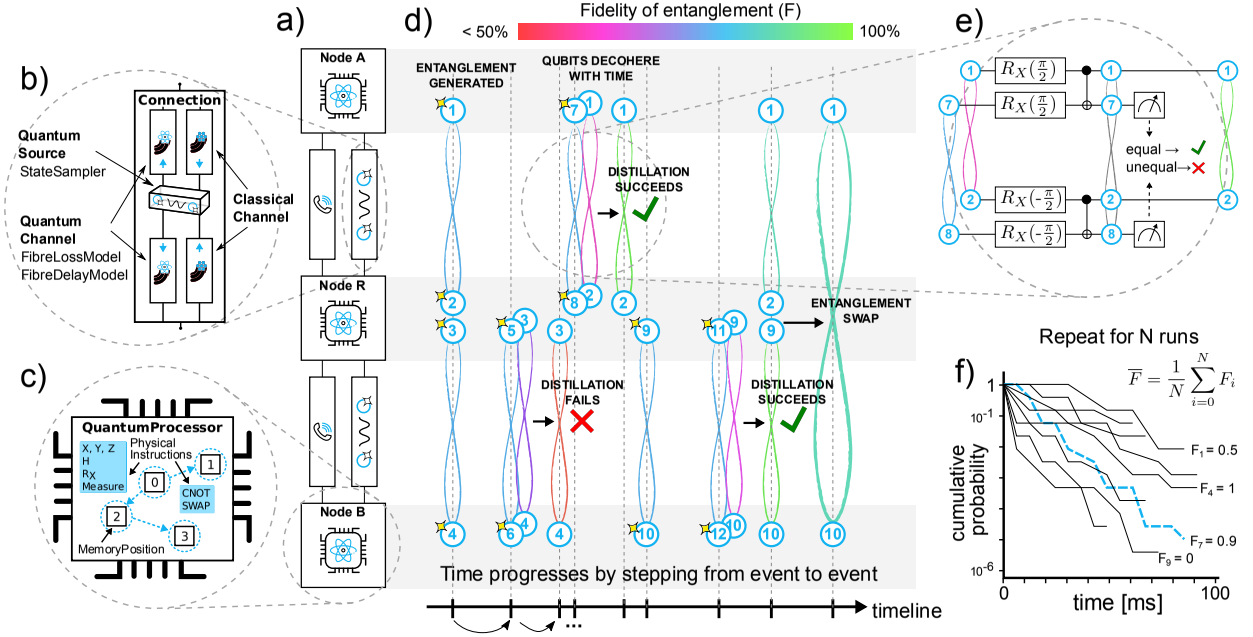


Figure 8.1: **Illustrative example of a NetSquid use case.** Each sub-figure explains part of the modelling and simulation process. For greater clarity the figures are not based on real simulation data. The scenario shown is a quantum repeater utilising entanglement distillation (see main text). **a)** The setup of a quantum network using node and connection components. **b)** A zoom in showing the subcomponents of the entangling connection component. The quantum channels are characterised using fibre delay and loss models. The quantum source samples from an entangled bipartite state sampler when externally triggered by the classical channel. **c)** A zoom in of the quantum memory positions within a quantum processor illustrating their physical gate topology. The physical single-qubit instructions possible on each memory in this example are the Pauli (X, Y, Z), Hadamard (H), and X-rotation (R_X) gates, and measurement. The blue-dashed arrows show the positions and control direction (where applicable) for which the two-qubit instructions controlled-X (CNOT) and swap are possible. Noise and error models for the memories and gates are also assigned. **d)** Illustration of a single simulation run. Time progresses by discretely stepping from event to event, with new events generated as the simulation proceeds. Qubits are represented by circles, which are numbered according to the order they were generated. A star shows the moment of generation. The curved lines between qubits denote their entanglement with the colour indicating fidelity. The state of each qubit is updated as it is accessed during the simulation, for instance to apply time-dependent noise from waiting in memory. **e)** A zoom in of the distillation protocol. The shared quantum states of the qubits are combined in an entangling step, which then shrinks as two of the qubits are measured. The output is randomly sampled, causing the simulation to choose one of two paths by announcing success or failure. **f)** A plot illustrating the stochastic paths followed by multiple independent simulation runs over time, labelled by their final end-to-end fidelity F_i. The blue dashed line corresponds to the run shown in panel (d). The runs are typically executed in parallel. Their results are statistically analysed to produce performance metrics such as the average outcome fidelity and run duration.

runs to collect statistics with which to determine the performance of the network. To explain these steps and the features involved further, we consider a simple use case for illustration. For a more detailed presentation of the available functionality and design of the NetSquid framework see section 8.3.1 of the Methods.

The scenario we will consider is the analysis of an entanglement distribution protocol over a quantum repeater chain with three nodes. The goal of the analysis is to estimate the average output fidelity of the distributed entangled pairs. The entanglement distribution protocol is depicted in Figure 8.1(d-e). It works as follows. First, the intermediate node generates two entangled pairs with each of its adjacent neighbours. Entanglement generation is modelled as a stochastic process that succeeds with a certain probability at every attempt. When two pairs are ready at one of the links, the DEJMPS entanglement distillation scheme [63] is run to improve the quality of the entanglement. If it fails, the two links are discarded and the executing nodes restart entanglement generation. Once both distilled states are ready, the intermediate node swaps the entanglement to achieve end-to-end entanglement. We remark that already this simple protocol is rather involved to analyse.

We begin by modelling the network. The basic element of NetSquid's modular framework is the "component". It is capable of describing the physical model composition, quantum and classical communication ports, and, recursively, any subcomponents. All hardware elements, including the network itself, are represented by components. For this example we require three remote nodes linked by two quantum and two classical connections, the setup of which is shown in Figure 8.1(a). In Figure 8.1(b,c) the nested structure of these components is highlighted. A selection of physical models is used to describe the loss and delay of the fibre optic channels, the decoherence of the quantum memories, and the errors of quantum gates.

Quantum information in NetSquid is represented at the level of qubits, which are treated as objects that dynamically share their quantum states. These internally shared states will automatically merge or "split" – a term we use to mean the separation of a tensor product state into two separately shared sub-states – as qubits entangle or are measured, as illustrated by the distillation protocol in Figure 8.1(e). The states are tracked internally, i.e. hidden from users, for two reasons: to encourage a node-centric approach to programming network protocols, and to allow a seamless switching between different quantum state representations. The representations offered by NetSquid are ket vectors, density matrices, stabiliser tableaus and graph states with local Cliffords, each with trade-offs in modelling versatility, computation speed and network (memory) scalability (see the subsection 8.2.4 below and Supplementary Note 8.6.1).

Discrete-event simulation, an established method for simulating classical network systems [64], is a modelling paradigm that progresses time by stepping through a sequence of events – see Figure 8.2 for a visual explanation. This allows the simulation engine to efficiently handle the control processes and feedback loops characteristic of quantum networking systems, while tracking quantum state decoherence based on the elapsed time between events. A novel requirement for its application to quantum networks is the need to accurately evolve the state of the quantum information present in a network with time. This can be achieved by retroactively updating quantum states when the associated qubits are accessed during an event. While it is possible to efficiently

track a density matrix, quantum operations requiring a singular outcome for classical decision making, for instance a quantum measurement, must be probabilistically sampled. A single simulation run thus consists of a sequence of random choices that forms one of many possible paths. In Figure 8.1 (d) we show such a run for the repeater protocol example, which demonstrates the power of the discrete-event approach for tracking qubit decoherence and handling feedback loops.

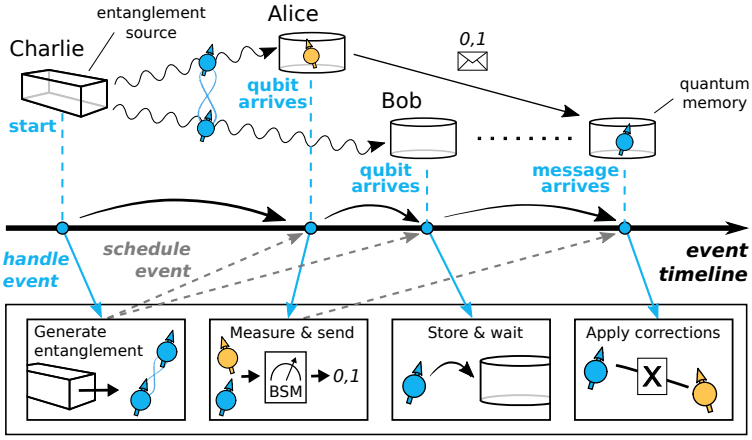


Figure 8.2: **Abstract example of simulating a quantum protocol with discrete events**. When setting up the simulation, protocol actions are defined to occur when a specific event occurs, as in the setup of a real system. Instead of performing a continuous time evolution, the simulation advances to the next event, and then automatically executes the actions that should occur when the event takes place. Any action may again define future events to be triggered after a certain (stochastic) amount of time has elapsed. For concreteness a simplified quantum teleportation example is shown where a qubit, shown as an orange circle with arrow, is teleported between the quantum memories of Alice and Bob. Here, entanglement is produced using an abstract source sending two qubits, shown as blue circles with arrows, to Alice and Bob. Once the qubit has traversed the fibre and reaches Alice's lab, an event is triggered that invokes the simulation of Alice's Bell state measurement (BSM) apparatus. The simulation engine steps from event to events defined by the next action, which generally occur at irregular intervals. This approach allows time-dependent physical non-idealities, such as quantum decoherence, to be accurately tracked.

The performance metrics of a simulation are determined statistically from many runs. Due to the independence of each run, simulations can be massively parallelised and thereby efficiently executed on computing clusters. For the example at hand we choose as metrics the output fidelity and run duration. In Figure 8.1 (f) the sampled run from (d), which resulted in perfect fidelity, is plotted in terms of its likelihood and duration together with several other samples, some less successful. By statistically averaging all of the sampled runs the output fidelity and duration can be estimated.

In the following sections, we will outline two use cases of NetSquid: first, a quantum switch, followed by simulations of quantum repeaters based on nitrogen-vacancy technology. We will also benchmark NetSquid's scalability in both quantum state size and number of quantum network nodes. Although the use cases each provide relevant insights into the performance of the studied hardware and protocols, we emphasise that one can use NetSquid to simulate arbitrary network topologies.

8.2.2. SIMULATING A QUANTUM NETWORK SWITCH BEYOND ITS ANALYTICALLY KNOWN REGIME

As a first use case showcasing the power of NetSquid, we study the control plane of a recently introduced quantum switch beyond the regime for which analytical results have been obtained, including its performance under time-dependent memory noise.

The switch is a node which is directly connected to each of k users by an optical link. The communications task is distributing Bell pairs and n-partite Greenberger-Horne-Zeilinger (GHZ) states [65] between $n \leq k$ users. The switch achieves this by connecting Bell pairs which are generated at random intervals on each link. See Figure 8.3.

Intuitively, the switch can be regarded as a generalisation of a simple repeater performing entanglement swapping with added logic to choose which parties to link. Even with a streamlined physical model, it is already rather challenging to analytically characterise the switch use case [52].

In the following, we recover via simulation a selection of the results from Vardoyan et al. [52], who studied the switch as the central node in a star network, and extend them in two directions. First, we increase the range of parameters for which we can estimate entanglement rates using the same model as used in the work of Vardoyan et al. Second, simulation enables us to investigate more sophisticated models than the exponentially distributed erasure process from their work, in particular we analyse the behaviour of a switch in the presence of memory dephasing noise.

The protocol for generating the target n-partite GHZ states is simple. Entanglement generation is attempted in parallel across all k links. If successful they result in bipartite Bell states that are stored in quantum memories. The switch waits until n Bell pairs have been generated until performing an n-partite GHZ measurement, which converts the pairs into a state locally equivalent to a GHZ state. An additional constraint is that the switch has a finite buffer B of number of memories dedicated for each user (see Figure 8.3). If the number of pairs stored in a link is B and a new pair is generated, the old one is dropped and the new one is stored.

The protocol can be translated to a Markov chain. The state space is represented by a k-length vector where each entry is associated with a link and its value denotes the number of stored links. The switch's mean capacity, i.e. the number of states produced per second, can be derived from the steady-state of the Markov chain [52].

Using NetSquid, it is straightforward to fully reproduce the previous model and study the behaviour of the network without constructing the Markov Chain (details can be found in Supplementary Note 8.6.3). In Figure 8.4(a), we use NetSquid to study the capacity of a switch network serving nine users. The figure shows the capacity (number of produced GHZ-states per second), which we investigate for three use cases. First we consider a switch network distributing bipartite entanglement. Second, we consider also a switch-network serving bipartite entanglement but with link generation rates that do not satisfy the stability condition for the Markov Chain if the buffer B is infinitely large, i.e. a regime so far intractable. Third, we consider a switch-network distributing four-partite entanglement where the link generation rates μ differ per user, a regime not studied so far, and compute the capacity.

Beyond rate, it is important to understand the quality of the states produced. Answering this question with Markov chain models seems challenging. In order to analyse

entanglement quality, we introduce a more sophisticated decoherence model where the memories suffer from decay over time. In particular, we model decoherence as exponential T_2 noise, which impacts the quality of the state, as expressed in its fidelity with the ideal state. In Figure 8.4(b), we show the effect of the time-dependent memory noise on the average fidelity.

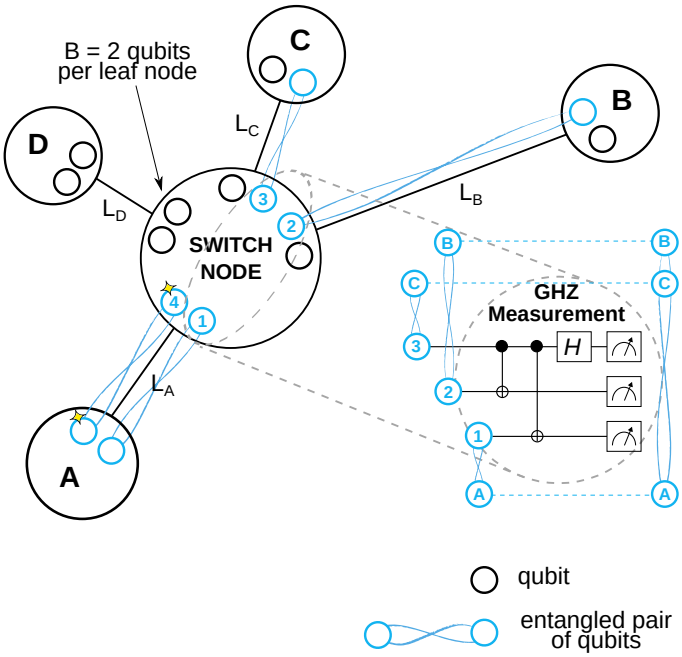


Figure 8.3: **A quantum switch in a star-shaped network topology as studied by Vardoyan et al.[52]**. The switch (central node) is connected to a set of users (leaf nodes) via an optical fibre link that distributes perfect Bell pairs at random times, following an exponential distribution with mean rate $\mu \propto e^{-\beta L}$, where L denotes the distance of the link and β the attenuation coefficient. Associated with each link there is a buffer that can store B qubits at each side of the link. As soon as n Bell pairs with different leaves are available, the switch performs a measurement in the n-partite Greenberger-Horne-Zeilinger (GHZ) basis, which results in an n-partite GHZ state shared by the leaves. The GHZ-basis measurement consists of: first, controlled-X gates with the same qubit as control; next, a Hadamard (H) gate on the control qubit; finally, measurement of all qubits individually. The figure shows 4 leaf nodes, GHZ size $n = 3$ and a buffer size $B = 2$.

8.2.3. SENSITIVITY ANALYSIS FOR THE PHYSICAL MODELLING OF A LONG RANGE REPEATER CHAIN

The next use case is the distribution of long-distance entanglement via a chain of quantum repeater nodes [1, 4] based on nitrogen-vacancy (NV) centres in diamond [66, 67]. This example consists of a more detailed physical model and more complicated control plane logic than the quantum switch or the distillation example presented at the start of this section. It is also an example of how NetSquid's modularity supports setting up simulations involving many nodes; in this case the node model and the protocol (which runs locally at a node) only need to be specified once, and can then be assigned to each

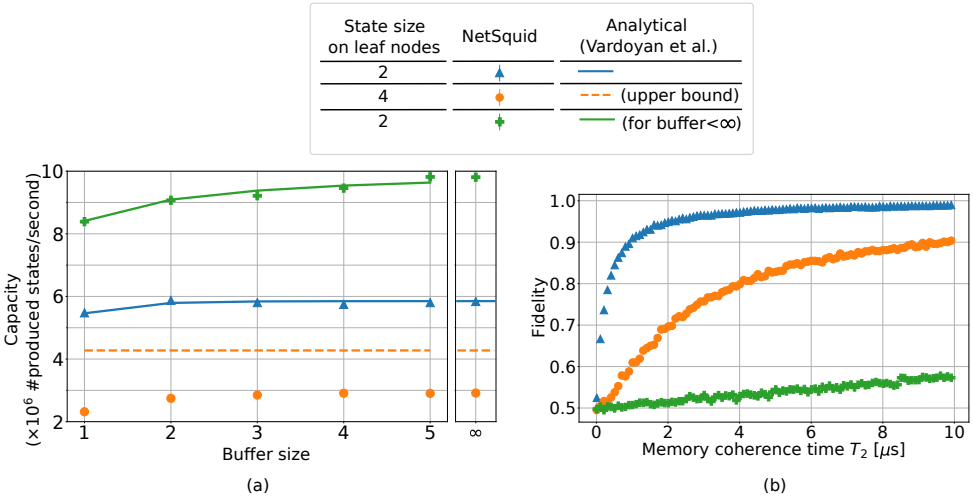


Figure 8.4: **Performance analysis of the quantum switch with 9 users using NetSquid. (a)** Capacity as a function of the buffer size (number of quantum memories that the switch has available per user) for either 2− or 4−qubit Greenberger-Horne-Zeilinger (GHZ)-states. For each scenario, the generation rate μ of pairs varies per user. For the blue scenario (2-partite entanglement, $\mu = [1.9, 1.9, 1.9, 1, 1, 1, 1, 1, 1]$ MHz), the capacity was determined analytically by Vardoyan et al. using Markov Chain methods [52, Figure 8]. Here we extend this to 4-partite entanglement (orange scenario, same μs), for which Vardoyan et al. have found an upper bound (by assuming unbounded buffer and each μ = maximum of original rates = 1.9 MHz) but no exact analytical expression. The green scenario ($\mu = [15, 1.9, 1.9, 1, 1, 1, 1, 1, 1]$ MHz) does not satisfy the stability condition for the Markov chain for unbounded buffer size (each leaf's rate < half of sum of all rates) so in that case steady-state capacity is not well-defined. We note that regardless of buffer size, the switch has a single link to each user, which is the reason why the capacity does not scale linearly with buffer size. **(b)** Average fidelity of the produced entanglement on the user nodes (no analytical results known) with unbounded buffer size. The fact that the green curve has lower fidelity than the blue one, while the former has higher rates, can be explained from the fact that the protocol prioritises entanglement which has the longest storage time (see Supplementary Note 8.6.3). Each data point represents the average of 40 runs (each 0.1 ms in simulation). Standard deviation is smaller than dot size.

node in the chain. Furthermore, the use of a discrete-event engine allows the actions of the individual protocols to be simulated asynchronously, in contrast to the typically sequential execution of quantum computing simulators.

The NV-based quantum processor includes the following three features. First, the nodes have a single communication qubit, i.e. a qubit acting as the optical interface that can be entangled with a remote qubit via photon interference. This seemingly small restriction has important consequences for the communications protocol. In particular, entanglement can not proceed in parallel with both adjacent nodes. As a consequence, operations need to be scheduled in sequence and the state of the communication qubit transferred onto a storage qubit. Second, the qubits in a node are connected with a star topology with the communication qubit located in the centre. Two-qubit gates are only possible between the communication qubit and a storage qubit. Third, communication and storage qubits have unequal coherence times. Furthermore, the storage qubits suffer additional decoherence when the node attempts to generate entanglement. Previous repeater-chain analyses, e.g. [17, 18, 38], did not take all three into account simultane-

ously.

Together with the node model, we consider two protocols: SWAP-ASAP and NESTED-WITH-DISTILL. In SWAP-ASAP, as soon as adjacent links are generated the entanglement is swapped. NESTED-WITH-DISTILL is a nested protocol [4] with entanglement distillation at every nesting level. For a description of the simulation, including the node model and protocols, see Methods, section 8.3.2.

The first question that we investigate is the distance that can be covered by a repeater chain. For this we choose two sets of hardware parameters that we dub near-term and 10× improved (see Supplementary Note 8.6.4) and choose two configurations: one without intermediate repeaters and one with three of them. We observe, see Figure 8.5(a), that the repeater chain performs worse in fidelity than the repeaterless configuration with near-term hardware. For improved hardware, we see two regimes, for short distances the use of repeaters increases rate but lowers fidelity while from 750 km until 1500 km the repeater chain outperforms the no-repeater setup.

The second question that we address is which protocol performs best for a given distance. We consider seven protocols: no repeater, and repeater chains implementing SWAP-ASAP or NESTED-WITH-DISTILL over 1, 3 or 7 repeaters. The latter is motivated by the fact that the NESTED-WITH-DISTILL protocol is defined for $2^n - 1$ repeaters ($n \geq 1$), and thus 1, 3, and 7 are the first three possible configurations. In Figure 8.5(b), we sweep over the hardware parameter space for two distances, where we improve all hardware parameters simultaneously and the improvement is quantified by a number we refer to as "improvement factor" (see section 8.3.2 of the Methods). For 500 km, we observe that the no-repeater configuration achieves larger or equal fidelity for the entire range studied. However, repeater schemes boost the rate for all parameter values. If we increase the distance to 800 km, then we see that the use of repeaters increases both rate and fidelity for the same range of parameters. If we focus on the repeater scheme, we observe for both distances that for high hardware quality, the NESTED-WITH-DISTILL scheme, which includes distillation, is optimal. In contrast, for lower hardware quality, the best-performing scheme that achieves fidelities larger than the classical bound 0.5 is the SWAP-ASAP protocol.

We note that beyond 700 km the entanglement rate decreases when the hardware is improved. This is due to the presence of dark counts, i.e. false signals that a photon has been detected. At large distances most photons dissipate in the fibre, whereby the majority of detector clicks are dark counts. Because a dark count is mistakenly counted as a successful entanglement generation attempt, improving (i.e. decreasing) the dark count rate in fact results in a lower number of observed detector clicks, from which the (perceived) entanglement rate plotted in Figure 8.5(a) is calculated.

Lastly, in Figure 8.6, we investigate the sensitivity of the entanglement fidelity for the different hardware parameters. We take as the figure of merit the best fidelity achieved with a SWAP-ASAP protocol. The uniform improvement factor is set to 3, while the following four hardware parameters are varied: a two-qubit gate noise parameter, photon detection probability (excluding transmission), induced storage qubit noise and visibility. We observe that improving the detection probability yields the largest fidelity increase from 2× to 50× improvement, while this increase is smallest for visibility. We also see that improving two-qubit gate noise or induced storage qubit noise on top of

an increase in detection probability yields only a small additional fidelity improvement, which however boosts fidelity beyond the classical threshold of 0.5. These observations indicate that detection probability is the most important parameter for realising remote-entanglement generation with the SWAP-ASAP scheme, followed by two-qubit gate noise and induced storage qubit noise.

8.2.4. FAST AND SCALABLE QUANTUM NETWORK SIMULATION

NetSquid has been designed and optimised to meet several key performance criteria: to be capable of accurate physical modelling, to be scalable to large networks, and to be sufficiently fast to support multi-variate design analyses with adequate statistics. While it is not always possible to jointly satisfy all the criteria for all use cases, NetSquid's design allows the user to prioritise them. We proceed to benchmark NetSquid to demonstrate its capabilities and unique strengths for quantum network simulation.

BENCHMARKING OF QUANTUM COMPUTATION

To accurately model physical non-idealities, it is necessary to choose a representation for quantum states that allows a characterisation of general processes such as amplitude damping, general measurements, or arbitrary rotations. NetSquid provides two representations, or "formalisms", that are capable of universal quantum computation: ket state vectors (KET) and density matrices (DM), both stored using dense arrays. The resource requirements for storage in memory and the computation time associated with applying quantum operations both scale exponentially with the number of qubits. While the density matrix scales less favourably, 2^{2n} versus 2^n for n qubits, its ability to represent mixed states makes it more versatile for specific applications. Given the exponential scaling, these formalisms are most suitable for simulations in which a typical qubit lifetime involves only a limited number of (entangling) interactions.

When scaling to large network simulations it can happen that hundreds of qubits share the same entangled quantum state. For such use cases, we need a quantum state representation that scales sub-exponentially in time and space. NetSquid provides two such representations based on the stabiliser state formalism: "stabiliser tableaus" (STAB) and "graph states with local Cliffords" (GSLC) [68, 69] that the user can select. Stabiliser states are a subset of quantum states that are closed under the application of Clifford unitaries and single-qubit measurement in the computational basis. In the context of simulations for quantum networks stabiliser states are particularly interesting because many network protocols consist of only Clifford operations and noise can be well approximated by stochastic application of Pauli gates. For a theoretical comparison of the STAB and GSLC formalisms see Supplementary Note 8.6.1.

The repetitive nature of simulation runs due to the collection of statistics via random sampling allows NetSquid to take advantage of "memoization" for expensive quantum operations, which is a form of caching that stores the outcome of expensive operations and returns them when the same input combinations reoccur to save computation time. Specifically, the action of a quantum operator onto a quantum state for a specific set of qubit indices and other discrete parameters can be efficiently stored, for instance as a sparse matrix. Future matching operator actions can then be reduced to a fast lookup and application, avoiding several expensive computational steps – see the Methods, sec-

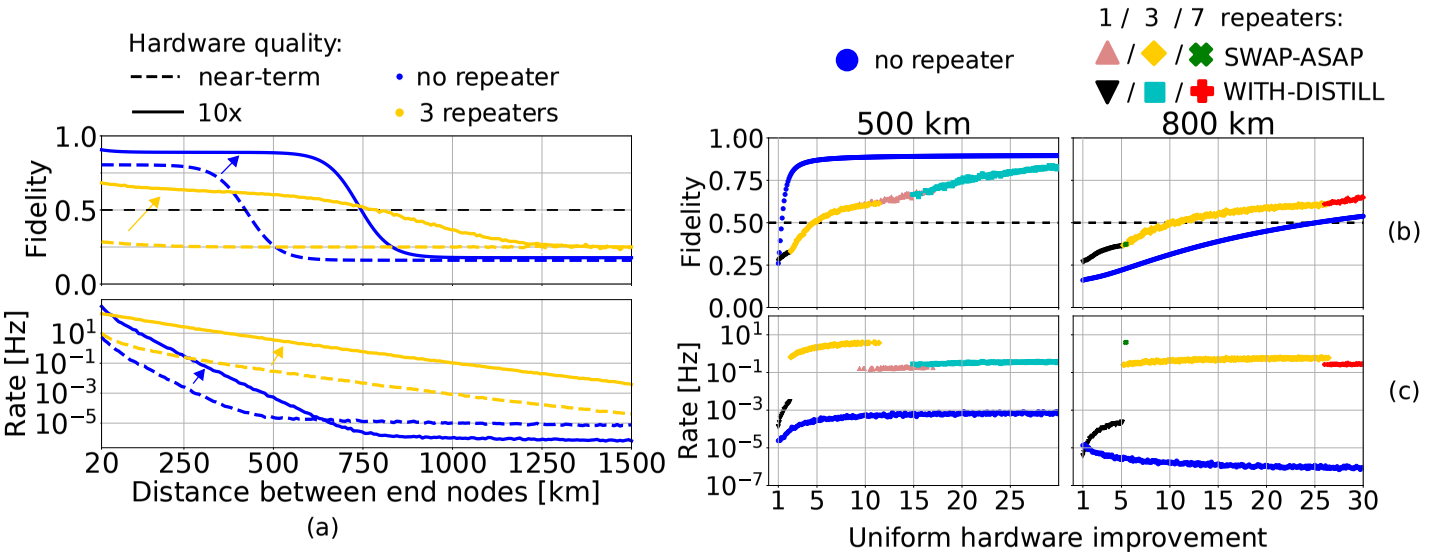


Figure 8.5: **Performance of repeaters based on nitrogen-vacancy (NV) centres in diamond. (a)** Fidelity and entanglement distribution rate achieved with near-term and 10× improved hardware (Supplementary Note 8.6.4) with the SWAP-ASAP protocol. Dashed line represents classical fidelity threshold of 0.5. We observe that for near-term hardware, the use of 3 repeaters yields worse performance in terms of fidelity than the no-repeater setup. For improved hardware we observe (i) that for approx. 0 - 750 kms, repeaters improve upon rate by orders of magnitude while still producing entanglement (fidelity> 0.5), while (ii) for approx. 750 - 1500 kms, repeaters outperform in both rate and fidelity. **(b-c)** Fidelity and rate achieved without and with repeaters (1, 3 or 7 repeaters) as function of a hardware improvement factor (Methods, section 8.3.2) for two typical distances from both distance regime (i) and (ii), for two protocols SWAP-ASAP and NESTED-WITH-DISTILL. For the repeater case, only the best-performing number-of-repeaters & protocol in terms of achieved fidelity is shown in (b), accompanied by its rate in (c). Each data point represents the average over (a) 200 and (b) 100 runs. Standard deviation is smaller than dot size.

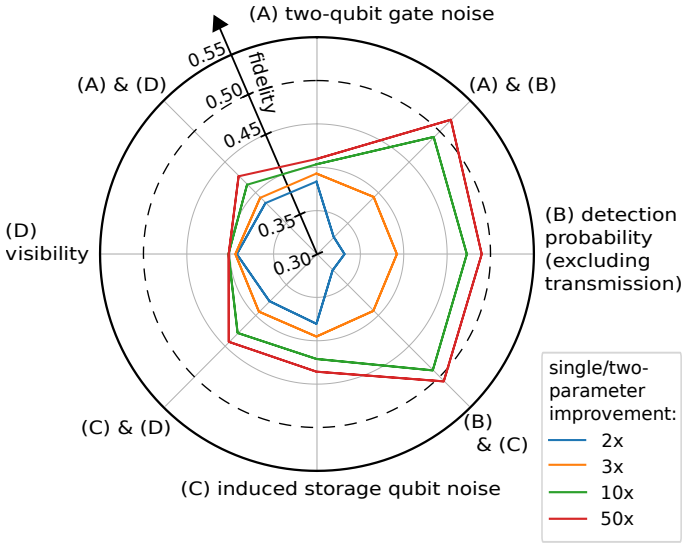


Figure 8.6: **Sensitivity of fidelity in various hardware parameters for nitrogen-vacancy (NV) repeater chains.** The NV hardware model consists of ~15 parameters and from those we focus on four parameters in this figure: (A) two-qubit gate fidelity, (B) detection probability, (C) induced storage qubit noise and (D) visibility. We start by improving all ~15 parameters, including the four designated ones, using an improvement factor of 3 (Methods, section 8.3.2). Then, for each of the four parameters only, we individually decrease their improvement factor to 2, or increase it to 10 or 50. The figure shows the resulting fidelity (horizontal and vertical grid lines; dashed line indicates maximal fidelity which can be attained classically). Note that at an improvement factor of 3 (orange line), all ~15 parameters are improved by 3 times, resulting in a fidelity of 0.39. In addition, we vary the improvement factor for combinations of two of the four parameters (diagonal lines). The 3× improved parameter values can be found in Supplementary Table 8.2. The other values (at 2/10/50×) are approximately: two-qubit gate fidelity F_{EC} (0.985/0.997/0.9994), detection probability $p_{\text{det}}^{\text{nofibre}}$ (6.8%/58%/90%), induced storage qubit noise $N_{1/e}$ (2800/14000/70000), visibility V (95%/99%/99.8%). The fidelities shown are obtained by simulation of the SWAP-ASAP protocol (3 repeaters) with a total spanned distance of 500 km. Each data point represents the average of 1000 runs (standard deviation on fidelity < 0.002).

tion 8.3.1 for more details.

In the following we benchmark the performance of the available quantum state formalisms. For this, we first consider the generation of an n qubit entangled GHZ state followed by a measurement of each qubit (see section 8.3.1 of the Methods). For a baseline comparison with classical quantum computing simulators we also include the ProjectQ [70] package for Python, which uses a quantum state representation equivalent to our ket vector. We show the average computation time for a single run versus the number of qubits for the different quantum computation libraries in Figure 8.7(a). The exponential scaling of the universal formalisms in contrast to the stabiliser formalisms is clearly visible, with the density matrix formalism performing noticeably worse. For the ket formalism we also show the effect of memoization, which gives a speed-up roughly between two and five.

Let us next consider a more involved benchmarking use case: the quantum computation involved in simulating a repeater chain i.e. only the manipulation of qubits, postponing all other simulation aspects, such as event processing and component mod-

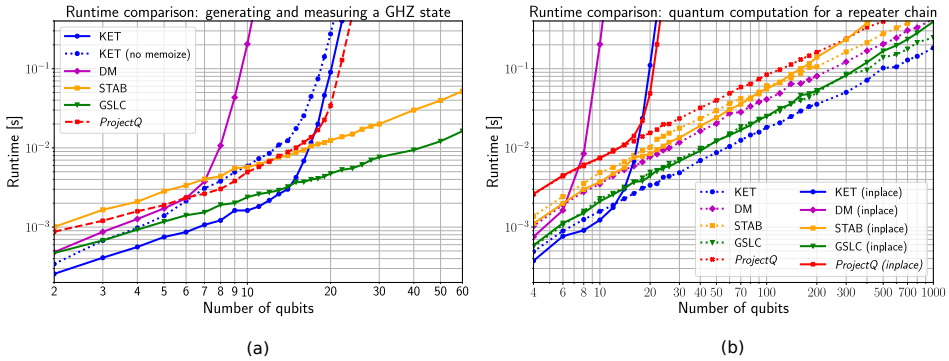


Figure 8.7: **Runtime comparison of NetSquid's quantum state formalisms.** Runtime comparisons of the available quantum state formalisms in NetSquid as well as ProjectQ ket vector for two benchmark use cases. The KET, DM, STAB and GSLC formalisms refer to the use of ket vectors, density matrices, stabiliser tableaus and graph states with local Cliffords, respectively. **(a)** Generating a Greenberger-Horne-Zeilinger (GHZ) state. Qubits are *split* off from the shared quantum state after a measurement. For the KET formalism the effect of turning off memoization (dotted line) is also shown. **(b)** Quantum computation involved in a repeater chain. Each formalism is shown with qubits split (dotted lines) versus being kept *in-place* (solid lines) after measurement.

elling, to the next section. This benchmark involves the following steps: first the $N-1$ pairs of qubits along an N node repeater chain are entangled, then each qubit experiences depolarising noise, and finally adjacent qubits on all but the end-nodes do an entanglement swap via a Bell state measurement (BSM). If the measured qubits are split from their shared quantum states after the BSM, then the size of any state is limited to four qubits.

The average computation time for a single run versus the number of qubits in the chain are shown for the different quantum computation libraries in Figure 8.7(b), where we have again included ProjectQ. We observe that for the NetSquid formalisms (but not for ProjectQ) keeping qubits "in-place" after each measurement is more performant than "splitting" them below a certain threshold due to the extra overhead of doing the latter. The ket vector formalism is seen to be the most efficient for this benchmarking use case if states are split after measurement. When the measurement operations are performed in-place the GSLC formalism performs the best beyond 15 qubits.

BENCHMARKING OF EVENT-DRIVEN SIMULATIONS

As explained in the results section, a typical NetSquid simulation involves repeatedly sampling many independent runs. As such NetSquid is "embarrassingly parallelisable": the reduction in runtime scales linearly with the number of processing cores available, assuming there is sufficient memory available. Nonetheless, given the computational requirements associated with collecting sufficient statistics and analysing large parameter spaces it remains crucial to optimise the runtime performance per core.

Depending on the size of the network, the detail of the physical modelling, and the duration of the protocols under consideration, the number of events processed for a single simulation run can range anywhere from a few thousand to millions. To efficiently

process the dynamic scheduling and handling of events NetSquid uses the discrete-event simulation engine PyDynAA [71] (see section 8.3.1 of the Methods). NetSquid aims to schedule events as economically as possible, for instance by streamlining the flow of signals and messages between components using inter-connecting ports.

To benchmark the performance of an event-driven simulation run in NetSquid we consider a simple network that extends the single repeater (without distillation) shown in Figure 8.1 into an N node chain – see Supplementary Note 8.6.2 for further details on the simulation setup. For the quantum computation we will use the ket vector formalism based on the benchmarking results from the previous section, and split qubits from their quantum states after measurement to avoid an exponential scaling with the number of nodes. In Figure 8.8 we show the average computation time for deterministically generating end-to-end entanglement versus the number of nodes in the chain. Also shown is a relative breakdown in terms of the time spent in the NetSquid sub-packages involved, as well as the PyDynAA and NumPy packages. We observe that the biggest contribution to the simulation runtime is the components sub-package, which accounts for 30% of the total at 1000 nodes. The relative time spent in each of the NetSquid sub-packages, as well as NumPy and PyDynAA, is seen to remain constant with the number of nodes. The total runtime of each of the NetSquid sub-packages is the sum of many small contributions, with the costliest function for the components sub-package for a 1000 node chain, for example, contributing only 7% to the total.

Extending this benchmark simulation with more detailed physical modelling may shift the relative runtime distribution and impact the overall performance. For example, more time may be spent in calls to the "components" and "components.models" sub-packages, additional complexity can increase the volume of events processed by the "pydynaa" engine, and extra quantum characteristics can lead to larger quantum states. In case of the latter, however, the effective splitting of quantum states can still allow such networks to scale if independence among physical elements can be preserved.

8.2.5. COMPARISON WITH OTHER QUANTUM NETWORK SIMULATORS

Let us compare NetSquid to other existing quantum network simulators. First, SimulaQron [72] and QuNetSim [73] are two simulators that do not aim at realistic physical models of channels and devices, or timing control. Instead, SimulaQron's main purpose is application development. It is meant to be run in a distributed fashion on physically-distinct classical computers. QuNetSim focuses on simplifying the development and implementation of quantum network protocols.

In contrast with SimulaQron and QuNetSim, the simulator SQUANCH [74] allows for quantum network simulation with configurable error models at the physical layer. However, SQUANCH, similar to SimulaQron and QuNetSim, does not use a simulation engine that can accurately track time. Accurate tracking is crucial for e.g. studying time-dependent noise such as memory decoherence.

Other than NetSquid, there now exist three discrete-event quantum simulators: the QuISP [75], qkdX [76] and SeQUeNCe [77] simulators. With these simulators it is possible to accurately characterise complex timing behaviour, however they differ in goals and scope. Similarly to NetSquid, QuISP aims to support the investigation of large networks that consist of too many entangled qubits for full quantum-state tracking. In contrast to

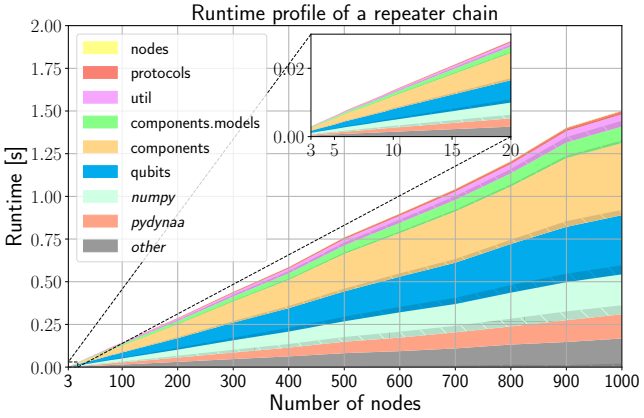


Figure 8.8: **Runtime profile of a repeater chain simulation using Netsquid.** Runtime profile for a repeater chain simulation with a varying number of nodes in the chain. The maximum quantum state size is four qubits. The total time spent in the functions of each NetSquid subpackage and its main package dependencies (in italics) is shown. The dark hatched bands show the largest contribution from a single function in each NetSquid sub-package, as well as in NumPy and uncategorised (*other*) functions. The sub-packages are stacked in the same order as they are listed in the legend.

NetSquid, which achieves this by managing the size of the state space, and providing the stabiliser representation as one of its quantum state formalisms, QuISP's approach is to track an error model of the qubits in a network instead of their quantum state. qkdX, on the other hand, captures the physics more closely through models of the quantum devices but is restricted to the simulation of quantum key distribution protocols. Lastly, SeQUeNCe, similar to NetSquid, aims at simulation at the level of hardware, control plane or application. It has a fixed control layer consisting of reprogrammable modules. In contrast, NetSquid's modularity is not tied to a particular network stack design. Furthermore, it is unclear to us how performant SeQUeNCe's quantum simulation engine is: currently, at most a 9-node network has been simulated, whereas NetSquid's flexibility to choose a quantum state representation enables scalability to simulation of networks of up to 1000 nodes.

8.2.6. CONCLUSIONS

In this chapter we have presented our design of a modular software framework for simulating scalable quantum networks and accurately modelling the non-idealities of real world physical hardware, providing us with a design tool for future quantum networks. We have showcased its power and also its limitations via example use cases. Let us recap NetSquid's main features.

First, NetSquid allows the modelling of any physical device in the network that can be mapped to qubits. To demonstrate this we studied a quantum repeater chain based on nitrogen-vacancy centres in diamond.

Second, NetSquid is entirely modular, allowing users to set up large scale simulations of complicated networks and to explore variations in the network design; for example,

by comparing how different hardware platforms perform in an otherwise identical network layout. Moreover, this modularity makes it possible to explore different control plane protocols for quantum networks in a way that is essentially identical to how such protocols would be executed in the real world. Control programs can be run on any simulated network node, exchanging classical and quantum communication with other nodes as dictated by the protocol. That allows users to investigate the intricate interplay between control plane protocols and the physical devices dictating the performance of the combined quantum network system. As an example, we studied the control plane of a quantum network switch. NetSquid has also already found use in exploring the interplay between the control plane and the physical layer in [34, 78, 79].

Finally, to allow large scale simulations, the quantum computation library used by NetSquid has been designed to manage the dynamic lifetimes of many qubits across a network. It offers a seamless choice of quantum state representations to support different modelling use cases, allowing both a fully detailed simulation in terms of wave functions or density matrices, or simplified ones using certain stabiliser formalisms. As an example use case, we explored the simulation run-time of a repeater chain with up to one thousand nodes.

In light of the results we have presented, we see a clear application for NetSquid in the broad context of communication networks. It can be used to predict performance with accurate models, to study the stability of large networks, to validate protocol designs, to guide experiment, etc. While we have only touched upon it in our discussion of performance benchmarks, NetSquid would also lend itself well to the study of modular quantum computing architectures, where the timing of control plays a crucial role in studying their scalability. For instance, it might be used to validate the microarchitecture of distributed quantum computers or more generally to simulate different components in modular architectures.

8.3. METHODS

8.3.1. DESIGN AND FUNCTIONALITY OF NETSQUID

The NetSquid simulator is available as a software package for the Python 3 programming language. It consists of the sub-packages “qubits”, “components”, “models”, “nodes”, “protocols” and “util”, which are shown stacked in Figure 8.9. NetSquid depends on the PyDynAA software library to provide its discrete-event simulation engine [71]. Under the hood speed critical routines and classes are written in Cython [80] to give C-like performance, including its interfaces to both PyDynAA and the scientific computation packages NumPy and SciPy. In the following subsections we highlight some of the main design features and functionality of NetSquid; for a more detailed presentation see Supplementary Note 8.6.1.

DISCRETE EVENT SIMULATION

The PyDynAA package provides a fast, powerful, and lightweight discrete-event simulation engine. It is a C++ port of the core engine layer from the DynAA simulation framework [71], with bindings added for the Python and Cython languages. DynAA defines a concise set of classes and concepts for modelling event-driven simulations. The simulation engine manages a timeline of “events”, which can only be manipulated by objects

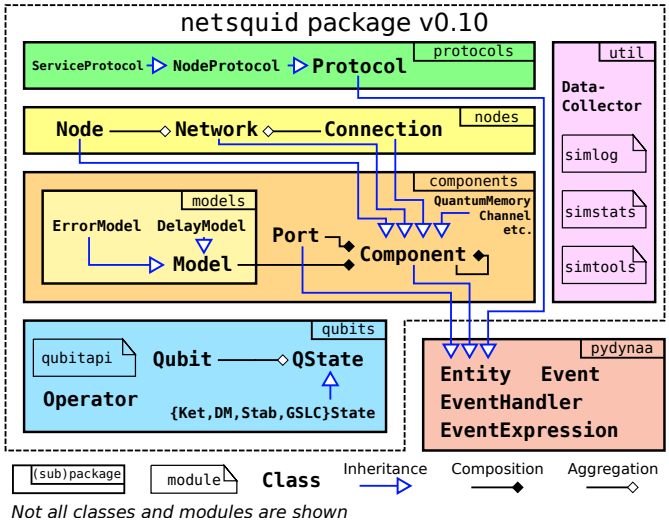


Figure 8.9: **Overview of NetSquid's software architecture.** The sub-packages that make up the NetSquid package are shown stacked in relation to each other and the PyDynAA package dependency. The main classes in each (sub-)package are highlighted, and their relationships in terms of inheritance, composition and aggregation are shown. Also shown are the key modules users interact with, which are described in the main text. In this chapter NetSquid version 0.10 is described.

that are sub-classes of the "entity" base class. Simulation entities can dynamically schedule events on the timeline and react to events by registering an "event handler" object to wait for event(s) with a specified type, source entity, or identifier to be triggered.

To deal with the timing complexities encountered in NetSquid simulations, an "event expression" class was introduced to PyDynAA to allow entities to also wait on logical combinations of events to occur. Atomic event expressions, which describe regular wait conditions for standard events, can be combined to form composite expressions using logical "and" and "or" operators to any depth. This feature has been used extensively in NetSquid to model both the internal behaviour of hardware components, as well as for programming network protocols.

QUBITS AND QUANTUM COMPUTATION

The qubits sub-package of NetSquid defines the "qubit" object that is used to track the flow of quantum information. Qubits internally share quantum state ("QState") objects, which grow and shrink in size as qubits interact or are measured. The "QState" class is an interface that is implemented by a range of different formalisms, as presented in section 8.2.4 of the Results and Discussion. Via the qubit-centric API, which provides functions to directly manipulate qubits without knowledge of their shared quantum states, users can program simulations in a formalism agnostic way. Functionality is also provided to automatically convert between quantum states that use different formalisms, and to sample from a distribution of states, which is useful for instance for pure state formalisms.

The ket and density matrix formalisms use dense arrays (vectors or matrices, respectively) to represent quantum states. Applying a k qubit operator to an n qubit ket vector state generally involves the computationally expensive task of performing 2^{n-k} matrix multiplications on 2^k temporary sub-vectors and aggregating the result (only in special cases can this be done in-place) [81, 82]. The analogous application of an operator to a density matrix is more expensive due to the extra dimension involved. However, as discussed in section 8.2.4 of the Results and Discussion, the repetitive nature of NetSquid simulations allows us to take advantage of operators frequently being applied to the same qubit indices for states of a given size. For these operators, we compute a $2^n \times 2^n$ dimensional sparse matrix representation of the k qubit operator via tensor products with the identity and memoize this result for the specific indices and size. When the memoization is applicable the computational cost of applying a quantum operator can then be reduced to just sparse matrix multiplication onto a dense vector or matrix. Memoization is similarly applicable to general Clifford operators in the stabiliser tableau formalism. To use memoization on operators that depend on a continuous parameter, such as arbitrary rotations, the parameter can be discretised i.e. rounded to some limited precision.

PHYSICAL MODELLING OF NETWORK COMPONENTS

All physical devices in a quantum network are modelled by a "component" object, and are thereby also all simulation entities, as shown in Figure 8.9. Components can be composed of subcomponents, which makes setting up networks in NetSquid modular. The network itself, for instance, can be modelled as a composite component containing "node" and "connection" components; these composite components can in turn contain components such as quantum memories, quantum and classical channels, quantum sources, etc., as illustrated in Figure 8.1. The physical behaviour of a component is described by composing it of "models", which can specify physical characteristics such as transmission delays or noise such as photon loss or decoherence. Communication between components is facilitated by their "ports", which can be connected together to automatically pass on messages.

NetSquid also allows precise modelling of quantum computation capable devices. For this it provides the "quantum processor" component, a subclass of the quantum memory. This component is capable of executing "quantum programs" i.e. sequences of "instructions" that describe operations such as quantum gates and measurements or physical processes such as photon emission. Quantum programs fully support conditional and iterative statements, as well as parallelisation if the modelled device supports it. When a program is executed its instructions are mapped to the physical instructions on the processor, which model the physical duration and errors associated to carrying out the operation. A physical instruction can be assigned to all memory positions or only to a specific position, as well as directionally between specific memory positions in the case of multi-qubit instructions.

ASYNCHRONOUS FRAMEWORK FOR PROGRAMMING PROTOCOLS

NetSquid provides a "protocol" class to describe the network protocols and classical control plane logic running on a quantum network. Similarly to the component class, a protocol is a simulation entity and can thereby directly interact with the event timeline.

Protocols can be nested inside other protocols and may describe both local or remote behaviour across a network. The "node protocol" subclass is specifically restricted to only operating locally on a single node. Inter-protocol communication is possible via a signalling mechanism and a request and response interface defined by the "service protocol" class. Protocols can be programmed using both the standard callback functionality of PyDynAA and a tailored asynchronous framework that allows the suspension of a routine conditioned on an "event expression"; for example, to wait for input to arrive on a port, a quantum program to finish, or to pause for a fixed duration.

The "util" sub-package shown in Figure 8.9 provides a range of utilities for running, recording and interacting with simulations. Functions to control the simulation are defined in the "simtools" module, including functions for inspecting and diagnosing the timeline. A "data collector" class supports the event-driven collection of data during a simulation, which has priority over other event handlers to react to events. The "simstats" module is responsible for collecting a range of statistics during a simulation run, such as the number of events and callbacks processed, the maximum and average size of manipulated quantum states, and a count of all the quantum operations performed. Finally, the "simlog" module allows fine grained logging of the various modules for debugging purposes.

BENCHMARKING

To perform the benchmarking described in section 8.2.4 of the Results and Discussion we used computing nodes with two 2.6 GHz Intel Xeon E5-2690 v3 (Haswell) 12 core processors and 64 GB of memory. Because each process only requires a single core, care was taken to ensure sufficient cores and memory were available when running jobs in parallel. The computation time of a process is the arithmetic average of a number of successive iterations; to avoid fluctuations due to interfering CPU processes the reported time is a minimum of five such repeated averages. To perform the simulation profiling the Cython extension modules of both NetSquid and PyDynAA were compiled with profiling on, which adds some runtime overhead. Version 0.10.0 and 0.3.5 of NetSquid and PyDynAA were benchmarked. We benchmarked against ProjectQ version 0.4.2 using its "MainEngine" backend. See Supplementary Note 8.6.2 for further details.

Using the same machine, simulations for Figure 8.5(b-c) were run, which took almost 260 core hours wallclock time in total. For Figure 8.4 (≈10 hours in total), Figure 8.5(a) (≈90 minutes) and Figure 8.6 (≈30 minutes), a single core Intel Xeon Gold 6230 processor (3.9GHz) with 192 GB RAM was used.

8.3.2. IMPLEMENTING A PROCESSING-NODE REPEATER CHAIN IN NETSQUID

Here, we explain the details of the most complex of our two use cases, namely the repeater chain of Nitrogen-Vacancy-based processing nodes from section 8.2.3 of the Results and Discussion (see Supplementary Note 8.6.3 for details on the quantum switch simulations). We first describe how we modelled the NV hardware, followed by the repeater protocols used. With regard to the physical modelling, let us emphasise that this is well established (see e.g. [83]); the main goal here is to explain how we used this model in a NetSquid implementation.

In our simulations the following NetSquid components model the physical repeater chain: "nodes", each holding a single "quantum processor" modelling the NV centre, and "classical channels" that connect adjacent nodes and are modelled as fibres with a constant transmission time. We choose equal spacing between the nodes. If we were to simulate individual attempts at entanglement generation, we would also need components for transmitting and detecting qubits such as was used in previous NetSquid simulations of NV centres [34]. However, in order to speed up simulations we insert the entangled state between remote NVs using a model. We designed two types of protocols to run on each node of this network that differ in whether they implement a scheme with or without distillation.

In the remainder of this section, we describe the components modelling. More detailed descriptions of the hardware parameters and their values used in our simulation can be found in Supplementary Note 8.6.4.

MODELLING A NITROGEN-VACANCY CENTRE IN DIAMOND

In NetSquid, the NV centre is modelled by a quantum processor component, which holds a single communication qubit (electronic spin-1 system) and multiple storage qubits (^{13}C nuclear spins). The decay of the state held by a communication qubit or storage qubit is implemented using a noise model, which is based on the relaxation time T_1 and the dephasing time T_2. If a spin is acted upon after having been idle for time Δt, then to its state ρ we first apply a quantum channel

$$\rho \mapsto E_0 \rho E_0^\dagger + E_1 \rho E_1^\dagger$$

where

$$E_0 = |0\rangle\langle 0| + \sqrt{1-p}\,|1\rangle\langle 1|, E_1 = \sqrt{p}\,|0\rangle\langle 1|$$

and $p = 1 - e^{-\Delta t/T_1}$. Subsequently, we apply a dephasing channel

$$\mathcal{N}_p^{\text{deph}} : \rho \mapsto (1-p)\rho + pZ\rho Z \tag{8.1}$$

where $Z = |0\rangle\langle 0| - |1\rangle\langle 1|$ and the dephasing probability equals

$$p = \frac{1}{2}\left(1 - e^{-\Delta t/T_2} \cdot e^{\Delta t/(2T_1)}\right).$$

The electron and nuclear spins have different T_1 and T_2 times.

We allow the quantum processor to perform the following operations on the electron spin: initialisation (setting the state to $|0\rangle$), readout (measurement in the $\{|0\rangle, |1\rangle\}$ basis) and arbitrary single-qubit rotation. In particular, the latter includes Pauli rotations

$$R_P(\theta) = \cos(\theta/2)\mathbb{1}_2 - i\sin(\theta/2)P \tag{8.2}$$

where θ is the rotation angle, $P \in \{X, Y, Z\}$ and $\mathbb{1}_2 = |0\rangle\langle 0| + |1\rangle\langle 1|$, $X = |0\rangle\langle 1| + |1\rangle\langle 0|$, $Y = -i|0\rangle\langle 1| + i|1\rangle\langle 0|$ and $Z = |0\rangle\langle 0| - |1\rangle\langle 1|$ are the single-qubit Pauli operators.

For the nuclear spin, we have only initialisation and rotations $R_Z(\theta)$ for arbitrary rotation angle θ. In addition, we allow the two-qubit controlled-$R_X(\pm\theta)$ gate between an electron (e) and a nuclear (n) spin:

$$|0\rangle\langle 0|_e \otimes R_X(\theta)_n + |1\rangle\langle 1|_e \otimes R_X(-\theta)_n.$$

We model each noisy operation O_{noisy} as the perfect operation O_{perfect} followed by a noise channel $\mathcal{N}$:

$$O_{\text{noisy}} = \mathcal{N} \circ O_{\text{perfect}}.$$

If O is a single-qubit rotation, then $\mathcal{N}$ is the depolarising channel:

$$\mathcal{N}_p^{\text{depol}} : \rho \mapsto \left(1 - \frac{3p}{4}\right)\rho + \frac{p}{4}\left(X\rho X + Y\rho Y + Z\rho Z\right) \tag{8.3}$$

with parameter $p = 4(1-F)/3$ with F the fidelity of the operation.

If O is single-qubit initialisation, $\mathcal{N} = \mathcal{N}_p^{\text{depol}}$ with parameter $p = 2(1-F)$. The noise map of the controlled-R_X gate is an identical single-qubit depolarising channel on both involved qubits, i.e. $\mathcal{N} = \mathcal{N}_p^{\text{depol}} \otimes \mathcal{N}_p^{\text{depol}}$.

Finally, we model electron spin readout by a POVM measurement with the Kraus operators

$$M_0 = \begin{pmatrix} \sqrt{f_0} & 0 \\ 0 & \sqrt{1-f_1} \end{pmatrix}, \quad M_1 = \begin{pmatrix} \sqrt{1-f_0} & 0 \\ 0 & \sqrt{f_1} \end{pmatrix} \tag{8.4}$$

where $1-f_0$ ($1-f_1$) is the probability that a measurement outcome 0 (1) is flipped to 1 (0).

SIMULATION SPEEDUP VIA STATE INSERTION

For generating entanglement between the electron spins of two remote NVs, we simulate a scheme based on single-photon detection, following its experimental implementation in [84]. NetSquid was used previously to simulate each generation attempt of this scheme, which includes the emission of a single photon by each NV, the transmission of the photons to the midpoint through a noisy and lossy channel, the application of imperfect measurement operators at the midpoint, and the transmission of the measurement outcome back to the two involved nodes [34]. For larger internode distances, simulating each attempt requires unfeasibly long simulation times due to the exponential decrease in attempt success rate. To speed up our simulations in the examples studied here, we generate the produced state between adjacent nodes from a model which has shown good agreement with experimental results [84]. This procedure includes a random duration and noise induced on the storage qubits, as we describe below.

Let us define

$$\begin{aligned}
p_{00} &= \alpha^2[2p_{\text{det}}(1-p_{\text{det}})(1-p_{\text{dc}}) \\
&\quad +2p_{\text{dc}}(1-p_{\text{dc}})(1-p_{\text{det}})^2 \\
&\quad +p_{\text{det}}^2(1-p_{\text{dc}})\cdot\frac{1}{2}(1+V)] \\
p_{10} &= \alpha(1-\alpha)\cdot[(1-p_{\text{dc}})\cdot p_{\text{det}} \\
&\quad +2p_{\text{dc}}(1-p_{\text{dc}})(1-p_{\text{det}})] \\
p_{01} &= p_{01} \\
p_{11} &= (1-\alpha)^2\cdot p_{\text{dc}}
\end{aligned}$$

where p_{det} is the detection probability, p_{dc} the dark count probability, V denotes photon indistinguishability and α is the bright-state parameter (see Supplementary Note 8.6.4

for parameter descriptions). We follow the model of the produced entangled state from the experimental work of [84], whose setup consists of a beam splitter with two detectors located between the two adjacent nodes. In their model, the unnormalised state is given by

$$\rho = \begin{pmatrix} p_{00} & 0 & 0 & 0 \\ 0 & p_{01} & \pm\sqrt{V p_{01} p_{10}} & 0 \\ 0 & \pm\sqrt{V p_{01} p_{10}} & p_{10} & 0 \\ 0 & 0 & 0 & p_{11} \end{pmatrix}$$

where $\pm$ denotes which of the two detectors detected a photon (each occurring with probability $\frac{1}{2}$). We also follow the model of [84] for double-excitation noise and optical phase uncertainty, by applying a dephasing channel to both qubits with parameter $p = p_{\text{dexc}}/2$, followed by a dephasing channel of one of the qubits, respectively.

The success probability of a single attempt is

$$p_{\text{succ}} = p_{00} + p_{01} + p_{10} + p_{11}.$$

The time elapsed until the fresh state is put on the electron spins is $(k-1)\cdot\Delta t$ with $\Delta t := (t_{\text{emission}} + L/c)$, where t_{emission} is the delay until the NV centre emits a photon, L the internode distance and c the speed of light in fibre. Here, k is the number of attempts up to and including successful entanglement generation and is computed by drawing a random sample from the geometric distribution $\Pr(k) = p_{\text{succ}} \cdot (1-p_{\text{succ}})^{k-1}$. After the successful generation, we wait for another time Δt to mimic the photon travel delay and midpoint heralding message delay.

Every entanglement generation attempt induces dephasing noise on the storage qubits in the same NV system. We apply the dephasing channel (eq. (8.1)) at the end of the successful entanglement generation, where the accumulated dephasing probability is

$$\frac{1-(1-2p_{\text{single}})^k}{2} \tag{8.5}$$

where p_{single} is the single-attempt dephasing probability (see eq. (8.13) in Supplementary Note 8.6.4).

HOW WE CHOOSE IMPROVED HARDWARE PARAMETERS

Here, we explain how we choose 'improved' hardware parameters. Let us emphasise that this choice is independent of the setup of our NetSquid simulations and only serves the purpose of showcasing that NetSquid can assess the performance of hardware with a given quality.

By 'near-term' hardware, we mean values for the above defined parameters as expected to be achieved in the near future by NV hardware. If we say that an error probability is improved by an improvement factor k, we mean that its corresponding no-error probability equals $\sqrt[k]{p_{\text{ne}}}$, where p_{ne} is the no-error probability of the near-term hardware. For example, visibility V is improved as $\sqrt[k]{V}$ while the probability of dephasing p of a gate is improved as $1-\sqrt[k]{1-p}$. A factor $k=1$ thus corresponds to 'near-term' hardware. By 'uniform hardware improvement by k', we mean that all hardware parameters

are improved by a factor k. We do not improve the duration of local operations or the fibre attenuation. The near-term parameter values as well as the individual improvement functions for each parameter can be found in Supplementary Note 8.6.4.

NV REPEATER CHAIN PROTOCOLS

For the NV repeater chain, we simulated two protocols: SWAP-ASAP and NESTED-WITH-DISTILL. Both protocols are composed of five building blocks: ENTGEN, STORE, RETRIEVE, DISTILL and SWAP. By ENTGEN, we denote the simulation of the entanglement generation protocol based on the description in the previous subsection: two nodes wait until a classical message signals that their respective electron spins hold an entangled pair. In reality, such functionality would be achieved by a link layer protocol [34]. STORE is the mapping of the electron spin state onto a free nuclear spin, and RETRIEVE is the reverse operation. The DISTILL block implements entanglement distillation between two remote NVs for probabilistically improving the quality of entanglement between two nuclear spins (one at each NV), at the cost of reading out entanglement between the two electron spins. It consists of local operations followed by classical communication to determine whether distillation succeeded. The entanglement swap (SWAP) converts two short-distance entangled qubit pairs $A-M$ and $M-B$ into a single long-distance one $A-B$, where A, B and M are nodes. It consists of local operations at M, including spin readout, and communicating the measurement outcomes to A and B, followed by A and B updating their knowledge of the precise state $A-B$ they hold in the perfect case. We opt for such tracking as opposed to applying a correction operator to bring $A-B$ back to a canonical state since the correction operator generally cannot be applied to the nuclear spins directly. Details of the tracking are given in Supplementary Note 8.6.6. The circuit implementations for the building blocks, "quantum programs" in NetSquid, are given in Supplementary Note 8.6.5.

Let us explain the SWAP-ASAP and NESTED-WITH-DISTILL protocols in spirit; the exact protocols run asynchronously on each node and can be found in Supplementary Note 8.6.5. In the SWAP-ASAP protocol, a repeater node performs ENTGEN with both its neighbours, followed by SWAP as soon as it holds the two entangled pairs. Next, NESTED-WITH-DISTILL is a nested protocol on 2^n+1 nodes (integer $n \geq 0$) with distillation at each nesting level which is based on the BDCZ protocol [4]. For nesting level $n = 0$, there are no repeaters and the two nodes only perform ENTGEN once. For nesting level $n > 0$, the chain is divided into a left part and a right part of $2^{n-1}+1$ nodes, and the middle node (included in both parts) in the chain generates twice an entangled pair with the left end node following the $(n-1)$-level protocol; STORE is applied in between to free the electron spin. Subsequently, DISTILL is performed with the two pairs as input (restart if distillation fails), after which the same procedure is performed on the right. Once the right part has finished, the middle node performs SWAP to connect the end nodes. If needed, STORE and RETRIEVE are applied prior to DISTILL and SWAP in order achieve the desired configuration of qubits in the quantum processor, e.g. for DISTILL to ensure that the two involved NVs hold an electron-electron and nuclear-nuclear pair of qubits, instead of electron-nuclear for both entangled pairs.

8.4. DATA AVAILABILITY

The data presented in this chapter have been made available at https://doi.org/10.34894/URV169 [85].

8.5. CODE AVAILABILITY

The NetSquid-based simulation code that was used for the simulations in this chapter has been made available at https://doi.org/10.34894/DU3FTS [86].

8.6. APPENDIX

8.6.1. ANATOMY OF THE NETSQUID SIMULATOR

This section supplements the Methods, section 8.3.1, by going into more depth on specific details of NetSquid's design. The version of NetSquid that we consider is 0.10. For up-to-date documentation of the latest NetSquid version, including a detailed user tutorial, code examples, and its application programming interface, please visit the NetSquid website: https://netsquid.org [62].

QUBITS AND THEIR QUANTUM STATE FORMALISMS

The *qubits* sub-package of NetSquid, shown in Figure 8.9 (main text), provides a specialised quantum computation library for tracking the lifetimes of many qubits across a quantum network. A class diagram of the main classes present in this sub-package is shown in Supplementary Figure 8.10. Rather than assigning a single quantum state for a predefined number of qubits, both the number of qubits and the quantum states describing them are managed dynamically during a simulation run. Every *Qubit* (`Qubit`) object references a *shared quantum state* (`QState`) object, which varies in size according to the number of qubits sharing it. When two or more qubits interact, for instance via a multi-qubit operation, their respective shared quantum states are merged together. On the other hand, when a qubit is projectively measured or discarded it can be split from the quantum state it's sharing and optionally be assigned a new single-qubit state.

The `QState` class is an interface for shared quantum states that NetSquid implements for four different *quantum state formalisms* – described in more detail below. To allow simulations to seamlessly switch between formalisms NetSquid offers a formalism agnostic API, which is defined in the *qubitapi* module. The functions in this API take as their primary input parameters the qubits to manipulate and the *operators* (`Operator`) describing a quantum operation to perform, if applicable. The merging and splitting of shared quantum states is handled automatically under the hood, as are conversions between states using different formalisms (where this is possible). This allows users to program in a "qubit-centric" way, by for instance applying local operations to qubits at a network node without knowledge of their positions within a quantum state representation or any entanglement they may have across the network.

We proceed to give a high-level description of the available quantum state formalisms. The first two formalisms are ket state vectors (KET) and density matrices (DM), which both enable universal quantum computation. A ket state vector represents a quantum pure state, while a density matrix can represent statistical ensembles of pure

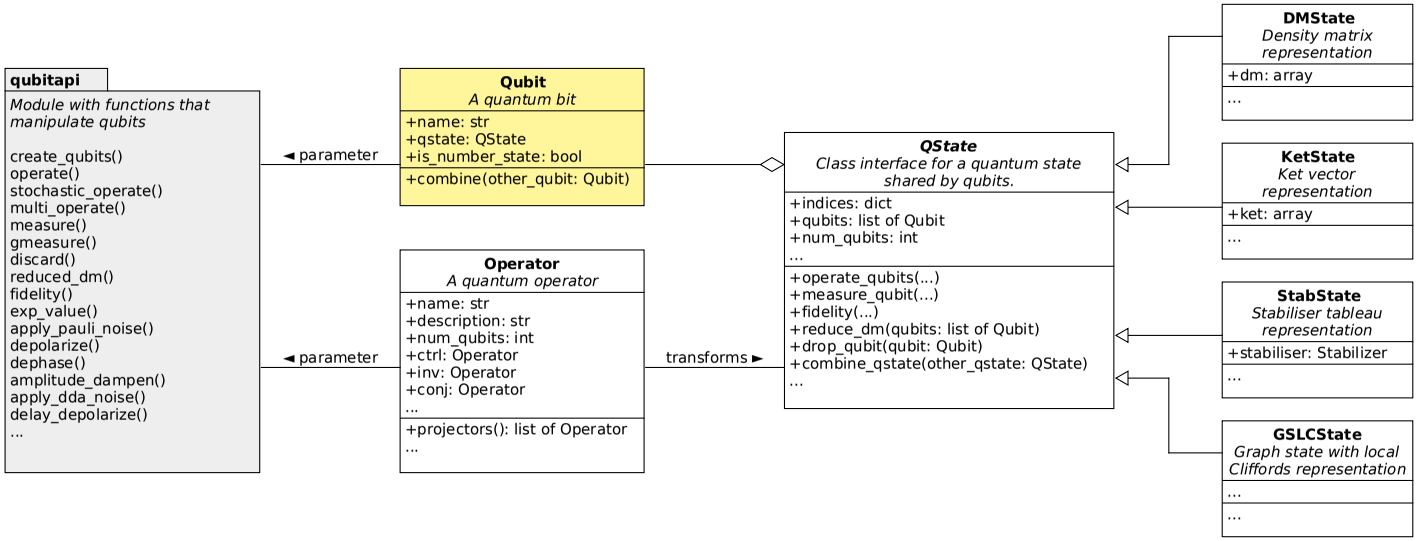


Figure 8.10: **Design overview of NetSquid's qubits sub-package.** The main classes and module of the *netsquid.qubits* sub-package. `Qubit` objects can be manipulated, as described for instance by `Operator` objects, using the functions of the *qubitapi* module. Under the hood the qubits share a specific sub-class of the `QState` interface. Ellipses indicate that not all of a class's public variables and methods are listed.

$$\begin{pmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{2}i \\ -\frac{1}{2}i \end{pmatrix}$$

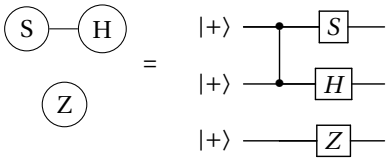
(a) *Ket vector*

$$\rho = \begin{pmatrix} \frac{1}{4} & -\frac{1}{4} & 0 & 0 & 0 & 0 & -\frac{1}{4}i & \frac{1}{4}i \\ -\frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 & \frac{1}{4}i & -\frac{1}{4}i \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{4}i & -\frac{1}{4}i & 0 & 0 & 0 & 0 & \frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4}i & \frac{1}{4}i & 0 & 0 & 0 & 0 & -\frac{1}{4} & \frac{1}{4} \end{pmatrix}$$

(b) *Density matrix*

$$\{Y \otimes X \otimes I, Z \otimes Z \otimes I, -I \otimes I \otimes X\} = \left(\begin{array}{ccc|ccc|c} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right)$$

(c) *Stabiliser tableau*



(d) *Graph state with local Cliffords*

Figure 8.11: **Quantum state representations available in NetSquid.** Four different representations of the same quantum state $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + i|11\rangle)|-\rangle$. Each representation type is supported by NetSquid and has different trade-offs (see text of section 8.6.1 in Supplementary Note 8.6.1).

states. The stabiliser formalism (STAB) [68, 87] and graph states with local Cliffords formalisms (GSLC) [69] can only represent stabiliser states. Stabiliser states form a subset of all quantum states that are closed under the application of:

- *Clifford gates.* Each Clifford gate can be written as circuit consisting of the following three gates only: the Hadamard gate H (eq. (8.15)), the phase gate $|0\rangle\langle 0| + i|1\rangle\langle 1|$ and the CNOT gate $|00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 01| + |01\rangle\langle 10|$. Not all unitaries are Clifford gates;
- single-qubit measurements in the standard $(|0\rangle, |1\rangle)$ basis.

As such, for the STAB and GSLC formalisms quantum operations are limited to these two procedures. The runtime complexity trade-off between GSLC and STAB is nontrivial, since the former is faster on single-qubit unitaries, where the latter outperforms in two-qubit gates. An overview of the four formalisms and their runtime complexities can be found in Supplementary Table 8.1.

Now, let us describe for each of the formalisms how a quantum state is represented. An example of the different representations of the same quantum state is given in Fig. 8.11.

KET VECTORS (KET)

In the KET formalism, an n-qubit pure state $|\psi\rangle = \sum_{k=1}^{2^n} c_k|k\rangle$ is stored as a vector of length 2^n containing the complex amplitudes c_k. Here, $|k\rangle$ denotes the product state of the binary representation of k, e.g. $|5\rangle = |1\rangle \otimes |0\rangle \otimes |1\rangle$.

	Density Matrix (DM)	Ket state vector (KET)	Stabiliser tableau (STAB)	Graph state with local Cliffords (GSLC)
Is universal	Yes	Yes	No[1]	No[1]
Supports mixed states	Yes	No	No	No
Memory (bits)	128×2^{2n}	128×2^{n}	$2n^2 + n$	$\mathcal{O}(nd + n)$
Operating complexity	$\mathcal{O}(2^{3n})^2$	$\mathcal{O}(2^{2n})^2$	$\mathcal{O}(n)$	Single qubit gates: $\mathcal{O}(1)$ Two-qubit gates: $\mathcal{O}(d^2+1)$
Measurement complexity	$\mathcal{O}(2^{3n})^2$	$\mathcal{O}(2^{2n})^2$	$\mathcal{O}(n^3)$	$\mathcal{O}(d^2+1)$

Table 8.1: **The four different quantum state formalisms implemented in NetSquid.** Where n is the amount of qubits in the quantum states and d is the average amount of edges per vertex in the GSLC formalism with $0 \leq d < n$.

[a] Can only represent stabiliser states. The only operators that can operate on these states are Clifford operators.
[b] A stricter upper bound exists, depending on the matrix multiplication implementation.

DENSITY MATRICES (DM)

The density matrix of a pure state $|\psi\rangle$ is $|\psi\rangle\langle\psi| = |\psi\rangle \cdot (|\psi\rangle)^\dagger$, where $\cdot$ denotes matrix multiplication and $(.)^\dagger$ refers to complex transposition. An n-qubit mixed state is a statistical ensemble of n-qubit pure states and can be represented as

$$\sum_{k=1}^{m} p_k |\psi_k\rangle\langle\psi_k|$$

where $|\psi_1\rangle, \ldots, |\psi_m\rangle$ are n-qubit pure states (with $1 \leq m \leq n$) and the p_k are probabilities that sum to 1. In DM, the density matrix of a pure or mixed state is represented as a matrix of dimension $2^n \times 2^n$ with complex entries.

STABILISER TABLEAUS (STAB)

In the stabiliser formalism [87], one tracks the generators of the stabiliser group of a state. We briefly explain the concept here; for a more accessible introduction to the topic, we refer to [88]. In order to define a stabiliser group, let us give the Pauli group, which consists of strings of Pauli operators with multiplicative phases $\pm 1, \pm i$:

$$\{\beta \cdot \bigotimes_{k=1}^{n} P_k \mid P_k \in \{\mathbb{1}_2, X, Y, Z\} \text{ and } \beta \in \{\pm 1, \pm i\}\}.$$

A stabiliser group is a subgroup of the Pauli group which is commutative (i.e. any two elements A and B satisfy $A \cdot B = B \cdot A$) and moreover does not contain the element $-\mathbb{1}_2 \otimes \mathbb{1}_2 \otimes \cdots \otimes \mathbb{1}_2$. In case the stabiliser group contains 2^n elements, there is a unique quantum state $|\psi\rangle$ for which each element A from the stabiliser group stabilises $|\psi\rangle$, i.e. $A|\psi\rangle = |\psi\rangle$. Not all quantum states have such a corresponding stabiliser group; those that do are called stabiliser states. The intuition behind the stabiliser state formalism is that one tracks how the stabiliser group is altered by Clifford operations and $|0\rangle / |1\rangle$-basis measurements. Since the stabiliser state belonging to a stabiliser group is unique, one could in principle always convert the group back to any other formalism, such as KET. Concrete examples of stabiliser groups and their corresponding stabiliser states are:

- the stabiliser group $\{\mathbb{1}_2, Z\}$, which corresponds to the state $|0\rangle$;
- the stabiliser group $\{\mathbb{1}_2 \otimes \mathbb{1}_2, \mathbb{1}_2 \otimes Z, Z \otimes \mathbb{1}_2, Z \otimes Z\}$, which corresponds to the state $|0\rangle \otimes |0\rangle$;
- the stabiliser group $\{\mathbb{1}_2 \otimes \mathbb{1}_2, X \otimes X, Z \otimes Z, -Y \otimes Y\}$, which corresponds to the state $(|00\rangle + |11\rangle)/\sqrt{2}$.

Rather than tracking the entire 2^n-sized stabiliser group, it suffices to track a generating set, i.e. a set of n Pauli strings whose 2^n product combinations yield precisely the 2^n elements of the stabiliser group. The choice of generators is not unique. For the examples given above, example sets of stabiliser generators are:

- for $|0\rangle$, the stabiliser group is generated by the single element Z, since $Z^2 = \mathbb{1}_2$
- for $|00\rangle$, the stabiliser group is generated by $\{Z \otimes \mathbb{1}_2, \mathbb{1}_2 \otimes Z\}$, since squaring any of these two yields $\mathbb{1}_2 \otimes \mathbb{1}_2$, while multiplying them yields $Z \otimes Z$;

- for the state $(|00\rangle + |11\rangle)/\sqrt{2}$, one possible set of of generators is $\{X \otimes X, Z \otimes Z\}$.

In NetSquid we store generators as a stabiliser tableau:

$$\left|\begin{array}{ccc} X & Z & P \end{array}\right| = \left|\begin{array}{ccccccc} x_{11} & \dots & x_{1n} & z_{11} & \dots & z_{1n} & p_1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{n1} & \dots & x_{nn} & z_{n1} & \dots & z_{nn} & p_n \end{array}\right| \text{ where } p_k, x_{jk}, z_{jk} \in \{0,1\}, 0 < j,k \leq n$$

The k-th generator corresponds to the k-th row of this tableau and is given by

$$(-1)^{p_k} \bigotimes_{j=1}^{n} X^{x_{jk}} Z^{z_{jk}}$$

For updating the stabiliser tableau after the application of a Clifford gate or a $|0\rangle$ / $|1\rangle$-basis measurement, NetSquid uses the algorithms by [87] and [68]. The runtime performance of stabiliser tableau algorithms is a direct function of the number of qubits: linear for applying single- or two-qubit Clifford unitaries, which any Clifford can be composed into, and cubic for single-qubit measurement [87].

GRAPH STATES WITH LOCAL CLIFFORDS (GSLC)

The last formalism is GSLC: graph states with local Cliffords [69]. Graph states are a subset of all stabiliser states (see [89] for a review) and an n-qubit graph state $|\psi\rangle$ can be written as

$$|\psi\rangle = \prod_{(j,k)\in E} Z_{jk} |+\rangle^{\otimes n} \tag{8.6}$$

where Z_{jk} indicates a controlled-Z gate $|00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 10| - |11\rangle\langle 11|$ between qubits j and k, and we have denoted $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$. As such, a graph state is completely determined by the set of qubit index pairs (j,k) at which a controlled-Z operation is performed. These indices can be captured in a graph with undirected edges; in eq. (8.6), the edge set is E. Each stabiliser state can be written as a graph state, followed by the application of single-qubit Clifford operations. Thus, a stabiliser state in the GSLC formalism is represented by a set of edges E and a list of n single qubit Cliffords. There exist 24 single-qubit Cliffords, so the Clifford list only requires $\mathcal{O}(n)$ space. For updating the graph and the list of single-qubit Cliffords after the application of a Clifford gate or a $|0\rangle$ / $|1\rangle$-basis measurement, NetSquid uses the algorithms by [69]. The runtime scaling of the graph-state-based formalism depends on the edge degree d of the vertices involved in the operation – constant-time for single-qubit Cliffords, quadratic in d for two-qubit Cliffords and measurement – and thus scales favourably if the graph is sparse.

THE PYDYNAA SIMULATION ENGINE

The discrete-event modelling framework used by NetSquid is provided by the Python package PyDynAA, which is based on the core engine layer of DynAA, a system analysis and design tool [71]. This foundation provides a simple yet powerful language for describing large and complex system architectures. To realise PyDynAA, the simulation engine core was written in C++ for increased performance, and bindings to Python were

added using Cython. NetSquid takes advantage of the Cython headers exposed by PyDynAA to efficiently integrate the engine into its own compiled C extension libraries.

Several of NetSquid's sub-packages depend and build on the classes provided by PyDynAA, as illustrated in Figure 8.9 (main text). In Supplementary Figure 8.12 we highlight several of these key classes and how they interact with the simulation timeline in more detail, namely: the simulation engine (`SimulationEngine`), events (`Event` and `EventType`), simulation entities (`Entity`), and event handlers (`EventHandler`). We proceed to describe the concepts these classes represent in more detail.

Simulation *entities* represent anything in the simulation world capable of generating or responding to events. They may be dynamically added or removed during a simulation. The `Entity` superclass provides methods for scheduling events to the timeline at specific instances and waiting for them to trigger. The intended use is that users subclass the `Entity` class to implement their own entities. The *simulation engine* efficiently handles the scheduling of events at arbitrary (future) times by storing them in a self-balancing binary search tree. Events may only be scheduled by entities, which ensures that events always have a source entity. If an entity is removed during a simulation, then any future events it had scheduled will no longer trigger.

An entity responds to events by registering an event handler object with a callback function. Responses can be associated to a specific type, source, and id (including wildcard combinations). The simulation engine runs by stepping sequentially from event to event in a discrete fashion and checking if any event handlers in its registry match. A hash table together with an efficient hashing algorithm ensure efficient lookups of the event handlers in the registry.

PyDynAA implements an *event expression* class to allow entities to wait on logical combinations of events. Atomic event expressions, which describe regular wait conditions for standard events, can be combined to form composite expressions using logical *and* and *or* operators to any depth. Event expressions enable NetSquid simulations to deal with timing complexities. This feature has been used extensively in NetSquid to model both the internal behaviour of hardware components, as well as for programming network protocols. As example, consider DEJMPS entanglement distillation [63]: two nodes participate in this protocol and a node can only decide whether the distillation succeeded or failed when both its local quantum operations have finished and it has received the measurement outcome from the remote node. Thus, the node waits for the logical *and* of the receive-event and the event that the local operations have finished.

THE MODULAR COMPONENT MODELLING FRAMEWORK

The physical modelling of network devices is provided by several NetSquid subpackages: *components*, *models* and *nodes*, which are shown stacked with relation the NetSquid package in Figure 8.9 (main text). The pivotal base class connecting all them is the *component* (`Component`), which is used to model all hardware devices. Specifically, it represents all physical entities in the simulation, and as such sub-classes the *entity* (`Entity`),which enables it to interact with the event timeline. In Supplementary Figure 8.13 we show a class diagram of the component class and its relationships to other classes from these sub-packages.

The modularity of NetSquid's modelling framework is achieved by the composition of components in terms of *properties*, *models*, communication *ports* and *sub-*

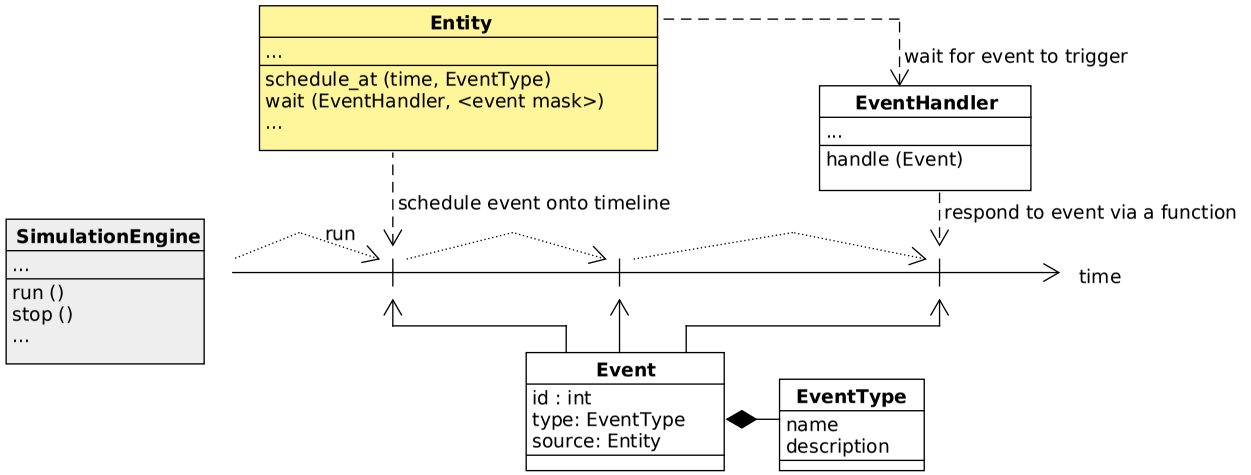


Figure 8.12: **Design overview of the PyDynAA package.** Schematic overview of key classes defined by the PyDynAA package, the discrete-event simulation engine used by NetSquid. Also shown is the relation of each class to the simulation timeline. Events are scheduled onto the simulation timeline by `Entity` objects. Entities wait for events to trigger by registering `EventHandlers`, which respond to an event by passing it as input to a specified callback function. The events to wait for can be specified by their type, id, and source entity. Ellipses indicate that not all of a class's public variables and methods are listed. Omitted from this class diagram is the `EventExpression` class – see the text for more details.

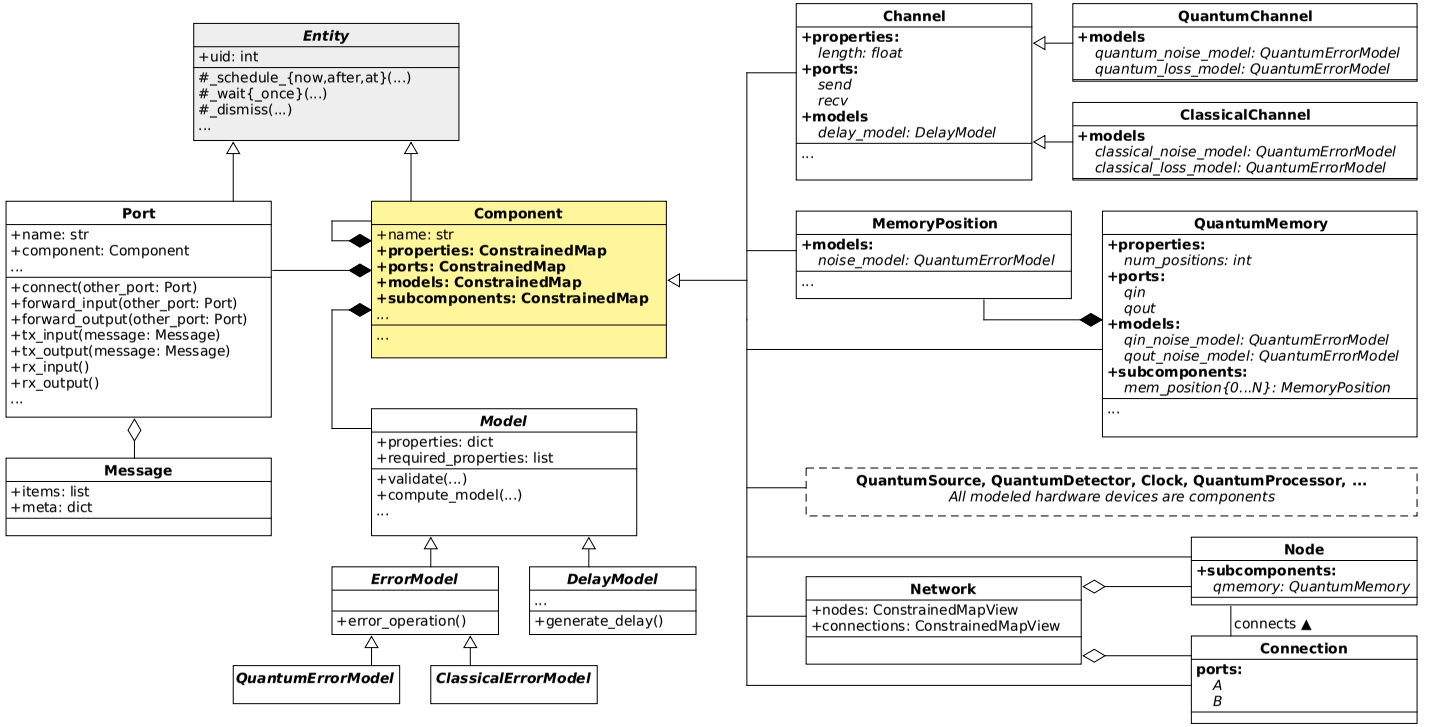


Figure 8.13: **Design overview of components in NetSquid.** Class diagram for the `Component` class, a simulation entity that is used to model all network hardware devices, including composite components such as nodes, connections and the network itself. A component is shown to be composed of *properties*, *ports*, *models* and *subcomponents*. Ellipses indicate that not all of a class's public variables and methods are listed.

components. A component's *properties* are values that physically characterise it, such as the length of a channel or the frequency of a source. A special *constrained map* (`ConstrainedMap`) container is used to store the properties (as well as the other composed objects) to give control of the expected types and immutability of properties during a simulation. *Models* (`Model`) are used to describe the physical behaviour of a component, such as the transmission delay of a channel, or the quantum decoherence of a qubit in memory. Model objects are essentially elaborate functions and generally do not store any state; when a model is called it is passed its component's properties, in addition to any modelling specific input, such as, in the case of a quantum noise model, the qubit to apply noise and the time the qubit has been waiting on a memory. Components can be composed of other *subcomponents*, which allows networks to be pieced together in a very modular fashion. For instance, a complete network can be represented by a single component, which is composed of node and connection sub-components, which in turn are composed of devices such as channels, sources, memories, etc. To streamline and automate the communication between components, including to and from subcomponents, components can be linked using *ports* (`Port`) that can send, receive and forward both quantum and classical *messages* (`Message`).

While the component base class defines a modular interface for modelling all kinds of hardware, it doesn't internally implement any event-driven behaviour itself. That behaviour is implemented by a library of base classes that sub-class `Component`. The right half of Supplementary Figure 8.13 shows the sub-classing hierarchy of the provided components, ranging from quantum and classical channels, quantum memory and processing devices, sources, detectors, clocks, to nodes, connections, and networks.

The *quantum processor* (`QuantumProcessor`) is a component from the base class library used for modelling general quantum processing devices. It sub-classes the *quantum memory* (`QuantumMemory`) component, from which it inherits a collection of *quantum memory positions* (`MemoryPosition`) for tracking the quantum noise of stored qubits. The processor can assign a set of *physical instructions* to these positions to describe the operations possible for manipulating their stored qubits, such as quantum gates and measurements, or initialisation, absorption, and emission processes. The physical instructions map to general device-independent instructions, for which they specify physical models such as duration and error models specific to the modelled device. This mapping allows users to write *quantum programs* in terms of device-independent instructions and re-use them across devices. The quantum programs can include classical conditional logic, make use of parallel execution (if supported by the device), and import other programs.

ASYNCHRONOUS PROGRAMMING NETWORKS USING PROTOCOLS

While components are entities in the simulation describing physical hardware, *protocols* – represented by the `Protocol` base class as shown in Supplementary Figure 8.14 – are entities that describe the intended virtual behaviour of a simulation. In other words, the protocol base class is used to model the various layers of software running on top of the components at the various nodes and connections of a network. That can include, for instance, any automated control software at the physical or link layers of a quantum network stack, up to higher-level programs written at the application layer.

Protocols in NetSquid can be likened to background processes: they can be started, stopped, as well as reset to clear any state. They can also be nested i.e. a protocol can manage the execution of *sub-protocols* under its control. To communicate changes of state, such as a successful or failed run, protocols can use a *signalling* mechanism (`Signal`).

NetSquid defines several sub-classes of the protocol base class that add extra restrictions or functionality. To restrict the influence of a protocol to only a local set of nodes the *local protocol* (`LocalProtocol`) can be used. Similarly, to restrict a protocol to executing on only a single node, which is a typical use case, a *node protocol* (`NodeProtocol`) is available. The *service protocol* (`ServiceProtocol`) describes a protocol interface in terms of the types of requests and responses they support. Lastly, a *data node protocol* adds functionality to process data arriving from a port linked to a connection, and the *timed node protocol* supports carrying out actions at regularly timed intervals.

Programming a protocol involves waiting for and responding to events, which is achieved in the simulation engine by defining event handlers that wrap callback functions. As the complexity of a protocol grows, typically the flow and dependencies of the callback calls do too. To make the asynchronous interaction between protocol and component entities easier and more intuitive to program and read, the main execution function of a protocol (the `run()` method) can be suspended mid-function to wait for certain combinations of events to trigger. This is implemented in Python using the `yield` statement, which takes as its argument an event expression. Several helper methods have been defined that generate useful event expressions a protocol can *await*, for instance: `await_port_input()` to wait for a message to arrive on a port, or `await_timer()` to have the protocol sleep for some time.

8.6.2. QUANTUM CIRCUITS AND NETWORK SETUPS FOR BENCHMARKING

In this section we extend the Methods, section 8.3.1, to provide additional details on the benchmarking simulations presented in the Results, section 8.2.4.

BENCHMARKING OF QUANTUM COMPUTATION RUNTIME

The quantum circuit used to benchmark the runtime for generating an n qubit GHZ state is shown in Supplementary Figure 8.15a. The n qubits are created in NetSquid with independent quantum states and are combined into the larger state via the CNOT operation. The measurement operations at the end of the circuit are performed sequentially and each split the measured qubit from its shared quantum state. Unless otherwise specified the KET and DM formalisms utilise memoization (see Methods, section 8.3.1). Memoization is effective because the circuit is successively iterated 30 times. The reported runtime is the mean runtime of the iterations. For the baseline comparison with the ProjectQ simulator we set up the circuit in an analogous way to NetSquid, and its default `MainEngine` was used with no special settings applied. Qubits are similarly added sequentially to the growing state via the CNOT operation, and also the measurements are performed sequentially with the measured qubit directly deallocated afterwards.

The quantum circuit used to benchmark the runtime of only the quantum computation involved for a simple repeater chain involving n qubits is shown in Supplementary Figure 8.15b. It is implemented for NetSquid and ProjectQ similarly to the GHZ bench-

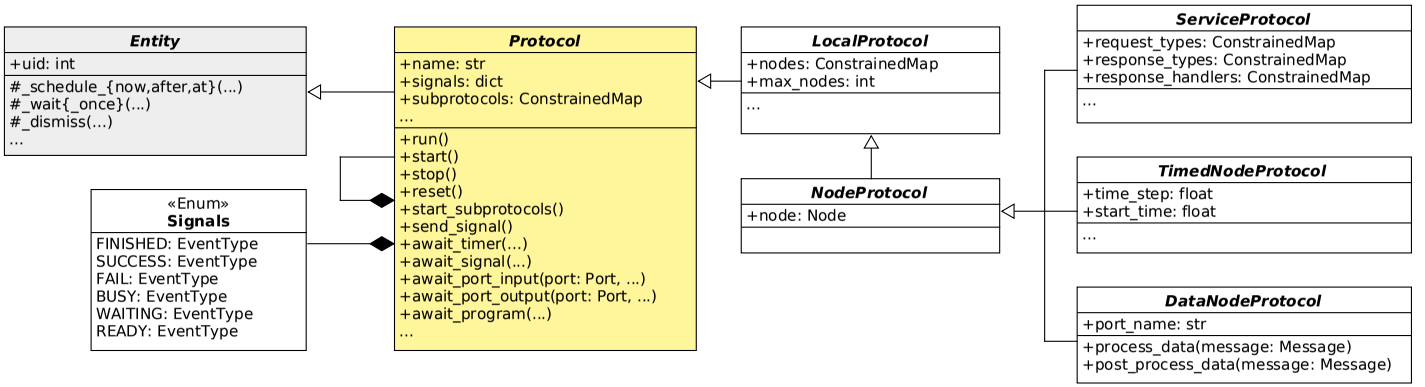
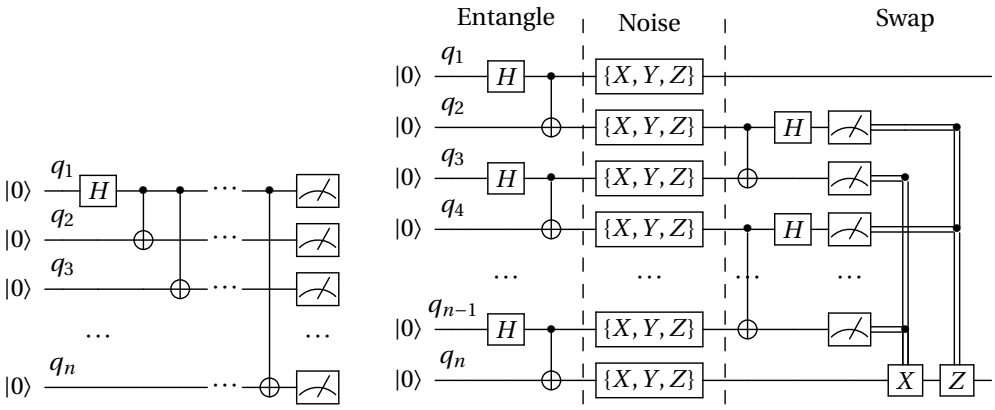


Figure 8.14: **Design overview of protocols in NetSquid.** Class diagram of the `Protocol` class and its subclasses. Ellipses indicate that not all of a class's public variables and methods are listed.



(a) GHZ state generation

(b) Repeater chain quantum computation

Figure 8.15: **Circuits used to benchmark quantum computation in the Results, section 8.2.4, for n qubits.** For panel (b) the CNOT control line crossing the ellipses represents multiple lines for $n > 6$ qubits, following the pattern of q_2 and q_3. Similarly, the classical control lines represent an AND of the measurement results for $q_3, q_5, \ldots, q_{n-1}$ and $q_2, q_4, \ldots, q_{n-2}$ to determine the control of the X and Z gates, respectively. The noise gates denoted by {X,Y,Z} cycle through the Pauli gates (see main text). Note that this circuit always requires an even number of qubits.

mark, with qubits only combining their quantum states when a multi-qubit gate is performed. An option has been added to keep qubits *inplace* after measurements i.e. they are not split from their shared quantum states – in ProjectQ this is achieved by keeping a reference to prevent deallocation. Noise is applied to each qubit after entanglement by selecting a Pauli gate to mimic depolarising noise, which is done deterministically for convenience. For this process the runtime is also determined as the mean of 30 successive iterations.

To benchmark the runtimes of quantum computation circuits the processes were timed in isolation from any setup code using the Python *timeit* package. Python garbage collection is disabled during the timing of each process. To avoid fluctuations due to interfering CPU processes the reported time is a minimum of five repetitions.

RUNTIME PROFILING OF A REPEATER CHAIN SIMULATION

The runtime profiling of NetSquid presented in the Results, section 8.2.4, is performed for a simple repeater chain. The network setup of this simulation extends the single repeater presented in Supplementary Figure 8.1 to a chain of nodes by adding the entangling connection shown between each pair of neighbouring nodes. Direct classical connections are connected between each node and one of the end-nodes, rather than between neighbouring nodes, and are used to transmit the swapping corrections. The chosen configuration for this network does not need to be physically realistic; it suffices for it to be representative of the typical computational complexities. The nodes are placed at 20km intervals and the channels transmit messages at the speed of light in fibre. The entanglement sources, assumed to be perfect, are all synchronised and oper-

ate at a frequency of 100 kHz. Physical non-idealities are represented by adding time-dependent depolarising noise to both the quantum channels and quantum memories, as well as dephasing noise to quantum gates. The corresponding depolarising and dephasing times are 0.1 s and 0.04 s, which correspond to the T_1 and T_2 times presented in section 8.3.2 of the Methods.

In a simulation run entanglement is created once between the end-nodes by performing entanglement swaps along the chain. Protocols are assigned to all but the end-nodes to perform entanglement swaps after each round of entanglement generation, and send their measurement results as corrections to the same end-node. A protocol running on the end-node collects these corrections, and applies them if needed.

The runtime of this simulation is profiled to determine the distribution of time spent in the functions of NetSquid's sub-packages, as well as its dependency packages NumPy and PyDynAA. To perform this profiling the `cProfile` package is used. The reported runtime for a given number of nodes is the mean of 400 successive simulation runs.

8.6.3. QUANTUM SWITCH: PHYSICAL NETWORK AND PROTOCOL

Here, we provide the details of the quantum switch simulations, whose results are presented in section 8.2.2 of the Results.

We implement the model of Vardoyan et al. [52], for which the parameters of the simulation are:

- the number of leaf nodes k;
- the desired size n of the shared entanglement on the leaf nodes;
- for each leaf node: the rate μ at which bipartite entanglement is generated between leaf node and switch;
- B: the buffer size, i.e. the number of dedicated qubits per leaf node at the switch.

In addition, we include T_2, the memory coherence time.

PHYSICAL NETWORK

In the scenario we study, the quantum switch is the centre node of a star-topology network, with $k \geq 2$ leaf nodes. Each leaf node individually is connected to the switch by a *connection*, which consists of a *source* producing perfect bipartite entangled states $(|00\rangle + |11\rangle)/\sqrt{2}$ on a randomised *clock* and two *quantum connections*, from the source to the leaf and switch node, respectively, for transporting the two produced qubits. The interval Δt between clock triggers is randomly sampled from an exponential distribution with probability $\mu \cdot e^{-\mu \cdot \Delta t}$ where μ is the rate of the source. We set the delay of the quantum channels to zero.

Each node holds a single *quantum processor* with enough quantum memory positions for the total duration of our runs. Each memory position has a T_2 *noise model*: if a qubit is acted upon after having resided in memory for time Δt, then a dephasing map (eq. (8.1)) is applied with dephasing probability $p = \frac{1}{2}\left(1 - e^{-\Delta t / T_2}\right)$. Each quantum processor can perform any unitary operation or single-qubit measurement; these operations are noiseless and take no time.

PROTOCOL OF THE SWITCH NODE

The switch node continuously waits for incoming qubits. Upon arrival of a qubit from leaf node ℓ, the switch first checks whether it shares more entangled pairs of qubits with ℓ than the pre-specified buffer size B; if so, it discards the oldest of those pairs. Then, it checks whether it holds entangled pairs with at least n different leaves. If so, then it performs and n-qubit GHZ-basis measurement (see below) on its qubits of those pairs. If multiple groups of n qubits from n distinct nodes are available, then it chooses the oldest pairs.

Directly after completion of the GHZ-basis measurement, we register the measurement outcomes and obtain the resulting n-partite entangled state $|\psi\rangle$ on the leaf nodes. From these, the fidelity $|\langle\psi|\phi_{\text{ideal}}\rangle|^2$ with the ideal target GHZ state $|\phi_{\text{ideal}}\rangle$ is computed.

The n-qubit GHZ states are

$$\left(|0\rangle\otimes|b_2\rangle\otimes|b_3\rangle\otimes\cdots\otimes|b_n\rangle+(-1)^{b_1}|1\rangle\otimes|\overline{b_2}\rangle\otimes|\overline{b_3}\rangle\otimes\cdots\otimes|\overline{b_n}\rangle\right)\Big/\sqrt{2} \tag{8.7}$$

where $b_j\in\{0,1\}$ and we have denoted $\overline{b}=1-b$. The n-qubit *quantum program* that the switch node applies for performing a measurement in the n-qubit GHZ basis is as follows: first, a CNOT operation on qubits 1 and j (1 is the control qubit) is applied for all $j=2,3,\ldots,n$, followed by a Hadamard operation (eq. 8.15) on qubit 1. Then, all qubits are measured in the $|0\rangle/|1\rangle$-basis. If we denote the outcome of qubit j as b_j, the GHZ-state that is measured is precisely the one in eq. (8.7).

8.6.4. HARDWARE PARAMETERS FOR THE NV REPEATER CHAIN

Here, we provide the values for the hardware parameters of the nitrogen-vacancy setup used in our simulations. An overview of all parameters is provided in Supplementary Table 8.2, including two example sets of improved parameters following the approach in section 8.3.2 of the Methods.

ELEMENTARY LINK GENERATION

For generating entanglement between the electron spins of two remote NV centres in diamond, we simulate a scheme based on single-photon detection, following its experimental implementation in [84]. The setup consists of a middle station which is positioned exactly in between two remote NV centres in diamond. The middle station is connected to the two NVs by glass fibre and contains a 50:50 beam splitter and two non-number resolving photon detectors. In the single-photon scheme, each NV performs the following operations in parallel. First, the electron of each NV system is brought into the state $\sqrt{\alpha}|0\rangle+\sqrt{1-\alpha}|1\rangle$ by optical and microwave pulses, where α is referred to as the bright-state parameter. Then, a laser pulse triggers the emission of a photon, yielding the spin-photon state $\sqrt{\alpha}|0\rangle_s\otimes|1\rangle_p+\sqrt{1-\alpha}|1\rangle_s\otimes|0\rangle_p$, where $|0\rangle$ ($|1\rangle$) denotes absence (presence) of a photon. We set $\alpha=0.1$ since for that value, fidelity is approximately maximal at lab-scale distances [84]; optimising over α is out of the scope for this work. We assume that the delay until emission of the photon is fixed at 3.8 μs [90].

From each NV centre, the emitted photons are transmitted to the middle station through glass fibre, where a 50:50 beam splitter effectively erases the which-way information of an incoming photon. An attempt at generating entanglement using this

	Noise parameter ('near-term')	Duration/time	Probability of no-error	Improved noise param. 3×	Improved noise param. 10×
Probability of double excitation p_{dexc} (8.6.4)	0.06	-	p_{dexc}	0.01	0.003
Transmission loss γ (dB/km, 8.6.4)	0.2	-	×	×	×
Dark count probability p_{dc} (8.6.4)	$2.5 \cdot 10^{-8}$	-	$1 - p_{\text{dc}}$	$8.3 \cdot 10^{-8}$	$2.5 \cdot 10^{-9}$
Probability of photon detection (8.6.4) for zero-length fibre $p_{\text{det}}^{\text{nofibre}}$	0.0046	-	$p_{\text{det}}^{\text{nofibre}}$	0.16	0.58
Interferometric phase uncertainty σ_{phase} (rad, 8.6.4)	0.35	-	$1 - p_{\text{phase}}$ (eq. (8.10))	0.20	0.11
Photon visibility V (8.6.4)	0.9	-	V	0.97	0.99
$N_{1/e}$: indicates nuclear dephasing during electron initialization (8.6.4)	1400	-	p from eq. (8.12)	4206	14006
Electron T_1 (8.6.4)	-	1h	e^{-1/T_1}	2.8h	10h
Electron T_2^* (8.6.4)	-	1.46 s	e^{-1/T_2^*}	4.4s	14.6s
Carbon T_1 (8.6.4)	-	10h	e^{-1/T_1}	27h	100h
Carbon T_2 (8.6.4)		1s	e^{-1/T_2}	3s	10s
Carbon initialization to $\|0\rangle$ (8.6.4)	F=0.997	310 μs	$2F-1$	$F = 0.999$	$F = 0.9997$
Carbon Z-rotation gate (8.6.4)	F=0.999	20 μs	$4(F-1)/3$	$F > 0.9999$	$F > 0.9999$
E-C controlled-R_X-gate (electron=control) (8.6.4)	F_{EC}=0.97	500 μs	$(4\sqrt{F_{EC}}-1)/3$	$F_{EC} = 0.990$	$F_{EC} = 0.997$
Electron initialization to $\|0\rangle$ (8.6.4)	F=0.99	2 μs	$2F-1$	$F = 0.997$	$F = 0.999$
Electron single-qubit gate (8.6.4)	F=1	5 ns	$(4F-1)/3$	$F = 1$	$F = 1$
Electron readout (eq. (8.4) and sec. 8.6.4)	0.95/0.995(f_0/f_1)	3.7 μs	f_x	0.983/0.9983	0.995/0.9995

Table 8.2: **Physical parameters dealing with elementary link generation, memory coherence times and duration and fidelities (F) of the gates.** Depicted are both parameters of the dataset 'near-term' and two examples of improved parameter sets (see Methods, section 8.3.2), for 3 times and 10 times improved, respectively, together with the function to convert the parameter to a 'probability of no-error' to compute the improved parameter value for other factors. The 'near-term' values correspond to 1× improvement. The transmission loss parameter γ is not changed by the improvement procedure and equals $\gamma = 0.2$ dB/km during any of our simulations.

single-click scheme is declared successful if precisely one of the detectors clicks, which happens if either (a) a single photon arrives at the detector and the other does not or (b) both photons arrive (in case (b), only a single detector clicks due to the Hong-Ou-Mandel effect). Case (a) yields the generation of the spin-spin state $|\phi_\pm\rangle = (|01\rangle \pm |10\rangle)/\sqrt{2}$, where $\pm$ indicates which of the two detectors clicked, while case (b) results in $|00\rangle\langle 00|$. Given that a single photon arrives, the probabilities that the other photon has or has not arrived are respectively $1-\alpha$ and α (in the absence of loss). Therefore, a successful attempt results in the generation of the spin-spin state $(1-\alpha)\left|\phi_\pm\right\rangle\left\langle\phi_\pm\right| + \alpha|00\rangle\langle 00|$. We refer to [84] for a more in-depth description of the scheme. We assume that the speed of the photons and of all classical communication equals c/n_{ri}, where c is the speed of light in vacuum and $n_{ri} = 1.44$ is the refractive index of glass [91].

In reality, however, several sources of noise affect the produced state, which we treat below.

IMPERFECT DETECTION

The total probability p_{det} that a photon, emitted by the NV, will be detected in the midpoint is given by the product of four probabilities [83]

- the probability $p_{\text{zero\_phonon}}$ that the photon frequency is in the zero-phonon line [92];
- the probability $p_{\text{collection}}$ that the photon is collected into the glass fibre;
- the probability $p_{\text{transmission}}$ that the photon does not dissipate in the fibre during transmission;
- the probability $p_{\text{detection}}$ that the photon is detected, conditioned on the fact that it reaches the detector.

Thus we can write

$$p_{\text{det}} = p_{\text{det}}^{\text{nofibre}} \cdot p_{\text{transmission}} \tag{8.8}$$

where

$$p_{\text{det}}^{\text{nofibre}} = p_{\text{zero\_phonon}} \cdot p_{\text{collection}} \cdot p_{\text{detection}}. \tag{8.9}$$

The transmission probability is given by

$$p_{\text{transmission}} = 10^{-(L/2)\cdot\gamma/10}$$

where L is the internode distance (i.e. $L/2$ is the length of the fibre from NV to middle station) and γ is the loss parameter which depends on the photon frequency. In our simulations, we assume that the photon frequency is converted to the telecom frequency, corresponding to $\gamma = 0.2$ dB/km. Also, we assume that the emission in the zero-phonon line is enhanced by an optical cavity from $p_{\text{zero\_phonon}} = 3\%$ (without cavity) to $p_{\text{zero\_phonon}} = 46\%$ (with cavity) [92]. We set the detection efficiency $p_{\text{detection}}$ to 0.8 [93].

What remains is the collection efficiency $p_{\text{collection}}$, which we compute from experimental values (no cavity, no conversion to the telecom frequency) using eq. (8.8)

with $L = 2\text{m}$, $p_{\text{det}} = 0.001$ [90] and $\gamma = 5$ dB/km [34] (for the zero-photon line frequency), yielding $p_{\text{collection}} = 0.042$. Since frequency conversion to the telecom frequency is a probabilistic process and only succeeds with probability 30% [94], we set $p_{\text{collection}} = 0.3 \cdot 0.042$.

OTHER SOURCES OF NOISE

Other sources of noise on the freshly generated electron-electron entanglement are

- Dark counts: a photon detector falsely registering. The dark count probability follows a Poisson distribution $p_{\text{dc}} = 1 - e^{-t_w \cdot \lambda_{\text{dark}}}$ where $t_w = 25$ ns [84] is the duration of the time window at the midpoint. We set $\lambda_{\text{dark}} = 1$ Hz as the dark count rate.

- Imperfect photon indistinguishability. The generation of entanglement at the middle station is based upon the erasure of the which-way information with respect to the path of the photons. Only in case the photons are fully indistinguishable, the which-way information is erased perfectly. The overlap of the photon states is given by the visibility V, which we set to 0.9 [84].

- Double excitation of the electron spin. When triggered to emit a photon by a resonant laser pulse, an NV centre could be excited twice, which results into the emission of two photons. We set its occurrence probability to $p_{\text{dexc}} = 0.06$ [90].

- Photon phase uncertainty. The photons which interfere at the midpoint acquired a phase during transmission and a difference of these phases influences the precise entangled state that is produced [95]. Given a standard deviation $\sigma_{\text{phase}} = 0.35$ rad [90] of the acquired phase, we compute the dephasing probability as [84]

$$p_{\text{phase}} = \frac{1}{2}\left(1 - e^{-\sigma_{\text{phase}}^2/2}\right). \tag{8.10}$$

NUCLEAR SPIN DEPHASING DURING ENTANGLEMENT GENERATION

The initialisation of the electron spin state induces dephasing of the carbon spin states through their hyperfine coupling. Following [96], we model this uncertainty by a dephasing channel for each attempt with dephasing probability

$$p_{\text{single}} = \frac{1}{2}(1-\alpha) \cdot (1 - e^{-C_{\text{nucl}}^2/2}). \tag{8.11}$$

The parameter C_{nucl} is the product of the coupling strength between the electron spin and the carbon nuclear spin, and an empirically determined decay constant. Rather than expressing the dephasing probability as function of C_{nucl}, we express the magnitude of nuclear dephasing as $N_{1/e}$, the number of electron spin pumping cycles after which the Bloch vector length of a nuclear spin in the state $(|0\rangle + |1\rangle)/\sqrt{2}$ in the $X-Y$ plane of the Bloch sphere has shrunk to $1/e$, when the electron spin state has bright-state parameter $\alpha = 0.5$ (i.e the electron spin is in the state $(|0\rangle + |1\rangle)/\sqrt{2}$).

Let us compute how p_{single} depends on $N_{1/e}$ instead of on C_{nucl}. First, we find by direct computation that the equatorial Bloch vector length of a state is shrunk by a factor

$1-2p$ after a single application of the single-qubit dephasing channel (eq. (8.1)) with probability p ($p \leq \frac{1}{2}$).

Equating $(1-2p)^{N_{1/e}} = 1/e$ yields

$$p = \frac{1}{2}\left(1 - e^{-1/N_{1/e}}\right). \tag{8.12}$$

Equating p_{single} from eq. (8.11) with $\alpha = 0.5$ and p from eq. (8.12), followed by solving for C_{nucl} yields

$$1 - e^{-C_{\text{nucl}}^2/2} = 2\left(1 - e^{-1/N_{1/e}}\right).$$

Substituting back into eq. (8.11) yields an expression for general α:

$$p_{\text{single}} = (1-\alpha)\left(1 - e^{-1/N_{1/e}}\right). \tag{8.13}$$

We set $N_{1/e} = 1400$ [97].

LOCAL PROCESSING PARAMETERS

For the dynamics of the electron spin, we use $T_1 = 1$ hour and $T_2^* = 1.46$s [98]. For the carbon nuclear spin, we take $T_1 = 10$ hours and $T_2 = 1$ s (experimentally realised: $T_1 = 6$m and $T_2 \approx 0.26 - 25$s [99]). For the noise of the controlled-R_X gate (Methods, section 8.3.2), we set the depolarising probability $p = 0.02$ (denoted as p_{EC} in Supplementary Table 8.2), since by simulation of the circuit [95, Fig. 2a], we find that this value agrees with the experimentally found effective circuit fidelity of 0.95. The corresponding fidelity of the gate is $F_{EC} = (1 - 3p_{EC}/4)^2 = 0.97$. The initialisation fidelities of the electron and carbon spins are set at 0.99 [100] and 0.997 [99]. We use 0.999 for the carbon Z-rotation gate fidelity (experimentally achieved: 1 [101]). The durations of local operations are identical to our earlier simulations (see Appendix D, Table 6 in [34] and references therein). We summarise all hardware values in Supplementary Table 8.2.

8.6.5. PROTOCOLS AND QUANTUM PROGRAMS FOR THE NV REPEATER CHAIN

Here, we first elaborate on the sequence of quantum operations and classical communication that the NV protocol building blocks consist of (Methods, section 8.3.2). Then, we describe in detail the two repeater chain protocols we simulated.

OPERATIONS FOR THE BUILDING BLOCKS: STORE, RETRIEVE, DISTILL AND SWAP

STORE is the mapping of the electron spin state onto a free nuclear spin. The operation requires the nuclear spin state to be $|0\rangle$ and the circuit, given in Supplementary Figure 8.16(a), performs the following mapping:

$$|\phi\rangle_e \otimes |0\rangle_n \mapsto |0\rangle_e \otimes \left(H|\phi\rangle_n\right) \tag{8.14}$$

where $|\phi\rangle$ is an arbitrary single-qubit state and

$$H := \frac{1}{\sqrt{2}}\left(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| - |1\rangle\langle 1|\right) \tag{8.15}$$

is the Hadamard gate. By RETRIEVE (Supplementary Figure 8.16(b)), we denote the reverse operation,

$$|0\rangle_e \otimes |\phi\rangle_n \mapsto \left(H|\phi\rangle_e\right) \otimes |0\rangle_n .$$

We simulate the specific entanglement distillation protocol (DISTILL) from Kalb et al. [95], which acts upon an electron-electron state and a nuclear-nuclear state to probabilistically increase the quality of the nuclear-nuclear state, at the cost of having to read out the electron-electron state. In the protocol, the two involved nodes each perform a sequence of local operations including a measurement (Supplementary Figure 8.16(c)), followed by communicating the measurement outcome from the circuit to each other. In this work, we only use distillation in one of the two repeater schemes we consider (NESTED-WITH-DISTILL) and in that case, the success condition is as follows: if the nodes are adjacent, then the measurement outcomes should both equal 0 (i.e. the bright state of the electron), while otherwise the measurement outcomes only need to be equal (i.e. both 0 or both 1). In the case of failure, the nuclear-nuclear state is considered lost.

The entanglement swap (SWAP) converts two short-distance entangled qubit pairs $A-M$ and $M-B$ into a single long-distance one $A-B$, where A, B and M are nodes. It is equivalent to performing quantum teleportation [102] to a qubit which is part of a larger remote-entangled state. Our entanglement swapping protocol at node M starts by assuming that one of M's qubits which is involved in the entanglement swap is the electron spin. Then, a series of local operations including measurements (Supplementary Figure 8.16(d)) is performed; the measurement outcomes are transferred to both A and B. In the original teleportation proposal, B performs a local operation to correct the state $A-B$ to the expected one. However, due to the fact that such correction operation is generally not directly possible to perform on the nuclear spin state (see the allowed operations in section 8.3.2 of the Methods), we opt for the approach where the correction operation is tracked in a classical database. Details of this tracking, including how it affects the entanglement distillation and entanglement swap protocols, are given in Supplementary Note 8.6.6.

8

REPEATER CHAIN PROTOCOLS SWAP-ASAP AND NESTED-WITH-DISTILL

We describe two protocols for the NV repeater chain: SWAP-ASAP, a protocol where a node performs an entanglement swap as soon as it holds two entangled pairs, one in each direction of the chain, and NESTED-WITH-DISTILL, a nested protocol with distillation at each nesting level which is based on the BDCZ protocol [4]. Both protocols run asynchronously on each node.

In both protocols, a node remains idle until it is triggered to check whether it should perform an action. It is triggered at the following three moments: (a) at the start of the simulation, (b) when the node receives a classical message (if a node is busy upon reception, the message is stored and responded to later) , (c) when its previous action is finished. A simulation run finishes as soon as the two end nodes share a single entangled pair of qubits.

For the SWAP-ASAP protocol, the sequence of operations that a node performs depends on its index in the chain (start counting from left to right, nodes have indices $1, 2, 3, \ldots$). If the index of the node is even, the node sends a request for ENTGEN to its left neighbour, and starts the operation as soon as it has received confirmation. After

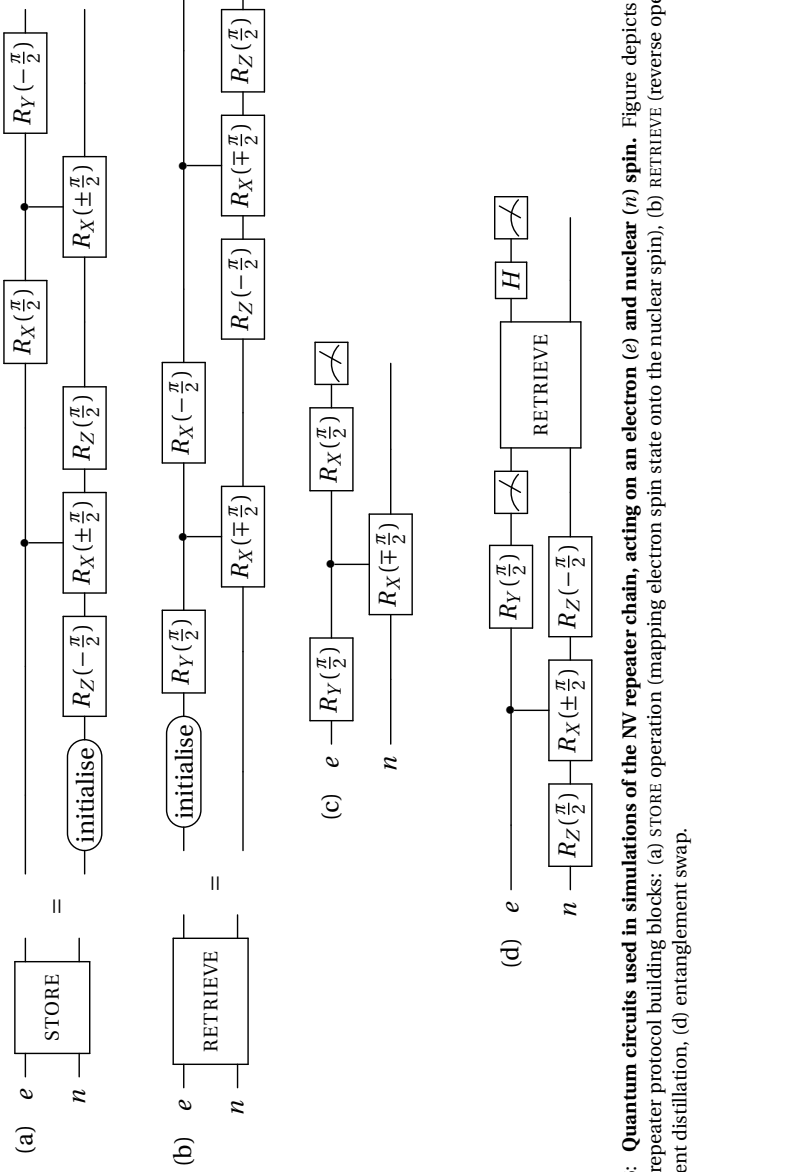


Figure 8.16: **Quantum circuits used in simulations of the NV repeater chain, acting on an electron (e) and nuclear (n) spin.** Figure depicts the quantum circuit for the NV repeater protocol building blocks: (a) STORE operation (mapping electron spin state onto the nuclear spin), (b) RETRIEVE (reverse operation to STORE), (c) entanglement distillation, (d) entanglement swap.

performing STORE to free the electron spin, it repeats it for its right neighbour. Odd-indexed nodes remain idle until reception of an ENTGEN request, after which they perform STORE if necessary to free the electron, send a confirmation, sleep for the duration of the message transmission, followed by performing ENTGEN. Once a node has entanglement with both directions, it performs a SWAP and sends the outcome to the end nodes.

The two end nodes are exceptions to the above. The left end node (i.e. with index 1) behaves like an odd-indexed node, but without performing SWAP. The same holds for the rightmost node (i.e. the node with the largest index), unless its index is even, in which case it initiates and performs entanglement generation with the adjacent node on the left.

The NESTED-WITH-DISTILL protocol is a variant of the BDCZ protocol [4], adapted to the fact that an NV cannot perform multiple ENTGEN , DISTILL or SWAP operations in parallel due to its restricted topology (Methods, section 8.3.2). In the adapted version, nodes take the role of *initiator* of one of the three main actions (ENTGEN, DISTILL, SWAP) if the action occurs at the highest nesting level that this node belongs to. To be precise, we do the following. In a repeater chain with 2^n+1 nodes, denote by $\{0,1,2,\ldots,2^n\}$ the indices of the nodes from left to right. A node (not an end node) with index $k \in \{1,2,3,\ldots,2^n-1\}$ initiates an action only if the entanglement that is involved in the task spans precisely $f_n(k)$ segments, where

$$
\begin{aligned}
f_1(k) &= 1 \text{ for all } k \\
f_n(k) &= \begin{cases} f_{n-1}(k) & \text{if } k < 2^{n-1}, \\ f_{n-1}(2^n-k) & \text{if } k > 2^{n-1}, \\ 2^{n-1} & \text{if } k = 2^{n-1}. \end{cases}
\end{aligned}
$$

End nodes are never initiators.

When a node (index k) is triggered, it performs the following checks in order and performs the first action for which the check holds true:

1. If it shares entangled pairs with nodes $k-f_n(k)$ and $k+f_n(k)$, and both are the immediate result of successful distillation: perform SWAP and send the measurement outcomes to the involved nodes

2. If it holds two entangled pairs with node $k-f_n(k)$ and neither pair is the result from successful entanglement distillation: send a request to distill to the node, wait for confirmation, followed by performing DISTILL

3. Same as 2, but now for DISTILL on the right, i.e. remote node has index $k+f_n(k)$

4. If there are any request-messages that have not been responded to yet: pick the oldest one and act as follows. Respond to the message with a confirmation message, followed by sleeping for the time that the confirmation takes to arrive at the remote node. Then perform the requested action (ENTGEN or DISTILL).

5. If $f_n(k) = 1$ and the node does not hold entanglement with its immediate left neighbour that is the result of successful entanglement distillation: send a request for ENTGEN to the node, wait for confirmation, followed by performing ENTGEN.

6. Same as 5 for right adjacent node.

If no action follows from the checks above, then the node remains idle until the next time at which it is triggered. In the operations above, if necessary ENTGEN is preceded by STORE to free the electron by storing its state into a free carbon spin. DISTILL is preceded by a combination of STORE and RETRIEVE to ensure the correct state lives on the electron spin, and so is SWAP in case neither of the two to-be-swapped qubits live on the electron. Since end nodes are never initiators, they only check 4.

8.6.6. TRACKING OF CORRECTION OPERATIONS IN THE NV REPEATER CHAIN

Here, we explain how nodes of the NV repeater chain track the precise entangled state they hold. This is done by associating unitary operations to each qubit, which map the state of two remotely entangled qubits back to $(|01\rangle + |10\rangle)/\sqrt{2}$ in the ideal case. Tracking these unitaries (gates) in a classical database, instead of performing them on the (imperfect) quantum hardware, has the advantage of avoiding gate noise. This argument is even stronger for NV centres in case the remote-entangled state is held by a carbon nuclear spin, because direct application of a correction operator to a carbon spin is generally not possible due to the restricted topology of the NV quantum processor (Methods, section 8.3.2). Thus, performing the correction operator to the nuclear spin requires even more gates, namely the ones to map the nuclear spin to the electron spin (the RETRIEVE operation, see section 8.3.2 of the Methods), where the correction operator could be applied.

In what follows, we first explain how we track the correction operations. Then, we describe how the tracking changes the protocol building blocks from section 8.3.2 of the Methods and subsequently prove the correctness of the tracking in the ideal case.

Let us denote the four Bell states as

$$|\phi[\pm 1, 1]\rangle = (|00\rangle \pm |11\rangle)/\sqrt{2},$$
$$|\phi[\pm 1, -1]\rangle = (|01\rangle \pm |10\rangle)/\sqrt{2}.$$

To each of the qubits it holds, a node associates a single-qubit Pauli operator $\mathbb{1}, X, Y$ or Z, which are defined as

$$\mathbb{1} = |0\rangle\langle 0| + |1\rangle\langle 1|, Z = |0\rangle\langle 0| - |1\rangle\langle 1|, X = |0\rangle\langle 1| + |1\rangle\langle 0|, Y = -i\,|0\rangle\langle 1| + i\,|1\rangle\langle 0|.$$

The goal of the tracking is, at any time during the simulation, for any two nodes A and B sharing electron-electron entanglement, that the target electron-electron state equals

$$|\psi\rangle \equiv (P_A \otimes P_B)\,|\phi[1, -1]\rangle\,. \tag{8.16}$$

Here, P_A and P_B denote the Pauli correction operations of node A or B, respectively, and $\equiv$ denotes equality modulo a complex number of norm 1.

In what follows, it will be more convenient to use the following equivalent statement to eq. (8.16):

$$(P_A \otimes P_B)\,|\psi\rangle \equiv |\phi[1, -1]\rangle\,. \tag{8.17}$$

TRACKING CORRECTION OPERATORS DURING THE NV REPEATER CHAIN PROTOCOL

Here, we explain how each of the four protocol building blocks from section 8.3.2 of the Methods are adjusted to ensure that eq. (8.17) holds after the operations ENTGEN, DISTILL and SWAP.

Entanglement generation. Suppose that nodes A and B perform the ENTGEN protocol. In the absence of noise, this protocol (approximately) produces the state $|\phi[\pm 1,-1]\rangle$, where $\pm$ denotes which detector clicked (Methods, section 8.3.2). If the $+$-detector clicked, then the state that A and B hold is the desired state $|\phi[1,-1]\rangle$, so we set $P_A = P_B = \mathbb{1}$. If the other detector clicked, then the produced state is $|\phi[-1,-1]\rangle$. Therefore, one of the nodes (for example, the one with the higher position index in the chain) sets the correction operator to Z, whereas the other sets it to $\mathbb{1}$, since $(\mathbb{1}\otimes Z)\,|\phi[-1,-1]\rangle = |\phi[1,-1]\rangle$.

Storing and retrieving qubits. Locally mapping the state of a qubit onto a different memory position by the STORE or RETRIEVE circuits does not alter the correction Pauli corresponding to that qubit.

Entanglement distillation. Suppose that nodes A and B wish to perform the DISTILL protocol, which starts by A and B sharing an electron-electron pair (correction Paulis P_A^e and P_B^e at node A and B, respectively) and a nuclear-nuclear pair (P_A^n and P_B^n). In the protocol, first both nodes apply $P^n \cdot P^e$ to their electron spin qubit. Then, both nodes locally perform the distillation circuit from Supplementary Figure 8.16(c), followed by sending both the measurement outcome and P^n to the other node. The nodes determine whether the distillation succeeded using the condition explained in section 8.6.5. In case of failure, the nuclear-nuclear state is discarded. In case of success, one of the nodes in the chain (for example, the one with the lower position index in the chain) sets $P^n = \mathbb{1}$, while the other sets

$$P^n = \begin{cases} Y & \text{if } P_A^n \in \{X,Y\} \text{ and } P_B^n \in \{\mathbb{1},Z\} \\ Y & \text{if } P_A^n \in \{\mathbb{1},Z\} \text{ and } P_B^n \in \{X,Y\} \\ \mathbb{1} & \text{otherwise.} \end{cases} \tag{8.18}$$

Below, in section 8.6.6 of this Supplementary Note, we show that after this procedure, eq. (8.17) still holds.

Entanglement swapping. Suppose that node M wants to execute the SWAP protocol on shared pairs $A-M$ and $M-B$, with nodes A and B respectively. We denote M's correction Paulis as P_M^A and P_M^B. First, M performs the Bell-state measurement circuit from Supplementary Figure 8.16(d). Let us denote the individual measurement outcomes of the circuit as m_{earlier} and m_{later} (both take values from $\{1,-1\}$), which correspond to the measured Bell state $|\phi[a,b]\rangle$ with $a = -1\cdot m_{\text{earlier}}\cdot m_{\text{later}}$ and $b = m_{\text{later}}$. Then, M sends

the Pauli $\mathbb{1}$ to A, while to B it sends $P_M^A \cdot P_M^B \cdot Q$, where Q is given by

(a,b)	Q
$(1,1)$	X
$(1,-1)$	$\mathbb{1}$
$(-1,1)$	Y
$(-1,-1)$	Z

(8.19)

Both nodes A and B multiply their local Pauli with the Pauli they received from M. The proof that after the swap, eq. (8.17) still holds can be found below in section 8.6.6 of this Supplementary Note.

CORRECTNESS PROOF OF THE CORRECTION OPERATOR UPDATE FOR DISTILL

Here, we prove that eq. (8.17) holds for the states that are outputted by the protocols for entanglement distillation and swapping explained above.

Let us start with entanglement distillation. For this, we denote by 'physical nuclear-nuclear state' the joint state of the nuclear spins of node A and B. By direct computation, one can show the following.

Proposition 8. *Suppose that nodes A and B share the state $|\phi[a,b]\rangle$ on the electrons and the physical nuclear-nuclear state $|\phi[c,d]\rangle$, where $a,b,c,d \in \{1,-1\}$. When both nodes execute the distillation circuit from Supplementary Figure 8.16(c), the resulting state on the carbon nuclear spins is*

$$|\phi[c,-a\cdot c\cdot d]\rangle$$

and the measurement outcome $m_1 \in \{1,-1\}$ on one side is uniformly random, while the outcome of the other node equals $m_2 = m_1 \cdot b \cdot c$.

We emphasise that using the correction-operator tracking for the STORE and RETRIEVE operations as described in section 8.6.6 of this Supplementary Note, the physical nuclear-nuclear state between any two nodes does not satisfy eq. (8.17). The reason for this is that the STORE operation maps the electron spin state to the nuclear spin in a rotated basis, where the rotation operator is a Hadamard gate H (eq. 8.14). However, the correction operators are not updated when the STORE is applied (see 'Storing and retrieving qubits' in section 8.6.6). Consequently, if nodes A and B share the physical nuclear-nuclear state $|\psi\rangle$, then mapping $|\psi\rangle$ to the reference Bell state $|\phi[1,-1]\rangle$ requires first the application of $H\otimes H$, followed by applying $P_A \otimes P_B$. By 'virtual nuclear-nuclear state', we mean the state $|\psi'\rangle = (H\otimes H)|\psi\rangle$, i.e. the state that satisfies eq. (8.17). Let us first convert Prop. 8 to a statement with the virtual-virtual nuclear state.

Proposition 9. *Suppose nodes A and B share the electron-electron state $|\phi[a,b]\rangle$ and the virtual nuclear-nuclear state $|\phi[c,d]\rangle$. Then after the distillation circuit from Supplementary Figure 8.16(c), the virtual state on the nuclear spins after performing the distillation equals*

$$|\phi[-a\cdot c\cdot d, d]\rangle$$

and the measurement outcomes are $m_1 \in \{1,-1\}$ (uniformly random) and $m_2 = m_1 \cdot b \cdot d$.

Proof. The virtual nuclear-nuclear state and the physical one are related by $H \otimes H$. It is not hard to see that $H \otimes H |\phi[x,y]\rangle = |\phi[y,x]\rangle$ for any $x, y \in \{1,-1\}$. Applying this to Prop. 8 results in the measurement outcomes m_1 (uniformly random) and $m_2 = m_1 \cdot b \cdot d$ and resulting physical nuclear-nuclear state $|\phi[d, -acd]\rangle$. Obtaining the virtual state is done by applying $H \otimes H$ again, which yields $|\phi[-acd, d]\rangle$. □

Using Prop. 9, it is straightforward to check that the output state of the distillation protocol from section 8.6.6 satisfies eq. (8.17).

Suppose A and B share the electron-electron state $|\phi[a,b]\rangle$ and the virtual nuclear-nuclear state $|\phi[c,d]\rangle$ for some $a,b,c,d \in \{1,-1\}$, with correction Paulis P^e_A (P^e_B) and P^n_A (P^n_B) for A (B). In the first step of the protocol, A and B apply $P^n \cdot P^e$ to the electron-electron state, resulting in the electron-electron state

$$(P^n_A P^e_A \otimes P^n_B P^e_B) |\phi[a,b]\rangle = (P^n_A P^e_A \otimes P^n_B P^e_B)(P^e_A \otimes P^e_B) |\phi[1,-1]\rangle = (P^n_A \otimes P^n_B) |\phi[1,-1]\rangle = |\phi[c,d]\rangle$$

where we made use of the fact that each Pauli squares to $\mathbb{1}$. In case of successful distillation, the virtual nuclear-nuclear state can be found using Prop. 9 and equals $|\phi[-ccd, d]\rangle = |\phi[-d,d]\rangle$. What remains is to determine the correction operators conditioned on the value of d. If $d = 1$, then the correction operators are $\mathbb{1}$ for one node and Y for the other (since $\mathbb{1} \otimes Y |\phi[-1,-1]\rangle$ equals the target Bell state $|\phi[1,-1]\rangle$), while for $d = -1$ the resulting state is already the target Bell state and both correction operators should be $\mathbb{1}$. Determining the value of d can be done by using the fact that eq. (8.17) was satisfied by the pre-distillation virtual nuclear-nuclear state,

$$(P^n_A \otimes P^n_B) |\phi[c,d]\rangle = |\phi[1,-1]\rangle$$

and thus $|\phi[c,d]\rangle = P^n_A \otimes P^n_B |\phi[1,-1]\rangle$. From checking all possible cases of P^n_A and P^n_B we find that $d = 1$ precisely if one of P^n_A, P^n_B equals X or Y, while the other equals $\mathbb{1}$ or Z.

CORRECTNESS PROOF OF THE CORRECTION OPERATOR UPDATE FOR SWAP

Here we show that eq. (8.17) holds for the state between nodes A and B after node M has performed an entanglement swap on Bell states $A-M$ and $M-B$. Let us denote A's (B's) correction operator as P_A (P_B) and M's correction operator as P^A_M (P^B_M) for the state it shares with node A (B). That is, in the ideal case, the nodes hold the state

$$(P_A \otimes P^A_M \otimes P^B_M \otimes P_B)(|\phi[1,-1]\rangle_{AM} \otimes |\phi[1,-1]\rangle_{MB}). \tag{8.20}$$

We will make use of the fact that

$$(P \otimes Q) |\phi[a,b]\rangle \equiv (\mathbb{1} \otimes PQ) |\phi[a,b]\rangle \tag{8.21}$$

for single-qubit Paulis P, Q and $a, b \in \{1,-1\}$, where $\equiv$ as before indicates that the two states differ only by a complex factor of norm 1 (in fact, for eq.(8.21) we can restrict this to a multiplicative factor ± 1). Using eq. (8.21), we rewrite eq. (8.20) to

$$(P^A_M P_A \otimes \mathbb{1} \otimes \mathbb{1} \otimes P^B_M P_B)(|\phi[1,-1]\rangle_{AM} \otimes |\phi[1,-1]\rangle_{MB}). \tag{8.22}$$

Eq. (8.22) implies that we may assume that M's two correction operators are both $\mathbb{1}$. Thus M only needs to communicate the correction operator that corresponds to having

measured one qubit of each pair of the pair $|\phi[1,-1]\rangle \otimes |\phi[1,-1]\rangle$. The resulting correction operator Q can be straightforwardly worked out in a similar way as in [102] and the result is given in 8.19.

The state after the entanglement swap is thus

$$(P_M^A P_A \otimes P_M^B P_B Q)\,|\phi[1,-1]\rangle_{AB}$$

which we rewrite using eq. (8.21) to

$$(P_A \otimes P_M^A P_M^B P_B Q)\,|\phi[1,-1]\rangle_{AB}\,.$$

Indeed, P_A and $P_M^A P_M^B P_B Q$ are (modulo possible factor -1) the correction operators of node A and B, respectively, after finishing the entanglement swapping protocol described above in section 8.6.6 of this Supplementary Note.

What remains is to convert the measurement outcomes from the circuit from Supplementary Figure 8.16(d) to the measured Bell state. For this, a direct computation shows that applying the circuit to the electron-nuclear state $(\mathbb{1}_e \otimes H_n)\,|\phi[a,b]\rangle$ (the Hadamard gate H is needed since the nuclear qubit lives in a rotated basis, see section 8.6.5) yields the measurement outcomes $m_{\text{earlier}} = -ab$ and $m_{\text{later}} = b$. Rewriting gives $a = -m_{\text{earlier}} m_{\text{later}}$ and $b = m_{\text{later}}$.

REFERENCES

[1] W. J. Munro, K. Azuma, K. Tamaki, and K. Nemoto, *Inside quantum repeaters,* IEEE Journal of Selected Topics in Quantum Electronics **21**, 78 (2015).

[2] S. Muralidharan, L. Li, J. Kim, N. Lütkenhaus, M. D. Lukin, and L. Jiang, *Optimal architectures for long distance quantum communication,* Scientific Reports **6**, 20463 EP (2016), article.

[3] N. Gisin and R. Thew, *Quantum communication,* Nature Photonics **1**, 165 EP (2007), review Article.

[4] H.-J. Briegel, W. Dür, J. I. Cirac, and P. Zoller, *Quantum repeaters: The role of imperfect local operations in quantum communication,* Phys. Rev. Lett. **81**, 5932 (1998).

[5] W. Dür, H.-J. Briegel, J. I. Cirac, and P. Zoller, *Quantum repeaters based on entanglement purification,* Phys. Rev. A **59**, 169 (1999).

[6] L.-M. Duan, M. D. Lukin, J. I. Cirac, and P. Zoller, *Long-distance quantum communication with atomic ensembles and linear optics,* Nature **414**, 413 EP (2001), article.

[7] J. Amirloo, M. Razavi, and A. H. Majedi, *Quantum key distribution over probabilistic quantum repeaters,* Phys. Rev. A **82**, 032304 (2010).

[8] F. Kimiaee Asadi, N. Lauk, S. Wein, N. Sinclair, C. O'Brien, and C. Simon, *Quantum repeaters with individual rare-earth ions at telecommunication wavelengths,* Quantum **2**, 93 (2018).

[9] N. K. Bernardes, L. Praxmeyer, and P. van Loock, *Rate analysis for a hybrid quantum repeater,* Phys. Rev. A **83**, 012323 (2011).

[10] J. Borregaard, P. Kómár, E. M. Kessler, A. S. Sørensen, and M. D. Lukin, *Heralded quantum gates with integrated error detection in optical cavities,* Phys. Rev. Lett. **114**, 110502 (2015).

[11] D. E. Bruschi, T. M. Barlow, M. Razavi, and A. Beige, *Repeat-until-success quantum repeaters,* Phys. Rev. A **90**, 032306 (2014).

[12] Z.-B. Chen, B. Zhao, Y.-A. Chen, J. Schmiedmayer, and J.-W. Pan, *Fault-tolerant quantum repeater with atomic ensembles and linear optics,* Phys. Rev. A **76**, 022329 (2007).

[13] O. A. Collins, S. D. Jenkins, A. Kuzmich, and T. A. B. Kennedy, *Multiplexed memory-insensitive quantum repeaters,* Phys. Rev. Lett. **98**, 060502 (2007).

[14] S. Guha, H. Krovi, C. A. Fuchs, Z. Dutton, J. A. Slater, C. Simon, and W. Tittel, *Rate-loss analysis of an efficient quantum repeater architecture,* Phys. Rev. A **92**, 022357 (2015).

[15] L. Hartmann, B. Kraus, H.-J. Briegel, and W. Dür, *Role of memory errors in quantum repeaters,* Phys. Rev. A **75**, 032310 (2007).

[16] L. Jiang, J. M. Taylor, K. Nemoto, W. J. Munro, R. Van Meter, and M. D. Lukin, *Quantum repeater with encoding,* Phys. Rev. A **79**, 032325 (2009).

[17] K. Nemoto, M. Trupke, S. J. Devitt, B. Scharfenberger, K. Buczak, J. Schmiedmayer, and W. J. Munro, *Photonic quantum networks formed from NV- centers,* Scientific Reports **6**, 26284 EP (2016), article.

[18] M. Razavi, M. Piani, and N. Lütkenhaus, *Quantum repeaters with imperfect memories: Cost and scalability,* Phys. Rev. A **80**, 032301 (2009).

[19] M. Razavi and J. H. Shapiro, *Long-distance quantum communication with neutral atoms,* Phys. Rev. A **73**, 042303 (2006).

[20] C. Simon, H. de Riedmatten, M. Afzelius, N. Sangouard, H. Zbinden, and N. Gisin, *Quantum repeaters with photon pair sources and multimode memories,* Phys. Rev. Lett. **98**, 190503 (2007).

[21] S. E. Vinay and P. Kok, *Practical repeaters for ultralong-distance quantum communication,* Phys. Rev. A **95**, 052336 (2017).

[22] Y. Wu, J. Liu, and C. Simon, *Near-term performance of quantum repeaters with imperfect ensemble-based quantum memories,* Phys. Rev. A **101**, 042301 (2020).

[23] N. Sangouard, C. Simon, J. c. v. Minář, H. Zbinden, H. de Riedmatten, and N. Gisin, *Long-distance entanglement distribution with single-photon sources,* Phys. Rev. A **76**, 050301 (2007).

[24] N. Sangouard, C. Simon, B. Zhao, Y.-A. Chen, H. de Riedmatten, J.-W. Pan, and N. Gisin, *Robust and efficient quantum repeaters with atomic ensembles and linear optics,* Phys. Rev. A **77**, 062301 (2008).

[25] N. Sangouard, R. Dubessy, and C. Simon, *Quantum repeaters based on single trapped ions,* Phys. Rev. A **79**, 042340 (2009).

[26] S. Abruzzo, S. Bratzik, N. K. Bernardes, H. Kampermann, P. van Loock, and D. Bruß, *Quantum repeaters and quantum key distribution: Analysis of secret-key rates,* Phys. Rev. A **87**, 052315 (2013).

[27] J. B. Brask and A. S. Sørensen, *Memory imperfections in atomic-ensemble-based quantum repeaters,* Phys. Rev. A **78**, 012350 (2008).

[28] S. Muralidharan, J. Kim, N. Lütkenhaus, M. D. Lukin, and L. Jiang, *Ultrafast and fault-tolerant quantum communication across long distances,* Phys. Rev. Lett. **112**, 250501 (2014).

[29] M. Pant, H. Krovi, D. Englund, and S. Guha, *Rate-distance tradeoff and resource costs for all-optical quantum repeaters,* Phys. Rev. A **95**, 012304 (2017).

[30] T. D. Ladd, P. van Loock, K. Nemoto, W. J. Munro, and Y. Yamamoto, *Hybrid quantum repeater based on dispersive CQED interactions between matter qubits and bright coherent light,* New Journal of Physics **8**, 184 (2006).

[31] P. van Loock, T. D. Ladd, K. Sanaka, F. Yamaguchi, K. Nemoto, W. J. Munro, and Y. Yamamoto, *Hybrid quantum repeater using bright coherent light,* Phys. Rev. Lett. **96**, 240501 (2006).

[32] M. Zwerger, B. Lanyon, T. Northup, C. Muschik, W. Dür, and N. Sangouard, *Quantum repeaters based on trapped ions with decoherence-free subspace encoding,* Quantum Science and Technology **2**, 044001 (2017).

[33] L. Jiang, J. M. Taylor, and M. D. Lukin, *Fast and robust approach to long-distance quantum communication with atomic ensembles,* Phys. Rev. A **76**, 012301 (2007).

[34] A. Dahlberg, M. Skrzypczyk, T. Coopmans, L. Wubben, F. Rozpędek, M. Pompili, A. Stolk, P. Pawełczak, R. Knegjens, J. de Oliveira Filho, R. Hanson, and S. Wehner, *A link layer protocol for quantum networks,* in *Proceedings of the ACM Special Interest Group on Data Communication*, SIGCOMM '19 (Association for Computing Machinery, New York, NY, USA, 2019) pp. 159–173.

[35] R. V. Meter, *Quantum networking and internetworking,* IEEE Network **26**, 59 (2012).

[36] L. Aparicio, R. Van Meter, and H. Esaki, *Protocol design for quantum repeater networks,* in *Proceedings of the 7th Asian Internet Engineering Conference*, AINTEC '11 (Association for Computing Machinery, New York, NY, USA, 2011) pp. 73–80.

[37] R. V. Meter and J. Touch, *Designing quantum repeater networks,* IEEE Communications Magazine **51**, 64 (2013).

[38] R. V. Meter, T. D. Ladd, W. J. Munro, and K. Nemoto, *System design for a long-line quantum repeater,* IEEE/ACM Transactions on Networking **17**, 1002 (2009).

[39] A. Pirker and W. Dür, *A quantum network stack and protocols for reliable entanglement-based networks,* New Journal of Physics **21**, 033003 (2019).

[40] A. Acín, N. Brunner, N. Gisin, S. Massar, S. Pironio, and V. Scarani, *Device-independent security of quantum cryptography against collective attacks,* Phys. Rev. Lett. **98**, 230501 (2007).

[41] C. Branciard, E. G. Cavalcanti, S. P. Walborn, V. Scarani, and H. M. Wiseman, *One-sided device-independent quantum key distribution: Security, feasibility, and the connection with steering,* Phys. Rev. A **85**, 010301 (2012).

[42] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, *The security of practical quantum key distribution,* Rev. Mod. Phys. **81**, 1301 (2009).

[43] F. Xu, X. Ma, Q. Zhang, H.-K. Lo, and J.-W. Pan, *Secure quantum key distribution with realistic devices,* Rev. Mod. Phys. **92**, 025002 (2020).

[44] S. Pirandola, U. L. Andersen, L. Banchi, M. Berta, D. Bunandar, R. Colbeck, D. Englund, T. Gehring, C. Lupo, C. Ottaviani, *et al.*, *Advances in quantum cryptography,* Advances in Optics and Photonics **12**, 1012 (2020).

[45] S. Barz, E. Kashefi, A. Broadbent, J. F. Fitzsimons, A. Zeilinger, and P. Walther, *Demonstration of blind quantum computing,* Science **335**, 303 (2012), https://science.sciencemag.org/content/335/6066/303.full.pdf .

[46] N. H. Nickerson, J. F. Fitzsimons, and S. C. Benjamin, *Freely scalable quantum technologies using cells of 5-to-50 qubits with very lossy and noisy photonic links,* Phys. Rev. X **4**, 041041 (2014).

[47] V. Lipinska, G. Murta, and S. Wehner, *Anonymous transmission in a noisy quantum network using the w state,* Phys. Rev. A **98**, 052320 (2018).

[48] E. T. Khabiboulline, J. Borregaard, K. De Greve, and M. D. Lukin, *Optical interferometry with quantum networks,* Phys. Rev. Lett. **123**, 070504 (2019).

[49] S. Wehner, D. Elkouss, and R. Hanson, *Quantum internet: A vision for the road ahead,* Science **362** (2018), 10.1126/science.aam9288, https://science.sciencemag.org/content/362/6412/eaam9288.full.pdf .

[50] E. Shchukin, F. Schmidt, and P. van Loock, *Waiting time in quantum repeaters with probabilistic entanglement swapping,* Phys. Rev. A **100**, 032322 (2019).

[51] S. E. Vinay and P. Kok, *Statistical analysis of quantum-entangled-network generation,* Phys. Rev. A **99**, 042313 (2019).

[52] G. Vardoyan, S. Guha, P. Nain, and D. Towsley, *On the stochastic analysis of a quantum entanglement switch,* SIGMETRICS Perform. Eval. Rev. **47**, 27 (2019).

[53] M. Razavi, K. Thompson, H. Farmanbar, M. Piani, and N. Lütkenhaus, *Physical and architectural considerations in quantum repeaters,* in *Quantum Communications Realized II*, Vol. 7236, edited by Y. Arakawa, M. Sasaki, and H. Sotobayashi, International Society for Optics and Photonics (SPIE, 2009) pp. 18 – 30.

[54] M. M. Wilde, *Quantum information theory* (Cambridge University Press, 2013).

[55] M. Pant, H. Krovi, D. Towsley, L. Tassiulas, L. Jiang, P. Basu, D. Englund, and S. Guha, *Routing entanglement in the quantum internet,* npj Quantum Information **5**, 25 (2019).

[56] V. V. Kuzmin, D. V. Vasilyev, N. Sangouard, W. Dür, and C. A. Muschik, *Scalable repeater architectures for multi-party states,* npj Quantum Information **5**, 115 (2019).

[57] S. Khatri, C. T. Matyas, A. U. Siddiqui, and J. P. Dowling, *Practical figures of merit and thresholds for entanglement distribution in quantum networks,* Phys. Rev. Research **1**, 023032 (2019).

[58] A. Varga, *The OMNeT++ discrete event simulation system,* in *Proc. of the European Simulation Multiconference (ESM'2001)* (2001).

[59] G. F. Riley and T. R. Henderson, *The ns-3 network simulator,* in *Modeling and Tools for Network Simulation*, edited by K. Wehrle, M. Güneş, and J. Gross (Springer Berlin Heidelberg, Berlin, Heidelberg, 2010) pp. 15–34.

[60] B. Lantz, B. Heller, and N. McKeown, *A network in a laptop: rapid prototyping for software-defined networks,* in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks* (2010) pp. 1–6.

[61] M. Fingerhuth, T. Babej, and P. Wittek, *Open source software in quantum computing,* PLOS ONE **13**, e0208561 (2018).

[62] *Netsquid website and online documentation,* `https://netsquid.org`, access to documentation requires registration.

[63] D. Deutsch, A. Ekert, R. Jozsa, C. Macchiavello, S. Popescu, and A. Sanpera, *Quantum privacy amplification and the security of quantum cryptography over noisy channels,* Phys. Rev. Lett. **77**, 2818 (1996).

[64] K. Wehrle, M. Günes, and J. Gross, *Modeling and tools for network simulation* (Springer Science & Business Media, 2010).

[65] D. M. Greenberger, M. A. Horne, and A. Zeilinger, *Going beyond Bell's theorem,* in *Bell's theorem, quantum theory and conceptions of the universe* (Springer, 1989) pp. 69–72.

[66] D. D. Awschalom, R. Hanson, J. Wrachtrup, and B. B. Zhou, *Quantum technologies with optically interfaced solid-state spins,* Nature Photonics **12**, 516 (2018).

[67] M. W. Doherty, N. B. Manson, P. Delaney, F. Jelezko, J. Wrachtrup, and L. C. Hollenberg, *The nitrogen-vacancy colour centre in diamond,* Physics Reports **528**, 1 (2013).

[68] S. Aaronson and D. Gottesman, *Improved simulation of stabilizer circuits,* Physical Review A **70**, 052328 (2004).

[69] S. Anders and H. J. Briegel, *Fast simulation of stabilizer circuits using a graph-state representation,* Physical Review A **73**, 022334 (2006).

[70] D. S. Steiger, T. Häner, and M. Troyer, *ProjectQ: an open source software framework for quantum computing,* Quantum **2**, 49 (2018).

[71] J. de Oliveira Filho, Z. Papp, R. Djapic, and J. Oosteveen, *Model-based design of self-adapting networked signal processing systems,* in *2013 IEEE 7th International Conference on Self-Adaptive and Self-Organizing Systems* (IEEE, 2013) pp. 41–50.

[72] A. Dahlberg and S. Wehner, *SimulaQron – a simulator for developing quantum internet software,* Quantum Science and Technology **4**, 015001 (2018).

[73] S. DiAdamo, J. Nötzel, B. Zanger, and M. M. Beşe, *QuNetSim: A software framework for quantum networks,* arXiv:2003.06397 (2020).

[74] B. Bartlett, *A distributed simulation framework for quantum networks and channels,* arXiv:quant-ph/1808.07047 (2018).

[75] T. Matsuo, *Simulation of a dynamic, RuleSet-based quantum network,* arXiv:1908.10758 (2020).

[76] L. O. Mailloux, J. D. Morris, M. R. Grimaila, D. D. Hodson, D. R. Jacques, J. M. Colombi, C. V. Mclaughlin, and J. A. Holes, *A modeling framework for studying quantum key distribution system implementation nonidealities,* IEEE Access **3**, 110 (2015).

[77] X. Wu, A. Kolar, J. Chung, D. Jin, T. Zhong, R. Kettimuthu, and M. Suchara, *SeQUeNCe: A customizable discrete-event simulator of quantum networks,* arXiv:2009.12000 (2020).

[78] Y. Lee, E. Bersin, A. Dahlberg, S. Wehner, and D. Englund, *A quantum router architecture for high-fidelity entanglement flows in multi-user quantum networks,* arXiv:2005.01852 (2020).

[79] W. Kozlowski, A. Dahlberg, and S. Wehner, *Designing a quantum network protocol,* in *Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '20)* (ACM, 2020) p. 16.

[80] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, and K. Smith, *Cython: The best of both worlds,* Computing in Science & Engineering **13**, 31 (2011).

[81] K. De Raedt, K. Michielsen, H. De Raedt, B. Trieu, G. Arnold, M. Richter, T. Lippert, H. Watanabe, and N. Ito, *Massively parallel quantum computer simulator,* Computer Physics Communications **176**, 121 (2007).

[82] T. Häner and D. S. Steiger, *0.5 petabyte simulation of a 45-qubit quantum circuit,* in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '17 (Association for Computing Machinery, New York, NY, USA, 2017).

[83] F. Rozpędek, R. Yehia, K. Goodenough, M. Ruf, P. C. Humphreys, R. Hanson, S. Wehner, and D. Elkouss, *Near-term quantum-repeater experiments with nitrogen-vacancy centers: Overcoming the limitations of direct transmission,* Phys. Rev. A **99**, 052330 (2019).

[84] P. C. Humphreys, N. Kalb, J. P. J. Morits, R. N. Schouten, R. F. L. Vermeulen, D. J. Twitchen, M. Markham, and R. Hanson, *Deterministic delivery of remote entanglement on a quantum network,* Nature **558**, 268 (2018).

[85] T. Coopmans, R. Knegjens, A. Dahlberg, D. Maier, L. Nijsten, J. Oliveira, M. Papendrecht, J. Rabbie, F. Rozpędek, M. Skrzypczyk, L. Wubben, W. de Jong, D. Podareanu, A. Torres Knoop, D. Elkouss, and S. Wehner, *Replication Data for: NetSquid, a discrete-event simulation platform for quantum networks,* (2021).

[86] T. Coopmans, R. Knegjens, A. Dahlberg, D. Maier, L. Nijsten, J. Oliveira, M. Papendrecht, J. Rabbie, F. Rozpędek, M. Skrzypczyk, L. Wubben, W. de Jong, D. Podareanu, A. Torres Knoop, D. Elkouss, and S. Wehner, *Simulation Code for: NetSquid, a discrete-event simulation platform for quantum networks,* (2021).

[87] D. Gottesman, *The Heisenberg representation of quantum computers,* arXiv:quant-ph/9807006v1 (1998).

[88] M. A. Nielsen and I. L. Chuang, *Quantum information and quantum computation,* Cambridge: Cambridge University Press **2**, 23 (2000).

[89] M. Hein, W. Dür, J. Eisert, R. Raussendorf, M. Nest, and H.-J. Briegel, *Entanglement in graph states and its applications,* arXiv:0602096 (2006).

[90] S. Hermans, Personal communication (2020).

[91] R. Paschotta, *'Refractive Index' in RP Photonics Encyclopedia,* (2020).

[92] D. Riedel, I. Söllner, B. J. Shields, S. Starosielec, P. Appel, E. Neu, P. Maletinsky, and R. J. Warburton, *Deterministic enhancement of coherent photon generation from a nitrogen-vacancy center in ultrapure diamond,* Phys. Rev. X **7**, 031040 (2017).

[93] B. Hensen, H. Bernien, A. E. Dréau, A. Reiserer, N. Kalb, M. S. Blok, J. Ruitenberg, R. F. L. Vermeulen, R. N. Schouten, C. Abellán, W. Amaya, V. Pruneri, M. W. Mitchell, M. Markham, D. J. Twitchen, D. Elkouss, S. Wehner, T. H. Taminiau, and R. Hanson, *Loophole-free bell inequality violation using electron spins separated by 1.3 kilometres,* Nature **526**, 682 EP (2015).

[94] S. Zaske, A. Lenhard, C. A. Keßler, J. Kettler, C. Hepp, C. Arend, R. Albrecht, W.-M. Schulz, M. Jetter, P. Michler, and C. Becher, *Visible-to-telecom quantum frequency conversion of light from a single quantum emitter,* Phys. Rev. Lett. **109**, 147404 (2012).

[95] N. Kalb, A. A. Reiserer, P. C. Humphreys, J. J. W. Bakermans, S. J. Kamerling, N. H. Nickerson, S. C. Benjamin, D. J. Twitchen, M. Markham, and R. Hanson, *Entanglement distillation between solid-state quantum network nodes,* Science **356**, 928 (2017).

[96] N. Kalb, P. C. Humphreys, J. J. Slim, and R. Hanson, *Dephasing mechanisms of diamond-based nuclear-spin memories for quantum networks,* Phys. Rev. A **97**, 062330 (2018).

[97] H. Beukers, *Improving coherence of quantum memory during entanglement creation between nitrogen vacancy centres in diamond (master thesis),* (2019).

[98] M. H. Abobeih, J. Cramer, M. A. Bakker, N. Kalb, M. Markham, D. J. Twitchen, and T. H. Taminiau, *One-second coherence for a single electron spin coupled to a multi-qubit nuclear-spin environment,* Nature Communications **9**, 2552 (2018).

[99] C. E. Bradley, J. Randall, M. H. Abobeih, R. C. Berrevoets, M. J. Degen, M. A. Bakker, M. Markham, D. J. Twitchen, and T. H. Taminiau, *A ten-qubit solid-state spin register with quantum memory up to one minute,* Phys. Rev. X **9**, 031045 (2019).

[100] A. Reiserer, N. Kalb, M. S. Blok, K. J. M. van Bemmelen, T. H. Taminiau, R. Hanson, D. J. Twitchen, and M. Markham, *Robust quantum-network memory using decoherence-protected subspaces of nuclear spins,* Phys. Rev. X **6**, 021040 (2016).

[101] T. H. Taminiau, J. Cramer, T. van der Sar, V. V. Dobrovitski, and R. Hanson, *Universal control and error correction in multi-qubit spin registers in diamond,* Nature Nanotechnology **9**, 171 (2014).

[102] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, *Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels,* Phys. Rev. Lett. **70**, 1895 (1993).

9

CONCLUSION

9.1. SUMMARY OF RESULTS

In this thesis, we focused on (tools for) the analysis and optimisation of quantum repeater protocols. In the first part of the thesis, we considered an abstract hardware model and did the following:

- we developed efficient algorithms for characterising the completion time of `tree-shaped`-type quantum repeater schemes, and the fidelity of the entangled state that it produces. The runtimes of these algorithms is polynomial in the number of repeater nodes, which is an exponential improvement on existing algorithms;

- we used one of the two algorithms to predict the performance increase when adding cut-offs, e.g. the maximum storage time for entanglement, to `tree-shaped`-type repeater schemes, and showed that in some hardware parameter regimes, the use of cut-off is necessary for secret-key generation between remote parties. We also showed that our algorithm is fast enough to optimise the cut-off for maximal secret key rate;

- using a novel connection to reliability theory, we established analytical bounds on the completion time of `tree-shaped`-type quantum repeater schemes in case the success probability of all repeater building blocks is lower bounded by a constant. These bounds improve significantly on existing analytical work in some cases. In particular, for a famous nested repeater scheme [1, 2], we showed that the frequently used 3-over-2 approximation [3, 4] to the scheme's mean completion time is, in essence, an upper bound.

In the second part, we introduced the NetSquid simulator for quantum networks and used it to

- assess the performance of quantum repeater schemes on nitrogen-vacancy centres in diamond, a promising hardware platform for quantum networks. We considered both state-of-the-art as well as improved hardware parameters. We also found by how much the various noisy parts of the hardware (such as detection probability or induced storage qubit noise) should be improved to reach a fidelity beyond the classical bound;
- simulate a quantum switch to find its performance if the quantum memories are noisy and also limited in number.

9.2. FUTURE WORK

We discuss a few avenues for future research to extend the results from Chapters 5-8:

- **Analytical bounds on the fidelity of entanglement from quantum repeater chains.**
  In Chapters 5 and 6, we provided a deterministic algorithm for computing the probability distribution of the completion time (waiting time) of `tree-shaped`-type quantum repeater protocols and the average fidelity of the entanglement they produce. The algorithm's runtime is a function of the maximum waiting time that we are interested in (to be precise: the time at which we truncate the probability distribution; ideally, we have captured a close-to-100% of the probability mass at the truncation time). Consequently, the algorithm is fast for assessing high-quality hardware (specifically, for high success probabilities of the components such as entanglement swapping), in which case the truncation time can be chosen relatively small, but its runtime diverges in the limit of small success probabilities.

  In contrast, in Chapter 7, we provided analytical bounds on the completion time which become exact in the same limit. For this reason, converting the analytical completion time bounds to bounds on fidelity will yield asymptotically exact bounds on fidelity. Another argument in favour of studying the low-probability-range is that state-of-the-art hardware operates in this regime.

- **Extension of the analytical and semi-analytical tools to multiple entangled pairs.**
  Chapters 5-7 focused on the delivery time of the first end-to-end entangled pair of qubits, delivered by a `tree-shaped`-type quantum repeater chain. Our analysis relies on the fact that the `tree-shaped`-type protocols are nested; at each nesting level the nodes deliver a single pair of qubits, after which they remain idle until the higher level requests another pair. This setup is convenient since we could analyse it using a divide-and-conquer algorithm as we have seen in chapters 5-6.

  In order to increase the entanglement delivery rate, we would like the nodes not to remain idle in between requests. To achieve this, we consider a modification of the protocol in which each subprotocol does not wait for requests from higher levels, but keeps producing entanglement continuously. This is advantageous for two reasons. First, in case the higher-level-entanglement is lost, e.g. due to a failing entanglement swap, the higher level will ask the lower levels for delivering entanglement, who will have a head start in doing so. Next, because of this head start,

the production of the *second*, third, fourth, etc. pair will take less additional time than the first.

The modified protocol is more complex to analyse, since now the time between a higher-level request for an entangled pair and its delivery is dependent on the time at which the request is issued. Hence, analysis of the modified protocol is not possible directly with the numerical tools similar to the ones from Chapters 5 and 6.

The analytical approach from Chapter 7, however, might be more easily extendable. Our results in this chapter are based on the fact that the random variable which describes the delivery time of the first entangled pair, possesses the new-better-than-used property. We might be able to use existing results that the same property holds for so-called 'order statistics', i.e. the distributions of the second smallest, third smallest, etc. of a set of samples drawn from the same distribution [5].

An alternative approach would be to use queuing theory to find the average rate of quantum repeater chain protocols which continuously produce entanglement, following the quantum-switch analysis by Vardoyan et al. [6].

- **Integrate the tools as part of a routing algorithm to decide the cost of a path, both off-line and on-line.**
  In this thesis, we only applied our tools to quantum repeaters which are positioned on a line. Real networks are, however, two- or three-dimensional. Consequently, there are multiple paths through the network over which a chain of quantum repeaters could deliver entanglement. Determining the optimal paths over which quantum information operations are performed ('optimal' in terms of fidelity or rate of the delivered entanglement) is the topic of routing, which is a particularly nontrivial question if multiple (not-necessarily disjoint) subsets of all nodes request to share entanglement at the same time. In existing work on quantum network routing [7–9], the schemes used for generating remote entanglement often have very stringent timing requirements imposed to prevent the entanglement from decohering in memory (e.g. an entanglement swap may only be performed if the two links are delivered within the same time slot). This potentially limits the achievable entanglement distribution rates.

  The tools presented in this thesis capture more refined memory noise models, enabling us to assess the performance of routing protocols with less stringent timing restrictions (or even none). Such schemes potentially have a higher entanglement distribution rate at the cost of only a limited decrease in entanglement quality due to the longer storage times in imperfect memory.

  In addition, we note that since the repeater schemes we investigate are probabilistic, it might be that a pre-determined path is, once some time has passed, no longer the optimum. Hence, we might want to reschedule the path. It would be interesting to investigate the performance of a routing algorithm where a central classical processor computes the optimal path (or paths, in case of multiple pairs of users requesting bipartite entanglement), re-evaluates this path at every

timestep and reschedules the path if another becomes the new optimal path. We thus arrive at a global, on-line, dynamic routing algorithm. Since the routing happens on-line, the runtime of the algorithm is even more important than in the off-line case, because of the decrease in state quality due to finite memory coherence times while the new optimal path is being computed.

- **Satellite-based repeaters for very long distances.**
  In Chapter 8, we investigated which parameters have the largest effect on boosting the fidelity of the entanglement which is produced by an NV repeater chain. We observed that the most relevant hardware parameter is the photon detection probability, i.e. the success probability of emitting a photon locally by a communication qubit and having it detected at the midpoint. (The second most relevant parameter is the storage qubit noise which is induced during every elementary-link generation attempt and thus is automatically improved by an increased photon detection probability.) Improving the detection probability also evidently increases the entanglement delivery rate.

  If our goal is to obtain a high detection efficiency, glass fibre is not an ideal medium because of its relatively high attenuation (the probability of photon loss doubles every 22 kilometres). A better alternative could be free space, where photon loss increases only quadratically with distance instead of exponentially. It would therefore be interesting to use NetSquid for studying communication through free space and satellites instead of ground stations with glass fibre. Such a setup might be viable for very long distances, despite the increased cost and other technological challenges that need to be overcome.

- **Repeater protocol design, for example including cut-offs, in detailed models of quantum repeaters (NV).**
  In Chapter 6, we have optimised quantum repeater protocols including a cut-off. This optimisation was done using an abstract hardware model, in which there are no restrictions on the number of quantum memories available at a node, nor on the number of local operations that can be performed in parallel. Using this model, we quantified the benefits of the use of a cut-off.

  We cannot directly infer that the same benefits holds for the scenario where nodes hold a restricted quantum processor, such as a single NV centre. The reason for this is that a cut-off mitigates memory noise, which depends on the time that qubits are stored in memory. There are two differences with the abstract model, which influence this time: first, the NV can perform deterministic two-qubit operations, lowering the waiting time, while on the other hand, more time will possibly be spent on performing local gates due to the restricted gate topology of the NV centre (see sec. 8.3.2). Hence, we cannot directly use our results from Chapter 6 to quantitatively infer the performance increase from adding cut-offs to of an NV repeater chain. Qualitatively, however, we know that a cut-off can be beneficial, since the cut-off has already been used in state-of-the-art NV experiments of at most three nodes [10, 11].

  It would be interesting to extend the results of Chapter 6 to NV repeater chains.

In particular, to investigate the performance increase of chains of NV quantum repeaters using a cutoff beyond the three-node regime and the resulting lowered hardware requirements for realising a chain of NV quantum repeaters. For this analysis, we could attempt to extend the fast abstract-model algorithm from Chapter 6 to a more detailed NV-centre model. Unfortunately, in its current form, this will only allow us to treat `tree-shaped`-type protocols. Alternatively, we use our existing NV modelling in NetSquid, with the benefit that we can then also include both `tree-shaped` and `non-tree-shaped`-type protocols.

REFERENCES

[1] H.-J. Briegel, W. Dür, J. I. Cirac, and P. Zoller, *Quantum repeaters: The role of imperfect local operations in quantum communication,* Phys. Rev. Lett. **81**, 5932 (1998).

[2] L.-M. Duan, M. D. Lukin, J. I. Cirac, and P. Zoller, *Long-distance quantum communication with atomic ensembles and linear optics,* Nature **414**, 413 EP (2001), article.

[3] L. Jiang, J. M. Taylor, and M. D. Lukin, *Fast and robust approach to long-distance quantum communication with atomic ensembles,* Phys. Rev. A **76**, 012301 (2007).

[4] N. Sangouard, C. Simon, H. de Riedmatten, and N. Gisin, *Quantum repeaters based on atomic ensembles and linear optics,* Rev. Mod. Phys. **83**, 33 (2011).

[5] H. N. Nagaraja, *Some reliability properties of order statistics,* Communications in Statistics - Theory and Methods **19**, 307 (1990).

[6] G. Vardoyan, S. Guha, P. Nain, and D. Towsley, *On the stochastic analysis of a quantum entanglement switch,* SIGMETRICS Perform. Eval. Rev. **47**, 27 (2019).

[7] Y. O. Luís Bugalho, Bruno C. Coutinho, *Distributing multipartite entanglement over noisy quantum networks,* arXiv:2103.14759 (2021).

[8] M. Pant, H. Krovi, D. Towsley, L. Tassiulas, L. Jiang, P. Basu, D. Englund, and S. Guha, *Routing entanglement in the quantum internet,* npj Quantum Information **5**, 25 (2019).

[9] K. Chakraborty, F. Rozpędek, A. Dahlberg, and S. Wehner, *Distributed routing in a quantum internet,* arXiv:1907.11630 (2019), arXiv:1907.11630 .

[10] N. Kalb, A. A. Reiserer, P. C. Humphreys, J. J. W. Bakermans, S. J. Kamerling, N. H. Nickerson, S. C. Benjamin, D. J. Twitchen, M. Markham, and R. Hanson, *Entanglement distillation between solid-state quantum network nodes,* Science **356**, 928 (2017).

[11] M. Pompili, S. L. N. Hermans, S. Baier, H. K. C. Beukers, P. C. Humphreys, R. N. Schouten, R. F. L. Vermeulen, M. J. Tiggelman, L. dos Santos Martins, B. Dirkse, S. Wehner, and R. Hanson, *Realization of a multinode quantum network of remote solid-state qubits,* Science **372**, 259 (2021), https://science.sciencemag.org/content/372/6539/259.full.pdf .

ACKNOWLEDGEMENTS

Four years ago, I embarked on a tough but amazing journey. I would like to thank the incredible people whom I have had the pleasure to share this journey with.

First and foremost, **David**, I dearly thank you for your guidance throughout the PhD process. I wonder if I could have finished without your calm manner, honesty, pressure at the right moments while still providing lots of freedom. I am very grateful for the fact that you always had time for advice and that I have always felt I could ask anything. I learnt a great deal from your pragmaticism (mostly: finish things!), which is a lesson that will – no doubt – prove very valuable for the remainder of my career.

Stephanie, I strongly admire your determination and vision for where the field should head next. As part of your group, I have learnt many team working skills which are to a large part courtesy of your efforts to make your group focus on collaboration. I may only hope that, one day, I will have the same keen eye as you for which parts of research – whether it is about writing a paper or solving a technical problem – really matter, allowing one to progress rapidly. Many thanks for the great advice you have given over the years.

I would also like to thank the members of my PhD defence committee for so kindly spending their time to critically read this thesis.

I have been very privileged as supervisor of three highly talented master students. **Sebastiaan**, ik heb genoten van onze succesvolle samenwerking. Ik denk regelmatig terug aan je nuchtere houding ten aanzien van een PhD en onderzoekswerk wanneer ik me te druk maak over deadlines. Nu je werkt aan decision diagrams, hoop ik dat je bij je langs mag komen als ik voor lastige keuzes sta. **Boxi**, I am still amazed by the speed at which you seem to process incredibly complex information, and then add some new knowledge to it as if you had been thinking about it for years. I greatly enjoyed our collaboration, from which I learnt that no joke is too stupid to laugh about. **Matthijs**, je energie en positiviteit is aanstekelijk. Waar ieder ander in ieder geval één dag bij de pakken neer zou gaan zitten wanneer je net-nieuw-bedachte stelling toch niet klopt, ging je vol goede moed meteen terug naar de figuurlijke tekentafel. Het was een plezier om met je samen te werken, en ik heb er alle vertrouwen in dat je het ver gaat schoppen met je combinatie van technische kennis en je sterke communicatieve kant.

As part of the NLBlueprint team, I have had a warm, funny and interesting group of colleagues around me for many hours. **Axel**, **David M**, **Francisco**, **Guus**, **Hana**, **Julian**, **Adrià**, **Flors**, I really enjoyed to discuss and laugh with you. It has been a valuable experience to go through the many hours of coding and code reviews, all the sprint deadlines and stand-ups with you, in particular when we all had to suddenly work online for more than a year. I am certain your future colleagues, whoever they will be, will enjoy your presence as much as I do.

Kenneth, ik zal niet gauw vergeten hoe we 'clapping music' van Steve Reich samen probeerden uit te voeren, en hoe binnen mum van tijd ook Jaco Pastorius, entangle-

ment distillation en 'muscle memory' in onze gesprekken de revue waren gepasseerd. Als onbekende in de Delftse kwantumwereld heb ik veel van je geleerd, en zeker niet alleen wat kwantumtechnologie betreft. Dank voor je warme aanwezigheid, en: jaja, ik zal voorzichtig zijn :p.

Filip, dziękuję bardzo for being such an incredible travel mate during our Poland trip. I still feel highly priviliged that you were willing to take me, who speaks barely more than ten words of Polish, together with you on the cycling trip. You were great fun and I have no doubt your career will greatly benefit from your honesty and open-mindedness.

Álvaro, **Bas**, **Bart**, **Ben**, **Carl**, **Carlo**, **Eduardo**, **Gláucia**, **Gayane**, **Jérémy**, **Jonas**, **Kaushik**, **Lennart**, **Leon**, **Mark**, **Martijn**, **Matt**, **Roeland**, **Sébastian**, **Siddhant**, **Stefan**, **Tinh**, **Valentina**, **Victoria**, **Wojtek**, **Yves**, and all the others in David's and Stephanie's groups, my PhD would not have been the same without all of the interesting and funny conversations we have had. You are wonderful people and I feel priviliged to have shared the PhD experience with you. **Rob**, as far as I recall, my first work meeting was with you and David. Your calm manner and always-interested attitude were great to have around. I have learnt (and still can learn!) a great deal from the quality of code you write. The NetSquid paper was a big beast, and I am very proud that we brought it to completion. I hope you are too. **Lieuwe**, **Alfons**, dank jullie wel dat jullie me de wondere wereld van decision diagrams hebben laten zien. Onze samenwerking is voor mij een schoolvoorbeeld van het samenkomen van twee vakgebieden op een moment dat je het niet verwacht. Dat is niet op z'n minst te danken aan jullie altijd-open houding. Ik kijk ernaar uit weer met jullie voor een whiteboard te staan om te bedenken hoe we kwantumtoestanden nóg sneller kunnen simuleren. **Ariana**, many thanks for your advice on running big simulations, and for being there for any supercomputer-related question. Many thanks also to **Matteo**, **Sophie**, **Conor**, **Arian**, **Ronald** and the others from **Team Diamond** for teaching me about the NV center and its magical properties. **Przemek**, thank you for all the warm conversations, and the time you made for both chit-chats and deep conversations, despite your busy schedule. I would also like to thank the other co-authors I have not mentioned so far: **Damian**, **Julio**, **Koji**, **Leon**, **Loek**, **Martijn**, **Stefan**, **Vedran**, **Walter**.

Jed, Christian, if you had not been such great master thesis supervisors, I might not even have started a PhD. Many thanks for all the advice you have given over the years. **Sjoerd**, je positiviteit is ontzettend aanstekelijk. Ik hoop dat je kaartspel een groot succes wordt. **Matteo**, **Guan**, **Anne-Marije**, **Adriaan**, **Jonas**, **Aletta**, it has been great to work with you as part of the QuTech blog team. If it weren't for you, I would never have heard some of the QuTech gossip (and QuTech would not have had the pubquiz we organised during lockdown).

Playing soccer is always a very welcome interruption of research work, especially with the Kavli Warriors soccer team: **Alberto**, **Anta**, **Christian**, **Conor**, **Francesco**, **Francisco**, **Gustavo**, **Hany**, **Jorge**, **Kaushik**, **Luca**, **Lukas**, **Maarten**, **Michael**, **Nikos**, **Sébastian**, **Stefano**, and **Thijs**, it was great to play with you all. I was also fortunate to be part of the QuTech band for a single gig. **Q2**, I hope many QuTecheers may enjoy your music for a long time to come.

Josh, you mentioned one day you dreamt about my PhD defense, and that my entire PhD thesis consisted of art work. (Up to this day I am not sure if I should have taken that as an omen.) I am sorry to disappoint you that this has not come true (or do the images

at the bottom of this thesis's pages count as minimalist art?).

Many thanks also to all my family and friends from Utrecht, Apeldoorn and beyond. Your support during the tough weeks has been incredibly helpful, and you reminded me – again and again – that there are many more important things in life than work.

Marlies, met niemand anders had ik alle hoog- en dieptepunten van een PhD liever willen delen.