Quantum network routing via link prediction

Antal Száva

Quantum network routing via link prediction

 ;

Master's Thesis in Computer Science

Quantum Internet and Networked Computing
Faculty of Electrical Engineering, Mathematics, and Computer Science
Delft University of Technology

Antal Száva

21st August 2019

Author
Antal Száva

Title
Quantum network routing via link prediction

MSc presentation
28th of August 2019

Graduation Committee

Prof.dr. S.D.C. Wehner	Delft University of Technology
Prof.dr.ir. D.H.J. Epema	Delft University of Technology
dr. David Elkouss	Delft University of Technology

Abstract

Quantum communication promises to allow groundbreaking new ways of information exchange. Recent advancements in the field have shown that the emergence of a quantum internet is not too far in the future.

A network of devices capable of transmitting quantum information called a quantum network has promising applications with vast benefits. One of the most near term achievable application is quantum key distribution. It could be used for sharing a secret key between two end users through any insecure authenticated communication channel. Other than sharing secret key communication, a quantum network can be used for connecting quantum computers. Quantum computers have the potential to solve computational problems that their classical counterparts either can not do or would require significantly more time to do so. By connecting such devices, distributed computation problems such as leader election or distributed consensus between nodes can be solved securely.

For such applications to be put into practice on a large scale, there are open practical questions still to be answered. In a quantum network, a data qubit containing quantum information is transmitted by an operation called quantum teleportation. For it, two nodes need to share entanglement between each other. Quantum teleportation then ensures the safe transmission of a qubit by using the shared entanglement and the exchange of two classical bits. This operation, however, consumes entanglement between the nodes, changing the network topology with each served request.

In quantum networks, certain nodes share entanglement between each other. Routing in quantum networks entails determining which of these shared entanglements are used to transmit quantum information. As this proves to be a difficult task, we study quantum networks and in particular consider the problem of routing entanglement in quantum networks.

The average latency for routing entanglement in a quantum network has been studied so far using a distributed routing approach. Thus, we present numerical simulation results of centralised routing for the average latency of demands in two existing entanglement generation models. In the first, shared entanglement between nodes is created on-demand. In the second, certain nodes pre-share entanglement before the demand comes. Our results show the intuition that using a centralised routing approach in the on-demand model results in drastically more average latency than in the model with pre-shared entanglement.

Since some nodes might not be informed about the change in topology, we study the effect of the propagation of information about the network topology to nodes in the network. Our observations based on simulation results show that the average latency is significantly higher if the information about the topology is not propagated well in the network. The message complexity, however, for propagating information to all nodes in a network for a single request is $O(2|E| - |V|)$.

Since this would be a significant load for the network, we consider information propagation within a radius and still observe a considerable decrease in average latency.

At last, we present a technique called link prediction which can be performed by any node without the need for information propagation and still achieve a considerable decrease in average latency. It can be effectively used to predict the change of topology in a quantum network by using knowledge about the network topology for a certain future point in time and previous knowledge about the network traffic.

For a large scale quantum network, the use of information propagation within a certain radius together with link prediction can be used for a significant improvement in the average latency for requests.

Acknowledgements

First of all, I would like to thank Stephanie Wehner for accepting me as a Master student and giving me a chance to learn more about quantum communication.

Special thanks to Kaushik Chakraborty for the extensive professional support and contributions to this thesis. Also, I greatly value the life lessons that you have taught me.

Further thanks go to Dick Epema, for agreeing that I do my thesis related to the quantum internet and David Elkouss for joining my defence committee.

I appreciate the many productive discussions in the Wehner group and with everyone who has helped me during my work.

Furthermore, I would like to express my immense gratitude to EIT Digital for providing me with amazing opportunities during my Master programme including the chance to study abroad in Germany and the Netherlands.

Dear memories of fun times spent together with my friends have been invaluable in my life, thank you for that.

Last but not least, I would like to thank my parents and my sister for always supporting me and standing by my side unconditionally.

Antal Száva

Delft, The Netherlands
21st August 2019

Contents

Chapter 1

Introduction

Quantum computing and quantum communication promise to allow groundbreaking new ways of computation and information exchange through the use of quantum mechanics. Recent advancements in engineering combined with physics and computer science expertise have allowed the further development of these fields. The practical realisation of full-fledged quantum technologies can be expected in the coming years.

1.1 A first look at quantum networks

In classical computing technology, we use the concept of the bit as a basic unit of information. A bit can have 0 or 1 as a value. Quantum mechanics allows a new way of thinking about information. A single quantum bit, the unit of quantum information can be both 0 and 1 at the same time. In such a case we say that the qubit is in a superposition. An important feature of qubits is that unlike classical bits, they cannot be cloned or copied (no-cloning theorem). Through certain operations, however, several qubits can be made to interact with each other. Two quantum states can become correlated once a quantum mechanical phenomenon called entanglement is established between them. Entanglement can be utilised for quantum communication and quantum computation. The creation of entanglement has already been demonstrated in practice for several distances and setups over the years [TBZG98] [Hen15]. Once such a connection has been created, through the process of quantum teleportation, quantum information can be transmitted between two nodes separated by a distance [BBC+93].

Creating a network of devices connected with entanglement is the main goal of a quantum network, a vision that has been described over the years [Kim08] [Met14] [LSW+04][WEH18]. The basic application for a quantum network would be secure communication using quantum key distribution (QKD). QKD allows parties to share a random secret key that can be used for the encryption and decryption of messages guaranteeing secure communication [BB84] [Eke91].

Quantum networks could also connect quantum computers. Quantum computers

have the potential to solve computational problems that their classical counterparts either cannot do or would require significantly more time to do so. Several quantum algorithms like Shor's period-finding algorithm [Sho94], Grover's search algorithm [Gro96] and the quantum algorithm for solving linear systems of equations [WHHL09] are expected to offer a speedup compared to classical approaches. The modelling of complex quantum systems would be furthermore one of the important applications of quantum computing.

Once such quantum computers are connected, their tasks can be broken up into separate parts of computation and may be distributed to several quantum processors in the network [JCM99]. This capability gave rise to proving the possibility of leader election and distributed consensus in anonymous quantum networks [DP04], problems which are impossible to solve in the classical case. Later, algorithms for solving such problems also appeared [TKM05]. As more advanced forms of quantum networks and quantum computation are developed, more and more solutions are expected to be implemented using quantum networks. Further use cases include clock synchronisation, quantum sensor networks and secure remote quantum computations [WEH18].

1.2 Network components of a quantum internet

Optical fibers are used to transmit classical information. They serve as *classical communication links* in the classical internet. Similarly, optical fibers can be used as *physical quantum communication channels* to send photons and share entanglement, thus enabling quantum communication.

The transmission rate for classical and quantum information in an optical fibre decreases exponentially with the distance. This causes packet loss for vast distances. In classical networking, packets are retransmitted by intermediary repeaters in such cases. Due to the no-cloning theorem, however, quantum information cannot be copied for retransmission. Thus, special devices, so-called quantum repeaters need to be used in quantum networks.

Quantum repeaters separated by great distances can become entangled through the operation called entanglement swap [BBC$^+$93][ZZHE94] [GWZ$^+$08]. Let us consider three parties positioned in a chain as in figure 1.1, *Alice*, *Bob* and a *Router*. *Alice* is connected with the *Router* and the *Router* is also connected with *Bob* through a physical quantum communication channel. Although there is no such connection between *Alice* and *Bob*, they would like to share entanglement. This is possible for them if they both separately share entanglement with the *Router* at the same time. This requires one qubit to be stored with *Alice* and with *Bob* and two qubits to be stored at the *Router*. Once the *Router* shares entanglement with the two other nodes, it can perform entanglement swap. This entails teleporting the qubit of the *Router* entangled with *Alice* to *Bob*, eventually making *Alice* and *Bob* share entanglement. For this, the entanglement between the *Router* and *Bob* is consumed and a classical communication channel is used to send two classical

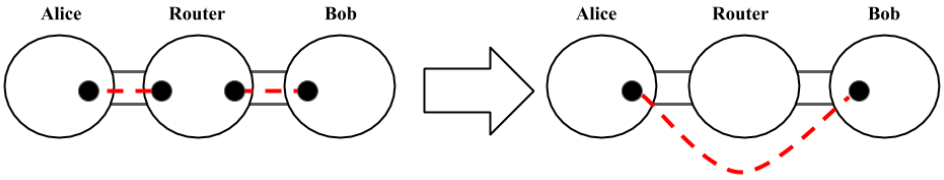


Figure 1.1: The operation of entanglement swap allows the creation of entanglement between two qubits even if there is no physical quantum communication channel between them.

bits. By repeating the entanglement swap operation iteratively, remote nodes can become entangled. For this, several intermediary quantum repeaters can be used. In a quantum network, nodes can share just as many entangled links as many qubits they can store. Once a node shares entangled links with multiple nodes at the same time, it needs to select which one to perform entanglement swap with. This selection is regarded as a routing decision. Quantum repeaters taking routing decisions are called *quantum routers*. Routing in a quantum network is the task of routing entanglement between source and destination with the assistance of intermediary nodes [Met14][Cal17][GI18].

1.3 Average latency of demands

A quantum network would be used to serve requests for transmitting quantum information. As for the present-day classical internet, there would be requirements imposed on the service provided by the network. One of such requirements would be keeping the time it takes for serving requests low. In quantum networks this time, referred to as latency, is greatly affected by the time entanglement generation takes [CRDW19]. In this thesis, we will be carrying out analysis to see estimates for the average latency of demands. Once factors affecting this measure are identified, techniques aimed at reducing it can be devised.

1.4 Research questions

Entanglement in a quantum network is generated and consumed rapidly. This property results in a network whose topology is dynamically changing. Information about the dynamically changing topology can affect routing decisions. Therefore, propagating information in the network might be a solution to achieving better routing decisions. It is, however, problematic to propagate information to all nodes at all times.

In this thesis we will be exploring the following research questions:

1. How can we express the dynamically changing topology of quantum net-

works with regards to time?

2. Given that information is propagated to nodes following a pre-defined approach, what is the average latency for several demands (requests) in a quantum network?

3. How can lower average latency be achieved for demands in a given quantum network even with limited information propagation?

1.5 Contributions

The main contributions in this thesis can be divided into three parts.

First, we describe a mathematical model for modelling the dynamic lifecycle of quantum networks. As mentioned previously, the entanglement between quantum routers is not permanent. It can either be consumed when quantum teleportation or entanglement swap takes place or its fidelity (a quantity describing the quality of the entanglement) can decay with time to such an extent that the entangled link is no longer useful. Therefore, a model which can incorporate the temporal nature of entangled links is essential to describe quantum networks. We present a model using temporal networks that enables a better way of modelling the lifecycle of quantum networks. In this model, we incorporate the time entangled links can be used. For this, we introduce the concept of threshold time and time window. The threshold time represents the amount of time for which the fidelity of entangled links is high enough to be used for quantum communication. The length of a time window is upper bounded by the threshold time. An entangled link can be used for quantum communication in such a time window. Using the proposed model, one can describe when entangled links are available in the network. This is beneficial when trying to determine certain paths that are dependent on a point in time. Analogous to the concept of time-respecting paths from temporal networks, the concept of a time-dependent path is introduced. In our proposed model, the problem of routing in quantum networks can be formulated as searching for such time-dependent paths between source and destination.

In the following part, we focus on the description and simulation of centralised routing approaches in quantum networks. Entangled links are generated and then consumed rapidly in quantum networks. The propagation of information about their current state of the network proves to be important for routing decisions. We define routing approaches based on the knowledge each node has in the network about the topology:

1. Initial knowledge,

2. Local knowledge and

3. Global knowledge.

We further introduce the concept of a *propagation radius* to describe the propagation of information. For each of these approaches, we would like to determine the average latency caused by the generation of end-to-end entanglement between source and destination. We present results of numerical simulations on the average latency of entanglement generation after the quantum network has served a specific number of demands.

At last, we define a *link prediction* algorithm with which the availability of entangled links in the quantum network can be predicted. This algorithm can be used for decreasing the average latency of demands. The idea of link prediction has been discussed widely in network science as an example to trying to predict links that will be present in a given network [LNK03]. In our case, statistical measures are calculated based on the shortest paths in quantum networks. For this, nodes use their local information about the state of the topology from a certain point in time and information about the elapsed time since then. We present simulation results for link prediction showing that a considerable advantage in average latency can be achieved by using this algorithm.

1.6 Outline

We provide a summary of the chapters in this section. In Chapter 2 and Chapter 3 cover the preliminaries of classical networking, quantum information theory and quantum networks. Chapter 4 introduces existing quantum network models. In Chapter 5 we present a mathematical structure for quantum networks capable of describing the lifecycle of the network. In Chapter 6 we analyse how propagating information about the change in network topology affects the average latency. Sections 6.4,6.5 and 6.6 cover three approaches with simulation results based on the level of information propagation. In Chapter 7 we describe link prediction used for reducing the average latency only by computing shortest paths based on local information. This technique can be used effectively for pathfinding with lower latency. At last, Chapter 8 summarises the results obtained and provides a list of potential future work.

Chapter 2

Quantum information theory and quantum networks

In this chapter, we describe the concepts connected to quantum information and quantum networks. First, we discuss the four postulates of quantum mechanics. We review the concept of the quantum state, quantum gate, tensor product, unitary evolution and give examples for quantum gates. Then we discuss some interesting tools of quantum information theory, such as the no-cloning theorem, quantum teleportation and entanglement swap. We then provide a description on the function and the building blocks of quantum networks: quantum links, quantum repeaters, quantum routers and entanglement generation in a quantum network. At the end of the chapter, we move our discussion to the phases of end-to-end entanglement creation in a quantum network.

2.1 Basic quantum mechanics and quantum information theory

One of the greatest advancements in physics in the 20th century was the emergence of quantum mechanics. In this section, we will discuss the basic postulates of quantum mechanics and important examples of quantum information theory. For a more detailed description, we refer to Chapter 2 of [NC10].

Postulate 1. *(Quantum State) A complex vector space with an inner product (Hilbert space) is associated with any isolated physical system and is called the state space of the system. The system is completely described by its state vector which is a unit vector in the system's state space.*

Similarly to the concept of *bits* in classical information, the smallest possible information processing unit in quantum information is called *qubit*. It is an element of $\mathbb{C}^2$. It is written as,

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{2.1}$$

where $\alpha, \beta \in \mathbb{C}$ satisfy

$$|\alpha|^2 + |\beta|^2 = 1 \tag{2.2}$$

and $|0\rangle, |1\rangle$ are two basis vectors:

$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

In general any pure quantum state in a Q-dimensional Hilbert space $\mathcal{H}_Q \simeq \mathbb{C}^Q$ can be written as,

$$|\psi\rangle = \sum_{j=0}^{Q-1} \lambda_j |j\rangle, \tag{2.3}$$

where $\{|j\rangle\}_{j=0}^{Q-1}$ forms an orthonormal basis for $\mathcal{H}_Q$ and $\sum_j |\lambda_j|^2 = 1$.

The change of $|\psi\rangle$ with time is described by the following postulate.

Postulate 2. *The evolution of a closed quantum system is described by a unitary transformation. That is, the state $|\psi\rangle$ of the system at time t_1 is related to the state $|\psi'\rangle$ of the system at time t_2 by a unitary operator U which depends only on the times t_1 and t_2,*

$$|\psi'\rangle = U |\psi\rangle. \tag{2.4}$$

Analogous to the way a classical computer is built from an electrical circuit containing wires and logic gates, a quantum computer is built from a *quantum circuit* containing wires and elementary *quantum gates* to carry around and manipulate the quantum information. Quantum circuits are to be read from left-to-right. Each line in the circuit represents a wire in the quantum circuit. The left-hand side shows the initial states, the right-hand side the resulting states. Examples of quantum circuits will follow later in this chapter.

Quantum gates on a single qubit can be described by two by two matrices. The only requirement for such a matrix to be a valid quantum gate is unitarity. An operator U is said to be *unitary*, if $UU^\dagger = U^\dagger U = I$, where $U^\dagger$ is the adjoint of an operator U (obtained by transposing and then complex conjugating U), and I is the two by two identity matrix. This comes from the fact that the normalization condition, equation 2.2 needs to be true also after the gate has acted.

The following operators are called *Pauli matrices*:

$$I \equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad X \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$Y \equiv \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad Z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

The X quantum gate represents the NOT operation, as it acts as a bit flip operation on single qubits:

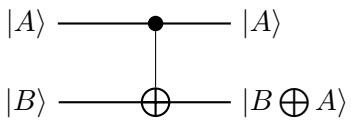
$$X|\psi\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}$$

The following gate is called the *Hadamard gate*:

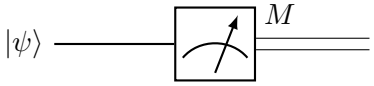
$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

There are also multi-qubit quantum gates. One of the most important two-qubit gates is the controlled-NOT or CNOT gate:

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$



(a) The quantum circuit of the controlled-NOT gate.



(b) The quantum circuit depiction of a measurement. M stands for the result of the measurement.

Figure 2.1: The quantum circuit depiction of a CNOT gate and a measurement.

This gate has two input qubits, known as the *control* qubit and the *target* qubit, respectively. The action of the gate may be described as follows. If the control qubit is set to 0, then the target qubit is left alone. If the control qubit is set to 1, then the target qubit is flipped. The circuit for a CNOT gate and the measurement of a qubit are shown in subfigures 2.1a and 2.1b.

Postulate 3 provides a means for describing the effects of measurements on quantum systems.

Postulate 3. *Quantum measurements are described by a collection* $\{M_m\}$ *of measurement operators. These are operators acting on the state space of the system being measured. The index* m *refers to the measurement outcomes that may occur in the experiment. If the state of the quantum system is* $|\psi\rangle$ *immediately before the measurement, then the probability* $p(m)$ *that the result* m *occurs is given by*

$$p(m) = \langle\psi| M_m^\dagger M_m |\psi\rangle , \tag{2.5}$$

and the state of the system after the measurement is

$$\frac{M_m |\psi\rangle}{\sqrt{\langle\psi| M_m^\dagger M_m |\psi\rangle}}. \tag{2.6}$$

The measurement operators satisfy the completeness equation

$$\sum_m M_m^\dagger M_m = I. \tag{2.7}$$

A measurement is called *projective* if the operators $\Pi_m = M_m^\dagger M_m$ are projectors, that is if $\Pi_m^2 = \Pi_m$. A property of such measurements is that performing the measurement again immediately after the first one yields the same result with probability 1.

For the fourth and last postulate, we will have to look at the definition of the tensor product. The *tensor product* is a way of putting vector spaces together to form larger vector spaces.

Suppose V and W are vector spaces of dimension *m* and *n* respectively. For convenience we also suppose that V and W are Hilbert spaces. Then $V \otimes W$ (read 'V tensor W') is an *mn* dimensional vector space.

By definition the tensor product satisfies the following basic properties:

1. For an arbitrary scalar z and elements $|v\rangle$ of V and $|w\rangle$ of W,
$$z(|v\rangle \otimes |w\rangle) = z(|v\rangle) \otimes |w\rangle = |v\rangle \otimes z(|w\rangle).$$
2. For an arbitrary $|v_1\rangle$ and $|v_2\rangle$ in V and $|w\rangle$ in W,
$$(|v_1\rangle + |v_2\rangle) \otimes |w\rangle = |v_1\rangle \otimes |w\rangle + |v_2\rangle \otimes |w\rangle .$$
3. For an arbitrary $|v\rangle$ in V and $|w_1\rangle$ and $|w_2\rangle$ in W,
$$|v\rangle \otimes (|w_1\rangle + |w_2\rangle) = |v\rangle \otimes |w_1\rangle + |v\rangle \otimes |w_2\rangle .$$

A convenient matrix representation for the tensor product is known as the *Kronecker product*. Suppose *A* is an m by n matrix, and *B* is a p by q matrix. Then we have the matrix representation:

$$A \otimes B \equiv \overbrace{\begin{bmatrix} A_{11}B & A_{12}B & A_{13}B & \dots & A_{1n}B \\ A_{21}B & A_{22}B & A_{23}B & \dots & A_{2n}B \\ \vdots & \vdots & \vdots & \dots & \vdots \\ A_{m1}B & A_{m2}B & A_{m3}B & \dots & A_{mn}B \end{bmatrix}}^{\text{nq}} \Bigg\} mp. \tag{2.8}$$

In this representation terms like $A_{11}B$ denote p by q submatrices whose entries are proportional to B, with overall proportionality constant A_{11}. For example, the tensor product of the vectors (1, 2) and (2, 3) is the vector

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} \otimes \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \times 2 \\ 1 \times 3 \\ 2 \times 2 \\ 2 \times 3 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 4 \\ 6 \end{bmatrix} . \tag{2.9}$$

Postulate 4. *The state space of a composite physical system is the tensor product of the state spaces of the component physical systems. Moreover, if we have systems numbered 1 through n, and system number i is prepared in the state $|\psi_i\rangle$, then the joint state of the total system is $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$.*

Postulate 4 enables us to define *entanglement*.

2.1.1 Entanglement

A special kind of correlation between multiple qubits is called entanglement.

Let us consider the following two qubit state:

$$|\Phi_{00}\rangle_{AB} = \frac{1}{\sqrt{2}}(|00\rangle_{AB}) + |11\rangle_{AB}) \tag{2.10}$$

This state has the property that there are no single qubit states $|a\rangle$ and $|b\rangle$ such that $|\Phi_{00}\rangle_{AB} = |a\rangle\,|b\rangle$. We say that a state of a composite system for which $|\Phi_{00}\rangle_{AB} \neq |\alpha\rangle \otimes |\beta\rangle$ is an entangled state, where $|\alpha\rangle$ and $|\beta\rangle$ holds for all single qubit states $|a\rangle$ and $|b\rangle$.

The following two-qubit entangled states form a basis in the two-qubit space and are called the Bell states or EPR pairs (after Einstein, Podolsky and Rosen):

$$\begin{aligned}
|\Phi_{00}\rangle_{AB} &= \frac{1}{\sqrt{2}}(|00\rangle_{AB}) + |11\rangle_{AB}) \\
|\Phi_{01}\rangle_{AB} &= \frac{1}{\sqrt{2}}(|00\rangle_{AB}) - |11\rangle_{AB}) \\
|\Phi_{10}\rangle_{AB} &= \frac{1}{\sqrt{2}}(|01\rangle_{AB}) + |10\rangle_{AB}) \\
|\Phi_{11}\rangle_{AB} &= \frac{1}{\sqrt{2}}(|01\rangle_{AB}) - |10\rangle_{AB})
\end{aligned}$$

Figure 2.2 shows the circuit of how the entangled state $|\Phi_{00}\rangle_{AB}$ state can be created from the $|00\rangle$ state.

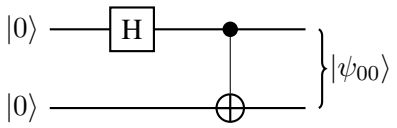


Figure 2.2: The quantum circuit of entanglement creation.

An important measure of entanglement is called the fidelity. It describes the quality of the entanglement with the constraint of $0 \leq F \leq 1$. The desired value of fidelity is 1, whereas a minimum value of $F \geq 0.5$ is often needed [AD].

2.1.2 No-cloning theorem

An important feature of qubits is called no-cloning. It states that it is not possible to make a copy of an unknown quantum state. Suppose we have a quantum machine with two slots labeled A and B. Slot A, the *data slot*, starts out in an unknown, but pure quantum state, $|\psi\rangle$. This is the state which is to be copied into slot B, the

target slot. We assume that the target slot starts out in some standard pure state, $|s\rangle$. Thus the initial state of the copying machine is

$$|\psi\rangle \otimes |s\rangle \,. \tag{2.11}$$

Some unitary evolution U now affects the copying procedure, ideally,

$$|\psi\rangle \otimes |s\rangle \xrightarrow{U} U(|\psi\rangle \otimes |s\rangle) = |\psi\rangle \otimes |\psi\rangle \tag{2.12}$$

Suppose this copying procedure works for two particular pure states, $|\psi\rangle$ and $|\phi\rangle$. Then we have

$$U(|\psi\rangle \otimes |s\rangle) = |\psi\rangle \otimes |\psi\rangle \tag{2.13}$$

$$U(|\phi\rangle \otimes |s\rangle) = |\phi\rangle \otimes |\phi\rangle \,. \tag{2.14}$$

Taking the inner product of these two equations gives

$$\langle\psi\,|\phi\rangle = (\langle\psi\,|\phi\rangle)^2. \tag{2.15}$$

But $x = x^2$ has only two solutions, $x = 0$ and $x = 1$, so either $|\psi\rangle = |\phi\rangle$ *or* $|\psi\rangle$ and $|\phi\rangle$ are orthogonal. Thus a cloning device can only clone states which are orthogonal to one another, and therefore a general quantum cloning device is impossible. A potential quantum cloner cannot, for example, clone the qubit states $|\psi\rangle = |0\rangle$ and $|\phi\rangle = \frac{(|0\rangle+|1\rangle)}{\sqrt{2}}$, since these states are not orthogonal. This shows that it is impossible to perfectly clone an unknown quantum state using unitary evolution.

2.1.3 Quantum Teleportation

An operation called quantum teleportation allows safe quantum information transmission even without a quantum communication channel between two parties.

Let us assume that Alice and Bob are separated by physical distance, but they share a $|\Phi_{00}\rangle$ entangled quantum state with each other. Alice has a qubit $|\psi\rangle$, which she would like to transmit to Bob. During quantum teleportation, she can do so just by sending classical messages and using the state $|\Phi_{00}\rangle$ that is pre-shared between the two of them. Consequently, she does not need to send her qubit $|\psi\rangle$ in a quantum communication channel to Bob.

The circuit for quantum teleportation is represented in figure 2.4. We denote the joint state with $|\Psi_0\rangle$ at the beginning of the operation. The first two qubits belong to Alice, whereas the third qubit belongs to Bob.

The quantum circuit shown in Figure 2.4 gives a more precise description of quantum teleportation. The state to be teleported is $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, where α and β are unknown amplitudes. The state input into the circuit $|\Psi_0\rangle$ is

$$|\Psi_0\rangle = |\psi\rangle|\Phi_{00}\rangle = \frac{1}{\sqrt{2}}[\alpha\,|0\rangle\,(|00\rangle + |11\rangle) + \beta\,|1\rangle\,(|00\rangle + |11\rangle)] \tag{2.16}$$

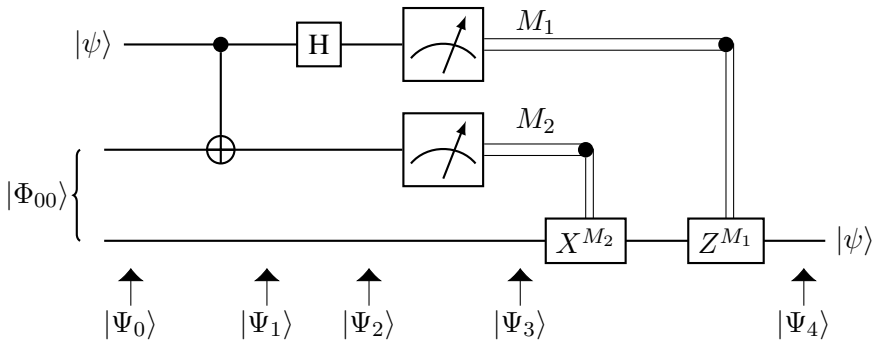


Figure 2.3: The quantum circuit for quantum teleportation of qubit $|\psi\rangle$ [BBC+93].

where we use the convention that the first two qubits (on the left) belong to Alice and the third qubit to Bob. As we explained previously, Alice's second qubit and Bob's qubit start out in an EPR state. Alice sends her qubits through a *CNOT* gate, obtaining

$$|\Psi_1\rangle = \frac{1}{\sqrt{2}}[\alpha\,|0\rangle\,(|00\rangle + |11\rangle) + \beta\,|1\rangle\,(|10\rangle + |01\rangle)] \tag{2.17}$$

She then sends the first qubit through a Hadamard gate, obtaining

$$|\Psi_2\rangle = \frac{1}{2}[\alpha(|0\rangle + |1\rangle)(|00\rangle + |11\rangle) + \beta(|0\rangle - |1\rangle)(|10\rangle + |01\rangle)]$$

This state may be re-written in the following way, simply by regrouping terms:

$$\begin{aligned}|\Psi_2\rangle = \frac{1}{2}[&|00\rangle\,(\alpha\,|0\rangle + \beta\,|1\rangle) + |01\rangle\,(\alpha\,|1\rangle + \beta\,|0\rangle)\\ &+ |10\rangle\,(\alpha\,|0\rangle - \beta\,|1\rangle) + |11\rangle\,(\alpha\,|1\rangle - \beta\,|0\rangle)]\end{aligned}$$

This expression naturally breaks down into four terms. The first term has Alice's qubits in the state $|00\rangle$, and Bob's qubit in the state $\alpha\,|0\rangle + \beta\,|1\rangle$ - which is the original state $|\psi\rangle$. If Alice performs a measurement and obtains the result $|00\rangle$ then Bob's system will be in the state $|\psi\rangle$. Similarly, from the previous expression, we can read off Bob's post- measurement state, given the result of Alice's measurement:

$$00 \longmapsto |\psi_3(00)\rangle \equiv \Big[\alpha\,|0\rangle + \beta\,|1\rangle\Big] \tag{2.18}$$

$$01 \longmapsto |\psi_3(01)\rangle \equiv \Big[\alpha\,|1\rangle + \beta\,|0\rangle\Big] \tag{2.19}$$

$$10 \longmapsto |\psi_3(10)\rangle \equiv \Big[\alpha\,|0\rangle - \beta\,|1\rangle\Big] \tag{2.20}$$

$$11 \longmapsto |\psi_3(11)\rangle \equiv \Big[\alpha\,|1\rangle - \beta\,|0\rangle\Big] \tag{2.21}$$

Depending on Alice's measurement outcome, Bob's qubit will end up in one of these four possible states. Of course, to know which state it is in, Bob must be told the result of Alice's measurement – we will show later that it is this fact which prevents teleportation from being used to transmit information faster than light. Once Bob has learned the measurement outcome, Bob can 'fix up' his state, recovering $|\psi\rangle$, by applying the appropriate quantum gate. For example, in the case where the measurement yields 00, Bob doesn't need to do anything. If the measurement is 01 then Bob can fix up his state by applying the X gate. If the measurement is 10 then Bob can fix up his state by applying the Z gate. If the measurement is 11 then Bob can fix up his state by applying first an X and then a Z gate. Summing up, Bob needs to apply the transformation $Z^{M_1}X^{M_2}$ (note how time goes from left to right in circuit diagrams, but in matrix products terms on the *right* happen *first*) to his qubit, and he will recover the state $|\psi\rangle$.

2.1.4 Entanglement swap

An operation that is vital in quantum networks is called entanglement swap. It allows the creation of entanglement between two parties even if there is no physical quantum communication channel between them. Figure 1.1 summarises this operation.

Alice and Bob would like to share entanglement. Unfortunately, there is no physical quantum communication channel between them. Both of them can, however, generate entanglement with an intermediary router. In such a case, we assume that the router can share entanglement with Alice and Bob at the same time. If it does so, then it can teleport its qubit of the shared entanglement with Alice to Bob. At the end of the teleportation Alice and Bob share entanglement.

Let us assume that a router shares a $|\Phi_{00}\rangle$ Bell state with Alice and shares another $|\Phi_{00}\rangle$ Bell state with Bob. In this case, the joint state of the three parties is as follows:

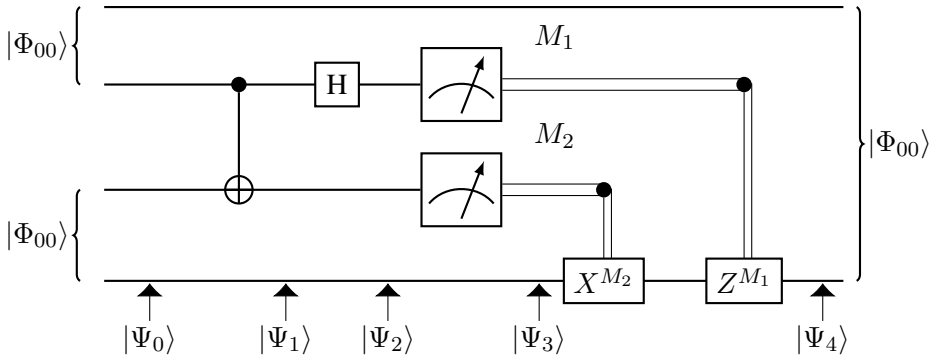


Figure 2.4: The quantum circuit of the entanglement swap operation.

$$
\begin{aligned}
|\Psi_0\rangle &= |\Phi_{00}\rangle_{AR_1} \otimes |\Phi_{00}\rangle_{R_2B} \\
&= \frac{|00\rangle_{AR_1} + |11\rangle_{AR_1}}{\sqrt{2}} \otimes \frac{|00\rangle_{R_2B} + |11\rangle_{R_2B}}{\sqrt{2}}
\end{aligned} \tag{2.22}
$$

This state may be re-written by regrouping terms (Alice and Bob form the first, the two qubits of the router the second group):

$$
\begin{aligned}
|\Psi_0\rangle &= \frac{1}{2}\Big(|0000\rangle_{AR_1R_2B} + |0011\rangle_{AR_1R_2B} \\
&\quad + |1100\rangle_{AR_1R_2B} + |1111\rangle_{AR_1R_2B} \Big) \\
&= \frac{1}{2}\Big(|00\rangle_{AB} |00\rangle_{R_1R_2} + |01\rangle_{AB} |01\rangle_{R_1R_2} \\
&\quad + |10\rangle_{AB} |10\rangle_{R_1R_2} + |11\rangle_{AB} |11\rangle_{R_1R_2} \Big)
\end{aligned} \tag{2.23}
$$

First, a CNOT is applied locally at the router. Qubit R_1 is the control qubit, R_2 is the target qubit of the operation:

$$
\begin{aligned}
|\Psi_1\rangle &= \frac{1}{2}\Big(|00\rangle_{AB} |00\rangle_{R_1R_2} + |01\rangle_{AB} |01\rangle_{R_1R_2} \\
&\quad + |10\rangle_{AB} |11\rangle_{R_1R_2} + |11\rangle_{AB} |10\rangle_{R_1R_2} \Big)
\end{aligned} \tag{2.24}
$$

Then a Hadamard gate is applied on R_1:

$$
\begin{aligned}
|\Psi_2\rangle = \frac{1}{2\sqrt{2}}\Big(& |00\rangle_{AB} (|0\rangle + |1\rangle)_{R_1} |0\rangle_{R_2} \\
& + |01\rangle_{AB} (|0\rangle + |1\rangle)_{R_1} |1\rangle_{R_2} \\
& + |10\rangle_{AB} (|0\rangle - |1\rangle)_{R_1} |1\rangle_{R_2} \\
& + |11\rangle_{AB} (|0\rangle - |1\rangle)_{R_1} |0\rangle_{R_2} \Big)
\end{aligned} \tag{2.25}
$$

This state may be re-written the following way, simply by regrouping terms:

$$
\begin{aligned}
|\Psi_2\rangle = \frac{1}{2\sqrt{2}}\Big(& (|00\rangle_{AB} + |11\rangle_{AB}) |00\rangle_{R_1R_2} \\
& + (|00\rangle_{AB} - |11\rangle_{AB}) |10\rangle_{R_1R_2} \\
& + (|01\rangle_{AB} + |10\rangle_{AB}) |01\rangle_{R_1R_2} \\
& + (|01\rangle_{AB} - |10\rangle_{AB}) |11\rangle_{R_1R_2} \Big)
\end{aligned} \tag{2.26}
$$

Similarly to the general quantum teleportation operation described by equations 2.18-2.21, we can read off Bob's post-measurement state. This state is determined by the result of the measurement of the two qubits at the router. After this, Bob can perform local operations to fix up his state.

2.1.5 Latency of entanglement generation

An important factor of entanglement is the time to generate end-to-end entanglement. This quantity can also be referred to as latency.

Let us assume a discrete-time model (similar to the one in [CRDW19], described later in section 4.3) and introduce the following notations:

- T_{th}: the threshold time for the decay in the fidelity of an entangled link,
- P_0: the probability of successfully generating entanglement between two nodes at a single time step,
- d: the hop distance which is equal to the number of intermediate quantum routers between a source and destination. The hop distance between Alice and Bob is 1 in figure 1.1.

Entanglement generation is attempted with probability P_0 at each time step until it is successful. Once entanglement is shared, the fidelity of the entangled link deteriorates over time. The threshold time represents a time quantity until the fidelity has decreased such that the entangled link can not be used anymore.

Using the previous notations, the event that entanglement can be successfully generated in a chain of $d+2$ routers within the threshold time follows a geometric distribution with probability

$$\mathcal{P}_d = (1 - (1 - P_0)^{T_{th}})^{d+1}, \tag{2.27}$$

where $(1 - P_0)$ is the probability of attempting, but not generating entanglement at a single time step and $1 - (1 - P_0)^{T_{th}}$ is the probability of generating entanglement in T_{th} time steps.

Hence, the expected time entanglement generation takes in this repeater chain is $\frac{1}{\mathcal{P}_d} = (1 - (1 - P_0)^{T_{th}})^{-d-1}$.

Thus, the entanglement generation time scales exponentially with d, the number of routers in the chain (or hop distance using networking terminology).

2.2 Physical realization of quantum networks

2.2.1 Quantum links

The most basic building block for quantum networks are quantum links. A quantum link is an abstraction for the physical realisation of shared entanglement through a physical connection, the quantum channel. Quantum channels are implemented using telecommunication fibres capable of transmitting photons and are essential in the creation of EPR pairs. In other words, quantum links are entangled pairs of qubits in a certain network [WEH18].

The practical realisation of entanglement creation include the use of nitrogen-vacancy centres [Ber13], atoms [HKO+12] [Rit12] and trapped-ions [Moe07]. In

practice, entanglement generation between two nodes in a quantum network has two main characteristics: it is probabilistic and it is a heralded procedure. Consequently, when two nodes would like to share entanglement, they often need to attempt entanglement generation multiple times. Once entanglement has been generated, the fidelity of the joint state will, however, decay over time. For this reason, a certain time threshold may be introduced. This quantity would determine the maximum amount of time for which an entangled link may be used for. Once this time has elapsed, the fidelity of the entangled link has decayed to such an extent, that it cannot be used for quantum communication. In such a case the entanglement is discarded and a new link needs to be generated.

2.2.2 Quantum repeaters and routers

The transmission rate for classical and quantum information in an optical fibre decreases exponentially with the distance. One basic solution used in classical networking to this problem is to amplify the signal. This, however, is not possible in the quantum case due to the no-cloning theorem.

A quantum repeater is a device that would potentially solve this problem. The authors in [BDCZ98] proposed setup for the use of quantum repeaters. A network of quantum repeaters would allow the creation of long-distance entanglement. This would be achieved by first creating entanglement between the end devices and intermediary repeaters that are separated by a smaller distance and eventually performing entanglement swap operations to create an end-to-end entanglement. Although there have been various further schemes proposed since the first description of quantum repeaters, in our models, the previously described notion of quantum repeaters is going to be used.

In our work, we will be referring to controllable quantum repeaters that can choose a next-hop based on a certain strategy as *quantum routers*.

2.2.3 Entanglement generation in a router chain network

There are two ways of creating entangled links between quantum routers in a quantum network:

- Either they are connected by a physical quantum communication channel,
- Or there is a path of physical quantum communication channels in the network connecting two end nodes.

In the first case, the two routers can attempt entanglement generation through the quantum communication channel. In the second case, each node along the path generates entanglement with a subsequent quantum router (pairwise entanglement generation) and then multiple entanglement swap operations are carried out to create an end-to-end entanglement.

Although the operation of entanglement swap is probabilistic for certain setups, for simplicity, in our models we regard it as a deterministic procedure. In further

chapters, two models will be distinguished: the on-demand model and the continuous model, based whether entanglement is generated only when a request has been received or it is already "pre-shared".

2.3 End-to-end entanglement creation in a quantum network

Before we delve into the study of various approaches for routing entanglement, it is important to separate phases of the end-to-end entanglement creation procedure. The main question that we would like to answer in this section is the following:

- What communication steps take place from the point in time when the demand comes from a node s until the point when end-to-end entanglement has been created between s and e?

As an answer to this question, we define the following phases of end-to-end entanglement creation [CRDW19]:

1. Path discovery
2. Entanglement reservation
3. Entanglement distribution

These phases are discussed without any requirements towards routing algorithms or entanglement generation schemes i.e. serve as a general list of entanglement creation steps. The first two phases belong to the network layer, whereas the third is executed by the link layer of the Quantum Internet stack. In the following section, we give further details for each of the phases.

2.3.1 Path discovery (Classical)

The goal of the path discovery phase is, to plan the path(s) between the source and the destination nodes. The path is specified by routing the demand and adhering to the requirements specified. It discovers the path according to the specific routing algorithm defined. In [CRDW19] distributed routing algorithms are proposed, where the next hop for a given path is specified based on the knowledge about the neighbours of the latest node. In this document, path discovery is carried out with a more centralised approach, the start node computes the path based on its current knowledge about the topology.

2.3.2 Entanglement reservation (Classical)

The entanglement reservation phase ensures that nodes along the discovered path are informed about the requirements of the demand, that the nodes allocate the needed resources and that no other demand will try to use these resources until the demand has been served. This means, that each node will :

1. either allocate the pre-shared entangled links with the specified requirements;
2. or allocate the link generation capabilities to serve the demand.

During our work, we assume that entanglement reservation takes place sequentially for each demand. This property will later be explained in more details.

2.3.3 Entanglement distribution (Quantum)

The entanglement distribution phase is the part of the end-to-end entanglement creation when the entangled links are provisioned on the quantum level. Two operations are performed between nodes along the selected path:

- Entanglement generation between intermediary nodes (if no pre-shared entangled links can be used);
- Pairwise entanglement swap between the intermediary nodes.

Once all the entanglement swap operations have finished, the end-to-end entangled link between source and destination is ready to be used for quantum communication.

Chapter 3

Graph theory and finding the shortest path

In this chapter, we discuss the definitions and ideas of graph theory that we will be using throughout this document. Basic concepts related to this field like the distance in a graph, the diameter of a graph, the neighbourhood of a vertex are described in section 3.1. Having introduced and explained the significance of these and other related terms, we discuss Dijkstra's algorithm for finding the shortest path in sections 3.2 and 3.2.3. In the subsequent section 3.3, we consider relevant results from complex network theory that have been achieved in the past years. After introducing the concept of random networks, we study the concept of the small world phenomenon. Then we discuss how Watts and Strogatz transitioned from a deterministically created network towards a random one. At the end of the chapter, we describe scale-free networks which will in coming chapters prove to be essential to our approach.

3.1 Graph theory

Graph theory has been an essential tool for representing networks ever since the 18th century. It allows a simple, yet powerful expression of the building blocks and the characteristics of networks. As we will later explain, to model quantum networks graph theory is essential. Consequently, we introduce the following relevant notations and definitions from the field.

3.1.1 Notations and definitions

First and foremost, we define the most fundamental tool from mathematics that allows us to represent networks, the graph.

Definition 3.1.1. *Graph [GR01]*
A graph G consists of a vertex set V, and an edge set E, where an edge is an unordered pair of distinct vertices of G.

A vertex and an edge in a graph are equivalent to a node and a link in a network respectively. For this reason, we will use these terms interchangeably. To ease the specification of a graph, the $G = (V, E)$ notation will be used.

Definition 3.1.2. *Undirected graph*
A graph is undirected if the set of edges consists of unordered pairs of elements of V.

According to the definition, considering an undirected graph $G = (V, E)$, an edge is a set of $\{u, v\} \in E$ such that $u, v \in V$ and $u \neq v$. We will be sticking to the notation of denoting an edge between vertices $v, u \in V$ as (u, v) instead of having the set notation of $\{u, v\}$. As a consequence, we regard (u, v) and (v, u) as the same edge in the graph. In an undirected graph, self-loops are not allowed, every edge consists of two distinct vertices.

Definition 3.1.3. *Adjacency*
Given a graph $G = (V, E)$, $u, v \in V$. *if* $(u, v) \in E$ *then vertex* u *is adjacent to vertex* v.

In undirected graphs the adjacency relation is symmetric, hence if $(u, v) \in E$ then $(v, u) \in E$ as well. Adjacent vertices are also called neighbours. Thus, we will be talking about the neighbours of a certain vertex.

Now that we have defined what graphs are, we can introduce further specific attributes related to each edge in a graph. One of such attributes is assigning a weight to each of the edges.

Definition 3.1.4. *Weighted graph*
A graph $G = (V, E)$ *with weight function* $w : E \to \mathbb{R}$ *defined on the set of edges is said to be a weighted graph.*

Weighted graphs can be useful for representing the cost of "using" edges. Most of the times in these cases we regard a sequence of edges and would like to know what the sum of the weights of the edges is. Such sequences called paths will prove to be extremely important.

Definition 3.1.5. *Path*
Given a graph $G = (V, E)$, *a path between two vertices* $s, e \in V$ *is sequence of edges such that* $path(s, e) = \{(s, v_1), (v_1, v_2), (v_2, v_3), ..., (v_n, e)\}$.

At times, we will be referring to vertices that are visited by a specific path and use the $v \in path(s, e)$ notation. Although this does not correspond to the definition of a path, as a path, according to its definition, contains edges. Consequently, we will be using the notation $v \in path(s, e)$, when $\exists u \in V : (v, u) \in E \vee (u, v) \in E$.

Now we know what paths are in a graph, can we always find one between the vertices? Given a graph, it is said to be connected if there is a path between any two of its vertices. We refer to the number of edges in a path as the *length* of the path. As mentioned earlier, we are curious about what the sum of the weights of the edges for a path is. The reason for this is that we would like to minimise it.

Definition 3.1.6. *Shortest path*
Given a graph G with a weight function and vertices $u, v \in V$, the shortest path between u and v is the path where the sum of the weights of the edges in the path is minimised.

If not specified otherwise, we will be using the weight function $w : E \to \{1\}$ in our work, that is each edge will have unit weight. Finding shortest paths between vertices is of vital importance to provide a certain optimal connection between vertices. We are also concerned about the length of such shortest paths.

Definition 3.1.7. *Distance*
Given a graph $G = (V, E)$, the distance between two vertices $u, v \in V$ is the length of the shortest path between u and v.

Now we can introduce a very important characteristic when describing the connectivity of a graph using the distance between any two vertices in a graph.

Definition 3.1.8. *Diameter of a graph*
Given a graph $G = (V, E)$, the diameter of the graph is:

$$diam_G = \max_{v \in V, u \in V} \{dist_G(u, v)\}$$

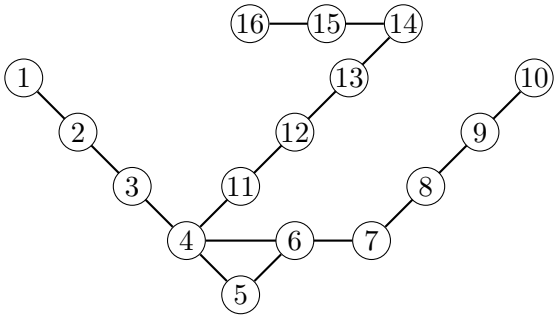
This definition can also be worded such that the diameter of a graph is equal to the shortest path with the biggest distance between any two vertices in a graph. For this reason, knowing the diameter of a network and how it might change proves to be very important for determining the shortest paths in a network. If it is obvious which graph we are seeking the diameter, then we just denote the value at $diam$. We will also see that having networks with a low diameter will be very much desirable for routing in quantum networks.

So the diameter of a graph shows the maximum distance between any two vertices in the graph. Sometimes, however, in certain graphs, the diameter might prove not to be the quantity that we are looking for. Consider a graph where there are certain nodes between which the distance is significantly bigger than the distance between other nodes (see figure 3.1). In such a case the diameter of the graph only tells us something about a certain number of extreme cases, but we might be interested in an average distance in the graph.

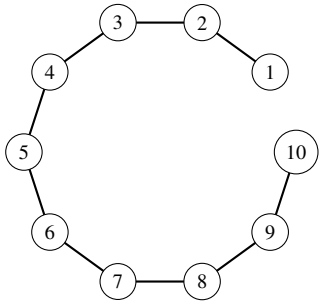
Definition 3.1.9. *Average distance*
Given a graph G with N vertices, the average distance or average path length in the graph is:

$$\langle d \rangle = \frac{1}{N(N-1)} \sum_{v,u \in V} dist_G(u, v)$$

As we have seen distance measures tell us how far apart vertices lie in a graph, and what length is for each path connecting them. Another important measure is the degree of a vertex.



(a) A graph with 16 nodes such that $\langle d \rangle = 4.35$, $diam = 11$



(b) One dimensional lattice with 10 nodes such that $\langle d \rangle = 3.67$, $diam = 9$.

Figure 3.1: Two examples for graphs where their diameter and average distance between two arbitrary nodes differ significantly.

Definition 3.1.10. *Degree of a vertex*
Given a graph $G = (V, E)$, the degree of a vertex is the number of edges whose start or end is the vertex:

$$\forall v \in V : deg(v) = |\{u \in V | \{v, u\} \in E\}|$$

Lemma 1. *(Handshaking lemma)*
Given a graph G, the following degree sum statement holds:

$$\sum_{v \in V} deg(v) = 2|E|$$

We see now, that we can determine the number of edges for a vertex in the graph. This is, however, specific to a vertex, and vertices can differ very much in a graph-based on how connected they are. It seems to be reasonable to consider the average edge count per vertex in a graph.

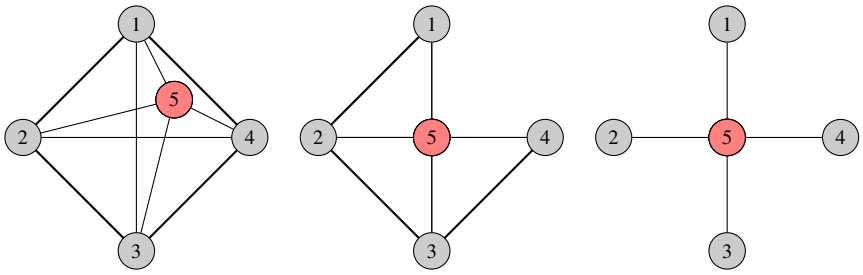
Definition 3.1.11. *Coordination Number*
Given a graph, the coordination number z is the average number of edges per vertex, i.e. the average degree of a vertex in a graph.

The coordination number provides the average degree in the graph. It does not tell us specifically, however, what is the distribution of degrees between the vertices in the graph. How many vertices have $deg(1), deg(2), ..deg(N)$ and how are these numbers related to the number of all edges?

Definition 3.1.12. *Degree distribution*
For a graph $G = (V, E)$, $N = |V|$, the probability distribution determined by $p_k = \frac{N_k}{N}$, where N_k is the number of degree-k nodes is called degree distribution.

Now that we have seen what the coordination number and degree distribution are, and have an idea about the edge count on average in a graph, we are interested in how well neighbours of a certain vertex are connected. In determining this, the concept of clustering and the local clustering coefficient of a vertex helps us out.

Definition 3.1.13. *Local clustering coefficient [DJW98]*
Given a graph G, a vertex $v \in V$, the local clustering coefficient of v is defined as: $Cl_v = \frac{2L_v}{deg(v)(deg(v)-1)}$, where L_v represents the number of links between the deg(v) neighbors of node v.



(a) $Cl_5 = 1$ (b) $Cl_5 = 0.5$ (c) $Cl_5 = 0$

Figure 3.2: The local clustering coefficient (Cl_5) of node 5 in three different examples (as shown in [Bar16]). In all three of the examples, node 5 has 4 neighbours. In the first graph there is an edge between any two of these neighbours, resulting in $L_v = 6$. In the second example the edges {1,2}, {2,3} and {3,4} are present, hence $L_v = 3$. In the last example none of the neighbours are connected. Using the value of L_v, the clustering coefficient can be computed as described in definition 3.1.13.

If for a certain node v the clustering coefficient is 0, then none of its neighbours has any links among themselves. On the other hand, if the clustering coefficient is 1, then the neighbours of v are all connected. The probability that two neighbours of v are connected is equal to C_v [Bar16].

Definition 3.1.14. *Edge Betweenness Centrality*
Given a graph $G = (V, E)$, the edge betweenness centrality of an $edge \in E$ is

$$c_B(edge) = \sum_{s,e \in V} \frac{\sigma(s, e|edge)}{\sigma(s, e)} \tag{3.1}$$

where $\sigma(s, e)$ denotes the number of all the shortest paths between vertices s, e and $\sigma(s, e|edge)$ denotes the number of shortest paths between vertices s, e such that $edge$ is in the shortest path.

3.2 Dijkstra's algorithm

3.2.1 The problem

Dijkstra's algorithm provides an approach to compute the shortest paths in a weighted graph G, where the edge weights in the graph are non-negative. For further description of the algorithm we refer to [CLRS09].

Given a graph G, weight function w and source vertex s, Dijkstra's algorithm computes a shortest path from s to each vertex in $V \setminus \{s\}$. For this reason, if possible, it solves the single-source shortest path problem.

3.2.2 The algorithm

Dijkstra serves as the main function, which uses the Relax and Initialize-Single-Source functions as subfunctions.

We use the adjacency list graph representation of an undirected, weighted graph $G = (V, E)$. The weight function is denoted by $w : E \to \mathbb{R}$. We assign a unique label from $\mathbb{N}$ to each element of the set of vertices V.

Furthermore, G has the following attributes:

- V: an array that contains the indices of vertices found in graph G
- E: an array that contains the edges found in graph G as pairs of vertices from V
- Neigh: an array that contains a list for each vertex from G.V. Each list stores the neighbours for the specific vertex, i.e.:

$$\forall s \in V : G.Neigh[s] = \{e \in V | \exists (s, e) \in E\}$$

For each vertex $s \in V$ we introduce the following two attributes:

- d: distance of vertex e from vertex s in graph G;
- π: parent vertex of e in the discovered path towards s.

Furthermore, a minimum priority queue denoted as Q will also be used. This data structure has an $Extract - Min$ method defined on the queue that removes and returns the element of Q with the smallest key. In the case of Dijkstra, the key corresponds to the value of d for each node in Q.

Using these notations the following algorithms describe Dijkstra's algorithm.

Algorithm 1 Relax(u, v, w)

```
1: if v.d > u.d + w(u, v) then
2:     v.d = u.d + w(u, v)
3:     v.π = u
```

Algorithm 2 Initialize-Single-Source(G, s)

1: **for** $each\ vertex\ v \in G.V$ **do**
2: $\quad v.d = \infty$
3: $\quad v.\pi =$ NIL
4: $s.d = 0$

Algorithm 3 Dijkstra(G, w, s)

1: $Initialize - Single - Source(G, s)$
2: $S = \emptyset$
3: $Q = G.V$
4: **while** $Q \neq \emptyset$ **do**
5: $\quad u = Extract - Min(Q)$
6: $\quad S = S \cup \{u\}$
7: $\quad$ **for** $each\ vertex\ v \in G.Neigh[u]$ **do**
8: $\quad\quad Relax(u, v, w)$

3.2.3 Shortest path algorithm

As later we will be interested in finding the shortest path between two specific vertices in a graph, we also introduce the ShortestPath algorithm to solve the single-pair shortest path problem. Using Dijkstra's algorithm it returns the set of vertices for the shortest path between a source and destination vertex.

Algorithm 4 Shortest-Path(G, w, s, e)

1: $Dijkstra(G, w, s)$
2: $Path_{s,e} = \emptyset$
3: $u_{curr} = e$
4: **while** $u_{\text{curr}} \neq$ NIL **do**
5: $\quad Path_{s,e} = Path_{s,e} \cup \{u_{\text{curr}}\}$
6: $\quad u_{\text{curr}} = u_{\text{curr}}.\pi$
7: **return**($Path_{s,e}$)

No-path property

In case there exists no path between s and e, after running $Dijkstra(G, s, e)$ the following equation holds: $e.d = \infty$ (∞ the arbitrarily chosen extreme value). After running $Shortest - Path(G, w, s, e)$, s is not in the set of $Path_{s,e}$.

3.3 Complex network theory

Complex network theory deals with the modelling of real systems found in practice which bear characteristics alike to simple network representations such as lattices and random graphs. Although a considerably new field of research, it had a great impact in all kinds of networks be its computer networks, neural networks or social networks. Consequently, in this section, we discuss the results of complex network theory that prove to be closely related to the building of quantum networks and routing in the quantum network. In our work, we will be studying one-dimensional lattice (ring) topologies or networks constructed from such graphs. Next to the concepts described in this section, for a more detailed description of complex network theory we refer to [Bar16].

3.3.1 Random networks

The origins of complex network theory date back to the random graph model created by Pál Erdős and Alfréd Rényi in 1959 [PE59]. Erdős and Rényi fixed the number of nodes and the number of links in a graph and then constructed the graph randomly. Such a graph, called the Erdős-Rényi graph can be specified by $G(N, L)$ (N: number of nodes, L: number of links in the graph). Another approach, that was introduced by Gilbert, fixes the probability with which two arbitrary nodes are connected. Thus, the graph is given by $G(N, p)$. In this definition N: denotes the number of nodes, while p the probability of two nodes being connected. [Gil59]. Following this second model, we can now define the concept of random networks.

Definition 3.3.1. *Random network*
A random network consists of N nodes where each node pair is connected with probability p.

In a random network of $G(N, p)$, the degree distribution follows a binomial distribution with the following probabilities: [Bol01]

$$p_k = \binom{N-1}{k} p^k (1-p)^{N-1-k} \tag{3.2}$$

where p_k is the probability that a node chosen at random has degree k.

Assuming that $\langle k \rangle \ll N$ (a property displayed by many real-world networks) we can approximate the degree distribution with the following Poisson distribution:

$$p_k = e^{-\langle k \rangle} \frac{\langle k \rangle^k}{k!} \tag{3.3}$$

where $\langle k \rangle$ is the coordination number in G.

We can thus describe the degree distribution of a random network with equations 3.2 and 3.3.

Having introduced what is meant by random networks and what their degree distributions are, we can now discuss an important empirical result related to random networks.

3.3.2 The small-world phenomenon

One of the simplest of networks that is apparent in our everyday life as well, is the people around us, the people with whom we interact with and know. However, of course, each person can only know a limited number of other people, so it is worth observing the entirety of such relationships. Following this idea, let us consider a network with each person as a node, while there is a link between two people if they know each other. In the graph representation of such a network, we would have over 7 billion nodes. What could be the separation (distance) between any two people on our planet? Well, to answer this question we have to consider the graph of this network.

First, let us take a one-dimensional lattice as an example. In such a topology the diameter of the graph scales linearly with the number of nodes in the graph (see figure 3.1b for an example). If we then, consider a two-dimensional lattice as the underlying graph for the network, we will observe that the diameter scales with the square root of the number nodes.

These two scaling factors call for the intuition that by adding new nodes to the graph, the diameter is also increased significantly. Say we take the example of all the people living in the world. Then, the degree of separation (diameter in this example) between people has a comparable "speed" to the increase of the world population.

This intuition was proven wrong following the empirical result coming from Stanley Milgram [Mil67]. Through Milgram's experiment something called as the idea of living in a "small-world" was set to challenge and the study of such "small-world" networks started to get traction. This study introduced a concept which later became known as the six-degrees of separation.

Here we assess the definition of what exactly the "small-world phenomenon" means.

Definition 3.3.2. *Small-world phenomenon*
For a random network G of N nodes, coordination number $\langle k \rangle$, the diameter of the network is:

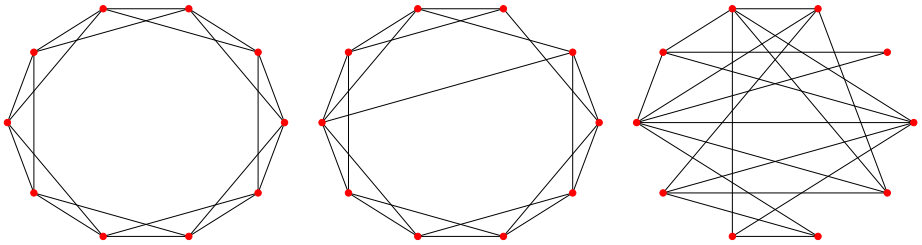
$$diam_G \approx \frac{logN}{log\langle k \rangle} \tag{3.4}$$

Let us now return to the example of the network comprised of the individuals on Earth. Using equation 3.4, $N \approx 7 * 10^9$ and $\langle k \rangle \approx 10^3$, we end up with the value of $\langle d \rangle \approx \frac{ln(7*10^9)}{ln10^3} = 3.28$ [IdSP78]. This value proves to be quite close to the result of Milgram's experiment.

Although the small-world phenomenon gives a close estimate to the result coming from Milgram's experiment, it has been observed that many other properties (such as degree distribution or local clustering coefficient) true for real-world networks do not correspond to that of random networks. For this reason, although the small-world phenomenon was the key result, it did not imply that random graphs could serve as suitable models for real-world networks.

3.3.3 Watts-Strogatz Model

In 1998, Duncan J. Watts and Steven Strogatz came up with a new way of incorporating randomness in the models attempting to depict real-world networks [DJW98]. They managed to construct graphs which showed the small-world property and had an average clustering coefficient that was higher than for random networks. They presented a procedure for creating such a graph. First, one starts from a regular one-dimensional lattice created in a deterministic manner: each node is connected with its two neighbours as well as with its k nearest neighbour where k is a predefined value. Then, determined by a fixed parameter p, each link is rewired with probability p to a vertex sampled uniformly at random. This procedure significantly decreases the distance between nodes in the network. With this procedure, when p takes intermediate values between 0 and 1, then graphs showing the small-world phenomenon were created, while a higher clustering was achieved than in random networks. Yet, certain properties of real-world networks were still not incorporated in these models.



(a) $p = 0$ (b) $p = 0.05$ (c) $p = 1$

Figure 3.3: Three graphs obtained by applying the procedure suggested by Duncan J. Watts and Steven Strogatz [DJW98]. We start from a one-dimensional lattice of $N = 10$ nodes, where each of the nodes are connected with $k = 4$ nearest neighbours. Then, each link is rewired with a certain probability p. The three examples show how graphs are constructed while increasing the randomness. First, we start from a highly clustered ring lattice and then end up with a random network showing the small-world phenomenon. For intermediate values of p, the graph created with this procedure shows the small-world phenomenon and is also highly clustered like a regular graph.

3.3.4 An algorithmic perspective

In 1999, Jon Kleinberg further explored the topic of small-world networks [Kle00]. In his work, he argued that Milgram's experiment did not only show that people are separated by small distances. It also showed that people (who can be represented as nodes in a network) may find short paths between each other in the network even if separated by great physical distance. For this, each node uses its local information about the location of its neighbours. The Watts-Strogatz model was used to create a fitting model for the algorithmic approach. A network of nodes using a two-

dimensional grid was constructed. Edges were added between nodes regardless of the location of such nodes in the grid, resulting in "long-range contacts". One of the key results of the paper was showing that there is a decentalized algorithm used in this model for which the expected delivery time of a message is at most $\alpha(log\ n)^2$, where n is the number of nodes in the network and α is a constant independent of n. Once such a model was constructed, a decentralized algorithm resembling Milgram's experiment was applied to find a short path between a source and destination with a high probability.

3.3.5 Scale-free networks

A property of networks that proved to be of vital importance was the degree distribution. While random networks carried a basic intuition and the Watts and Strogatz model gave an approach of how to have high clustering and add the small-world property in a graph, these networks were still different from those of real-world networks. The key towards the next step in the field was the realisation that the degree distribution of these networks was not following a Poisson distribution, but another one.

Definition 3.3.3. *Scale-free property[HJB99]*
A scale-free network is a network whose degree distribution follows a power-law:

$$p_k \approx k^{-\gamma} \tag{3.5}$$

Equation 3.5 is called a power law distribution and the exponent γ is its degree exponent. The first network that was empirically shown to demonstrate such a degree distribution was the World Wide Web in 1999 [HJB99]. Later that year, power-law relations found in the Internet topology were reported in [MFF99]. Ever since several other real-world networks were also reported to have a power-law degree distribution. However, it is important to note that not all such networks bear this property.

3.3.6 Summary

In this chapter first, we gave the relevant concepts and results of the field. First, we introduced a basic intuition as to how real-world networks were regarded initially. The use and study of random graphs were discussed. Then, we described what the small-world property is and how its study emerged through an experiment. Following that, we discussed a model introduced by Watts and Strogatz, that attempted to model real-world networks. Using this model and small-world networks, Jon Kleinberg's result was discussed. At the end of the chapter, the scale-free networks were introduced, networks whose degree distribution correspond to many real-world networks such as the World Wide Web and the Internet.

These concepts and results have influenced [CRDW19] and the work presented in this thesis. One of the key ideas is constructing network models with "long-

range" neighbours for nodes representing quantum networks. This effectively decreases the diameter of the network, resulting in the length of shortest-paths be smaller between arbitrary nodes. Characteristics of small-world and scale-free networks may be observed in the models introduced in the coming chapter.

Chapter 4

Quantum network models

In this chapter, we adapt the customary tools of studying the behaviour of networks to our specific case of quantum networks. For this, we use definitions and concepts from graph theory.

First, we look at why we need to create new models for quantum networks. Following that, we introduce a time model according to which we conduct our analysis. Then, we adapt the models of quantum networks based on the event of link generation. Based on how links are generated to serve the incoming demand, we distinguish between an on-demand and a continuous model.

Later in the chapter, a two-layered network model is introduced which describes the entanglement between nodes. Physical graphs consist of quantum routers and physical quantum communication channels. These graphs serve as an underlying network. Upon introducing them, we explain virtual graph models. They include entangled links and make up an overlay network for quantum networks.

4.1 Motivation for model creation

In this section, we describe the motivation behind using different models for quantum networks. Although they are inherently different from classical networks, certain similarities can be found.

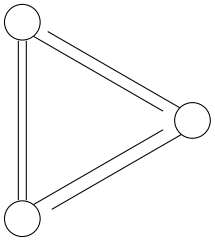
It is important to describe the time communication (quantum as well as classical) takes in a quantum network. As an example, quantum routers need to communicate via classical messages between each other. Other events, such as elementary link creation, also take a considerable amount of time. How shall we include time in our models? What could be the basic time unit in our models?

Assume that we already have a certain definition of time unit in a quantum network. We can then use entanglement to transmit quantum information between nodes. For this, entanglement is demanded by the sender. The process of entanglement generation is summarized in figure 4.1. When should such an entanglement be created?

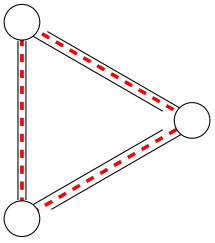
To create entanglement, physical quantum communication channels are needed

between certain nodes. The nodes not connected by a physical quantum communication channel can also share entanglement as a result of entanglement swap. This way a "shortcut" is created in the network (as described in [ESW16]). For which nodes is it worth having these shortcuts? Also, when creating shortcuts, the network made up of physical quantum communication channels and entanglement between nodes differs. What type of network topology could be used for analysis?

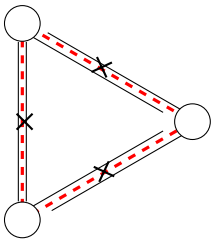
In this chapter provide answers to these questions by introducing quantum network models.



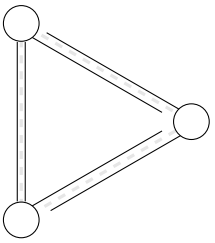
(a) Quantum routers prepare to generate entanglement using physical quantum channels. (For more details see 2.2.3).



(b) Entangled links are generated between the routers.



(c) The entangled links are used for quantum teleportation or they decohere over time.



(d) Once the entangled links have been used or decoherence took place, they need to be regenerated.

Figure 4.1: Entanglement generation and consumption between quantum routers.

4.2 Quantum network

In each of the following models, we assume that we have a quantum network with quantum routers and that some are connected with physical quantum channels.

Apart from this, we make the following additional assumptions for the routers (nodes) of the network:

- they can use the classical communication infrastructure to send and receive classical messages,
- they know about the physical quantum topology,

- they have a certain initial view about the shared entanglements in the network (this point will be explored more in detail in chapter 6).

Using these capabilities, quantum routers want to use entanglement to serve requests. We will be referring to these requests as demand appearing in the network. The frequency of such demands appearing at specific nodes and the overall distribution of demands in the network greatly depends on the structure of the quantum network.

4.3 Discrete time model

To be able to introduce the dynamic nature of the network, we are considering a discrete time model, as introduced in [CRDW19]. We assume that the distance between any two quantum routers can be upper bounded by $dist_{phys}$. One time step in our model is equal to $\frac{dist_{phys}}{c}$ time units, where c is the speed of light in a communication channel.

4.4 Demand based entanglement generation models

To analyse the behaviour of quantum networks, we need to create models based on the event of entanglement generation in the network. The created entanglement can be then used for various applications. There can be different models determined based on when entanglement is created.

4.4.1 On-demand model

The basic entanglement generation model used to represent a quantum network is the on-demand model. The key point for the on-demand model is that entanglement is generated between two nodes only once there is a need for it. Once the demand comes, a path made up of quantum communication channels is determined between the source and the destination node. After this, pairwise entanglement is generated between the nodes. This is followed by entanglement swaps performed at the intermediary nodes to create end-to-end entanglement.

Let us take an example. Consider two nodes connected by a physical quantum channel. Entanglement can be created using this channel. Each of the two nodes can send quantum information to the other one, once entanglement has been created. In the on-demand model, the process of entanglement creation is only initiated, once one of the nodes wants to transmit quantum information. As soon as such a shared entanglement has been created, it can be used for quantum teleportation.

4.4.2 Continuous model

Since the creation of end-to-end entanglement in the on-demand model might take a significant amount of time (in the subsequent chapter we study this more thoroughly), entangled links could potentially be generated even before the demand comes. We will be referring to this process as "pre-sharing" entanglement, as they are created in advance. Pre-shared entangled links can not be stored for an unlimited amount of time unfortunately due to the decoherence process that takes place. Consequently, after a certain amount of time, entangled links are not available anymore. We use the concept of *threshold time* as used in [CRDW19] and denote it with T_{th}. Hence, after T_{th}, the generation process of entangled links commences again.

Let us consider an example of the continuous model. Consider two nodes connected by a physical quantum channel. In the continuous model, the entanglement generation process between the two nodes is ongoing regardless of incoming requests. Let us assume that such a link has been created with a T_{th} value. If we a demand comes between the time of generation and T_{th}, the created entanglement can be used for quantum teleportation. This way no entanglement needs to be generated.

4.5 Physical graphs

In the previous section, we have considered models based on when entanglement generation takes place. In this section and the next one, we look at models needed to describe the state of the quantum network.

The quantum network infrastructure can be modelled as a simple static graph where the nodes are quantum routers and the edges are quantum channels.

Definition 4.5.1. *Physical graph[CRDW19]*
A physical graph is an undirected simple connected graph $G_{Ph} = (V, E_{Ph})$ where V represents quantum routers and E_{Ph} represents physical quantum channels.

In such a graph we can define physical neighbours of nodes.

Definition 4.5.2. *Physical neighbour[CRDW19]*
Given a physical graph $G_{Ph} = (V, E_{Ph})$, two nodes $v, u \in V$ are physical neighbours of each other, if $\exists e \in E_{Ph} : e = \{(v, u)\}$.

Physical graphs are also used to represent the on-demand model. A physical graph does not describe the current state of entangled links in the network. It is also static in nature, hence its diameter is constant.

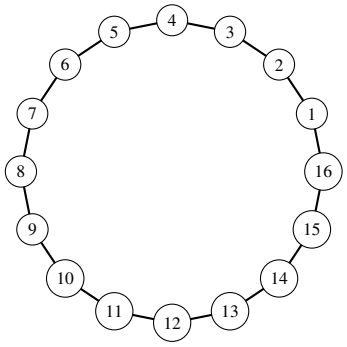


Figure 4.2: Physical graph of $|V| = 16$ nodes in a ring topology. The nodes and the edges represent quantum routers and quantum channels.

4.6 Virtual Graphs

As physical graphs are models of the physical channels capable of creating entanglement between nodes, we have not included entangled links in such graphs. We include entangled links in graphs called virtual graphs [CRDW19]. Virtual graphs contain virtual links which represent pre-shared entanglement in quantum networks. The pre-shared entanglements are then later recreated in a new time window. For these reasons, virtual graphs are equivalent to the graph representation of the continuous model.

Definition 4.6.1. *Virtual graph[CRDW19]*
Given a $G_{Ph} = (V, E_{Ph})$, a virtual graph is a graph $\mathcal{G} = (V, \mathcal{E})$ where V represents the set of quantum routers and $\mathcal{E}$ represents entangled links.

Later we will be referring to the edges comprising the virtual graph as *virtual edges* (virtual links), $u, v \in V$ connected by a virtual edge will be referred to as *virtual neighbours*. It is worth noting that $\mathcal{E} \cap E_{Ph} = \emptyset$. This comes from the fact that the edges in a physical graph are physical quantum communication channels, whereas edges in a virtual graph are entangled links.

Although virtual edges will create neighbours in the virtual graph, we will be referring to their lengths based on the properties in the underlying physical graph.

Definition 4.6.2. *Virtual edge length*
The length of an edge in a virtual graph between v_1, v_2 is $dist_{G_{Ph}}(v_1, v_2)$.

Definition 3.1.7 was used in the physical graph as a distance metric. We will later be referring to a virtual edge whose length is greater than 1 as a *long virtual edge*. As the generation of such long virtual edges can be quite costly in terms of time, it is worth posing a threshold on the length of virtual links.

Definition 4.6.3. *Distance threshold[CRDW19]*
Given a quantum network, the physical distance between any two virtual neighbours is upper bounded by the distance threshold (d_{th}).

An equivalent statement connected to the definition of the distance threshold is that given a virtual graph $\mathcal{G}$ constructed with a certain distance threshold, no virtual edge is longer than the value of d_{th}.

Defining a distance threshold and a specific procedure for constructing virtual graphs (see next section) will be useful to set the diameter of a virtual graph. Let us consider the example of a ring topology with 16 nodes and a distance threshold of 2. Then, construction of virtual graphs could be defined such that nodes are connected along the physical graph and each node is also connected to nodes at physical distance 2.

4.6.1 Deterministic virtual graphs

Deterministic virtual graphs are constructed according to a specific procedure of adding virtual edges. The motivation for construction comes from the fact that by reducing the diameter of the graph, a more connected graph can be used [Kle00]. It is hence worth to transition from a ring topology towards a more connected network. This in principle shall offer advantage while doing routing.

In order to create a deterministic virtual graph with n number of nodes, a distance threshold of d_{th} and a $maxd_{th}$ such that $d_{th} \leq maxd_{th}$, the following steps are carried out:

1. Neighbouring nodes of the physical graph are connected by a virtual edge,

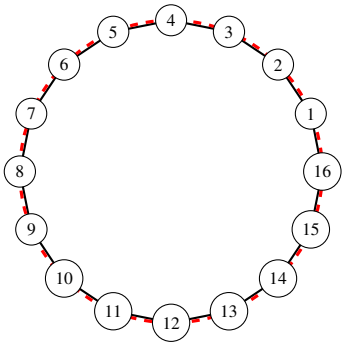
2. Long virtual edges starting from nodes with odd-numbered labels are added,

3. Additional $log_2 maxd_{th} - log_2 d_{th}$ many virtual edges between nodes labeled as x and $(x+1)\ mod\ n$ are added, where $x = (y * 2^i) + 1, i \in \mathbb{N}_0, y \in \mathbb{N}_0$ and y is 0 or odd.

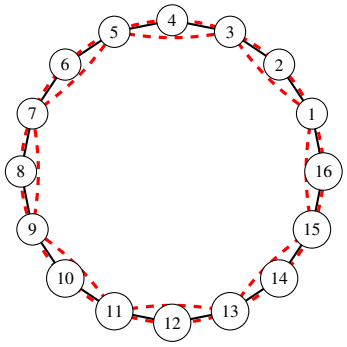
It is worth mentioning that by following this strategy, for the case $log_2 maxd_{th} = log_2 d_{th}$, there is exactly one virtual edge connecting any two nodes in the virtual graph.

For an input node of label $x \in \mathbb{N}$, the number of long edges is given by the following function:

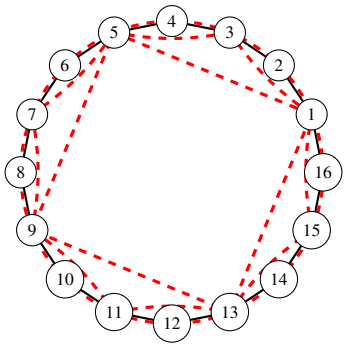
$$longcount_{d_{th}}(x) = \begin{cases} 0 & \text{if } x \in \mathbb{N} \text{ and } x \text{ is even} \\ 2log_{dth} & \text{if } x = 1 \\ min(2i, 2log_{dth}) & \text{if } x = (y * 2^i) + 1, i \in \mathbb{N}_0, y \in \mathbb{N} \\ & \text{and } y \text{ odd} \end{cases}$$



(a) $d_{th} = 1, maxd_{th} = 1$



(b) $d_{th} = 2, maxd_{th} = 2$



(c) $d_{th} = 4, maxd_{th} = 4$

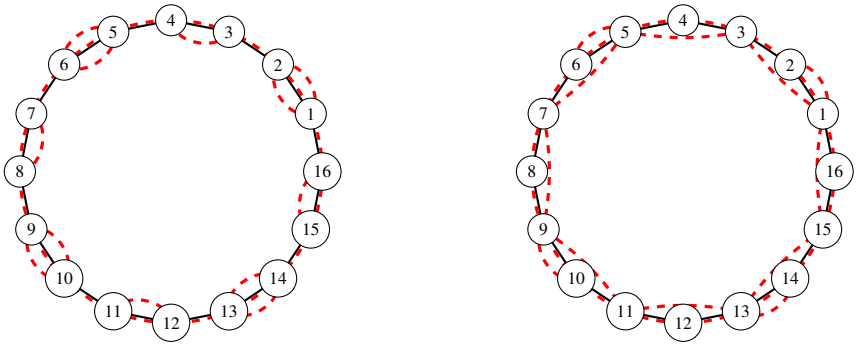
Figure 4.3: Quantum networks of $|V| = 16$ nodes. Each is made up of a physical graph (continuous black line) and a deterministically constructed virtual graph (red dashed line) determined by the distance threshold (d_{th}).

The number of long edges for a node with labels $x = 1$ or $x = 2^i * y + 1$ (where $i \in \mathbb{N}_0, y \in \mathbb{N}$ and y is odd) are connected with nodes labeled as $(x + 2)\ mod\ n, (x+2^2)\ mod\ n, ..., (x+2^{min(i,log_2 d_{th})})\ mod\ n$ and $(x+n-2)\ mod\ n, (x+n-2^2)\ mod\ n, ..., (x+n-2^{min(i,log_2 d_{th})})\ mod\ n$ [CRDW19].

For virtual graphs constructed this way, we have $|E| = N + \sum_{x=1}^{N} longcount_{d_{th}}(x)$.

With this construction, we increase the degree of every other node in the graph, eventually creating hubs in the virtual graph.

The coordination number (avereage degree of a node) of a deterministic virtual graph can be obtained the following way: $z = \frac{N + \sum_{x=1}^{N} longcount_{d_{th}}(x)}{N}$.



(a) $d_{th} = 1, maxd_{th} = 4$ (b) $d_{th} = 2, maxd_{th} = 4$

Figure 4.4: Quantum networks of $|V| = 16$ nodes, $maxd_{th} = 4$ and $d_{th} < maxd_{th}$. Extra links are added according to step 3.

4.6.2 Random virtual graphs

We can further construct virtual graphs starting from a some pre-defined topology and then by adding long edges randomly. The random virtual graph construction method described in this section is adapted from [CRDW19]. The graph creation described in section 3.3.3 results in graphs with the small-world phenomenon and also higher clustering than with totally random graphs. The random graph creation explained in this section follows steps similar to the Watts-Strogatz model.

In our case of random virtual graph creation, long virtual edges are added according to a power-law distribution. Any node u chooses another node v (such that $\mathrm{dist}_{G_{\mathrm{ph}}}(u, v) > 1$) as a neighbour with the following probability:

$$\mathrm{P}_{\mathrm{choose}}(u, v) := \begin{cases} \frac{1}{\beta_u} \frac{1}{\mathrm{dist}^{\alpha}_{G_{\mathrm{ph}}}(u,v)} & \text{if } \mathrm{dist}_{G_{\mathrm{ph}}}(u, v) \leq d_{\mathrm{th}} \\ 0 & Otherwise \end{cases}$$

where $\beta_u = \sum_{v' \in V} \frac{1}{\mathrm{dist}^{\alpha}_{G_{\mathrm{ph}}}(u,v')}$ and $\alpha > 0$.

In our models, each node chooses $k = log_2 d_{th}$ many long neighbours with replacement according to $\mathrm{P}_{\mathrm{choose}}$.

The graphs that are created this way are scale-free (section 3.3.5) and provide us with models similar to real-world networks.

4.7 Summary

In this chapter, we have introduced the already existing quantum network models that will prove to be relevant during our observations. We have taken a discrete time model based on classical communication latency between quantum routers, enabling the quantification of the duration of events in a quantum network. We have also adapted the on-demand and the continuous models to describe when entangled link creation takes place with respect to the arrival of demand. The concepts of

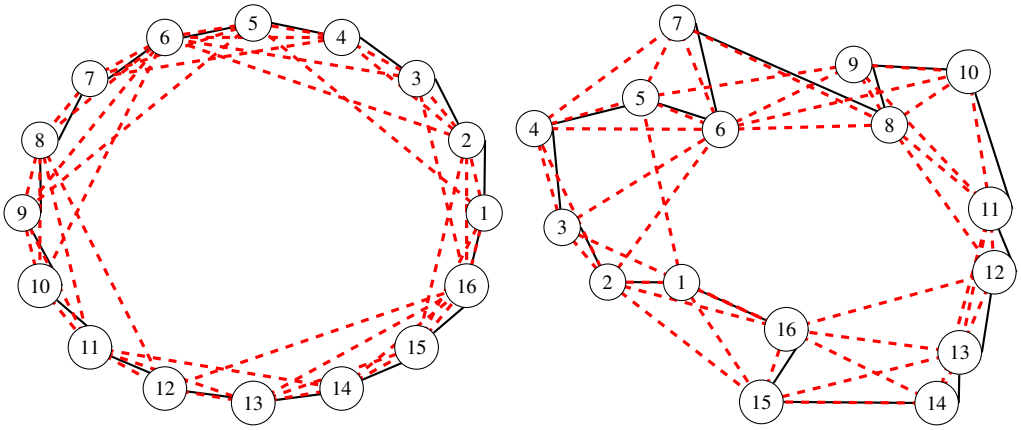


Figure 4.5: Two representations of a quantum network with random virtual graph of $|V| = 16, d_{th} = 4$ in a ring topology. Every node is connected with $k = 2$ many long neighbours based on a probability following a power law-distribution.

physical graphs and virtual graphs were discussed with relevant definitions such as the distance threshold in a virtual graph. At the end of the chapter, strategies of deterministic and random virtual graph generations were discussed with graphic examples.

Chapter 5

Entanglement routing in a quantum network

5.1 Introduction

In the previous chapter, we have introduced the existing models and techniques which are used to model quantum networks. In this chapter, we will see the need to introduce new ones, to make our analysis more realistic.

After phrasing certain reasons for coming up with a more realistic model, we include the concept of time and look at our network. The emerging field of analyzing temporal networks provides promising tools. They enable specifying graphs that change over time. Hence, in section 5.2, we introduce the mathematical concepts and terminologies from temporal networks used to describe the evolution of dynamic graphs. Later, we define quantum networks using these concepts and give a new definition in such a network.

In section 5.3 we phrase what pathfinding means in a quantum network and how it is related to routing in a quantum network.

Having formulated what a quantum network is using temporal graphs and what routing in such a network means, in section 5.3.2 we take a look at a specific example. We assume that a fixed number of requests are served in the network, paths are found for them and observe how the network changes. We also highlight the fact that for a more realistic demand model, traffic engineering needs to be carried out.

At the end of the chapter, in section 5.4 we look at the challenges faced when solving the problem of routing in a quantum network. We choose time as the main metric of interest for end-to-end entanglement creation. Each phase of this procedure is considered separately to determine the one that acts as a bottleneck for the entire process.

5.2 Quantum networks as temporal graphs

5.2.1 Approach for a new model creation

In our work, we will be looking at the change in the network topology over time. This would be necessary for modelling the entanglement generation and consumption taking place in a quantum network. For this reason, it will be important to see at which point in time we consider the network, hence the notion of time will play a key role. This serves as a motivation to shift from using the existing approaches of network theories. In most of the practical scenarios, the network does not change with time very frequently. So, one can easily represent this type of network using static graphs. However, these static graphs are not a good abstraction for dynamic graphs.

In the continuous model of quantum networks, the links between nodes are shared EPR pairs. They can be used for quantum teleportation or can decohere over time until they are no longer useful for quantum communication. These factors make quantum networks very dynamic. Given a set of EPR pairs in the network, each has its time of creation and will decohere at different times.

Suppose that some of the EPR pairs are used for quantum teleportation. In such a case, they are consumed and the representation of the network should be able to reflect this change. How could one use a simple graph to represent the behaviour? One solution would be creating a new graph for each change of the network, however, then a specific state of a point in time would be represented. It would be very useful if we would have a way of describing how the networks evolve.

To provide a solution to this problem, we make use of the already developing field of temporal networks. The main goal of this field is to encompass characteristics of the network in terms of time and introduce functions related to the network which take a point in time as an input. Therefore, in the following sections, we introduce and use definitions and ideas that could encompass the dynamic features of quantum networks. The same concepts proved to have been useful for describing evolving networks connected to other research areas such as health sciences, mobile communication networks and social networks.

5.2.2 Temporal graphs

As mentioned earlier, we would like to have a model where we can seamlessly use points in time to describe the state of the network.

Temporal networks and temporal graph algebra will be useful for that and eventually for defining quantum networks. Our motivation in introducing such concepts stems from the fact that we would like to describe the state of quantum links at specific points in time [MS17].

In our model, the points in time will have limited precision and have discrete time values from $\mathbb{N}$.

First, however, we define on which set of points in time we would like to observe the evolution of our quantum network.

Definition 5.2.1. *Time interval*
Given a $t_{init} \in \mathbb{N}$ *and* $n \in \mathbb{N}$*, the* $[t_{init}, t_{end}] = \{t_{init}, t_{init}+1, t_{init}+2...t_{init}+n = t_{end}\}$ *set of points in time is a time interval.*

We will be calling t_{init} *initial time*, whereas t_{end} as the *end time* of the time interval. Later we will be working with certain time intervals that are bounded by a threshold value. The reason for this is that in practice we know that no entangled link in a quantum network is permanent. A certain value (threshold value) can be determined as the maximum time an entangled link is available. The motivation behind introducing such a threshold time is that we can model the decoherence process of a quantum link using it. Therefore it makes sense to introduce time intervals that are determined by certain threshold values.

Definition 5.2.2. *Time window*
A time interval $[t_{init}, t_{end}]$*,* $[t_{init}, t_{end}]$ *is said to be a time window with respect to a* $T_{th} \in \mathbb{N}$*, if* $|t_{init} - t_{end}| \leq T_{th}$*.*

As we can see, the threshold value will determine the length of a time window (or the time for which entanglement is shared between two nodes in practice) and plays a key role in the life cycle of quantum networks.

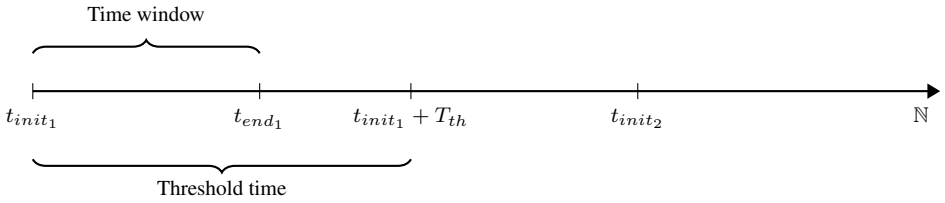


Figure 5.1: Representation of the linearly ordered time domain mapped to $\mathbb{N}$ with regards to the concepts of time window and threshold time. We denote the initial time in a time window by t_{init}, the end of the time window by t_{end}.

Now that we have a clear idea of what intervals of time we are interested in, we can adapt the definition of a temporal graph from temporal network literature.

Definition 5.2.3. *Temporal Graph[MS17]*
A temporal graph is a 6-tuple $\mathcal{G} = (V, \mathcal{E}, L, \Gamma, \xi^T, \lambda^T)$ *where:*

- *V is a finite set of nodes*
- $\mathcal{E}$ *is a finite set of edges*
- *L is a finite set of property labels*
- $\Gamma : \mathcal{E} \rightarrow (V \times V)$ *is a total function that maps an edge to its source and destination nodes*

- $\xi^T : (V \cup \mathcal{E}) \times \mathbb{N} \to \{0, 1\}$ *is a total function that maps a node or an edge and a point in time to a Boolean value, indicating the existence of the node or edge at the specific time; and*
- $\lambda^T : (V \cup \mathcal{E}) \times L \times \mathbb{N} \to val$ *is a partial function that maps a node or an edge, a property label, and a point in time to a value of the property at the specified time.*

In the definition of a temporal graph, V contains unique labels from $\mathbb{N}$ for each node and $\mathcal{E}$ contains unique identifiers for each edge. It is worth noting that neither the definition of $\mathcal{E}$, nor Γ restricts having multiple edges between two nodes (and in later chapters, we will usually have such multiple edges between nodes).

For our convenience, now we introduce a definition to the availability set of edges in a temporal graph and later use it to define quantum networks.

Definition 5.2.4. *Availability set*
Given a temporal graph $\mathcal{G}$ and an edge denoted by $id \in \mathcal{E}$, we define the availability set of id as the ordered set of time windows when the edge is available: $T_{id} = ([t_{init_1}, t_{end_1}], [t_{init_2}, t_{end_2}], ...)$, where:

- $\forall [t_{init}, t_{end}] \in T_{id}\, \forall t \in [t_{init}, t_{end}] : \xi^T(id, t) = True$;
- $\forall t \in \mathbb{N} \exists [t_{init}, t_{end}] \in T_{id} : t \in [t_{init}, t_{end}]$;
- *Given a* $[t_{init_i}, t_{end_i}], [t_{init_{i+1}}, t_{end_{i+1}}] \in T_{id}$, $\forall t_i \in [t_{init_i}, t_{end_i}] \forall t_{i+1} \in [t_{init_{i+1}}, t_{end_{i+1}}] : t_i < t_{i+1}$

Thus the availability set for a certain edge contains time windows when the edge is available.

Due to the dynamic nature of the topology, the existence of a path between any two vertex in a temporal graph is time-dependent. Therefore, a new notion of the path with regards to time will be needed.

Definition 5.2.5. *Time-dependent path*
Let $\mathcal{G}$ be a temporal graph, $s, e \in V$, $t \in \mathbb{N}$ and $path_t^T(s, e) = \{id_1, id_2, ...id_{n-1}\}$ a path between s and e. If $\forall i \in [1..n-1] : \xi^T(id_i, t) = True$, then $path^T(s, e)$ is a time-dependent path at t.

As $path^T(s, e)$ is a path in $\mathcal{G}$ connecting s and e, $\Gamma(id_1) = (s, v_1)$ and $\Gamma(id_n) = (v_n - 1, e)$ are trivially true.

It is worth noting that the definition of a time-dependent path is stricter than of the definition of a time-respecting path present in the literature of temporal networks (e.g. [HS12]). The difference comes from the fact that instead of requiring non-decreasing times for consecutive edges in the path, edges in a time-dependent path need to be available at the same point in time.

There is yet another difference between the two concepts when it comes to the transitivity property [HS12]. This property, does not hold for time-respecting paths, but holds for time-dependent paths:

Theorem 1. *Transitivity of time-dependent paths*
Let $\mathcal{G}$ be a temporal graph and let v_1, v_2, v_3 be vertices $\in V$ and $t \in \mathbb{N}$. If $\exists path_t^T(v_1, v_2) \wedge \exists path_t^T(v_2, v_3)$ then $\exists\, path_t^T(v_1, v_3)$.

Proof. Considering the time-dependent paths as simple paths at time t, the proof follows analogously to proving the transitivity of paths in an undirected graph. □

In our study, we will be examining whether, for a certain source-destination pair, we can find a time, for which there is a time-dependent path.

5.2.3 Defining quantum networks

Having built up a fitting representation of temporal graphs, we can now define quantum networks using these constructs as a temporal graph.

Definition 5.2.6. *Quantum network*
A quantum network is a temporal graph $\mathcal{G} = (V, \mathcal{E}, L, \Gamma, \xi^T, \lambda^T)$ and the underlying physical graph $G_{Ph} = (V, E_{Ph})$ such that:

- $\forall id \in \mathcal{E}\ \forall t \in \mathbb{N}$:

$$\xi(id, t) = \begin{cases} True & if\ \exists\, [t_{init}, t_{end}] \in T_{id} : t_{init} \leq t \leq t_{end} \\ False & Otherwise \end{cases}$$

- $\forall id \in \mathcal{E}\ \forall j \in \mathbb{N} : t_{end_j} + 1 < t_{init_{j+1}}$, *where* $[t_{init_j}, t_{end_j}], [t_{init_{j+1}}, t_{end_{j+1}}] \in T_{id}$

The first point in the definition describes the fact that if the link is available at a certain point in time, then this time belongs to some time window. When the time window ends, the link will become unavailable. The second point incorporates that once a link becomes unavailable, there is a certain latency until it becomes available again.

For brevity we will be using the notation $\mathcal{G} = (V, \mathcal{E}, L, \Gamma, \xi^T, \lambda^T)$ for quantum networks, and we also mean to incorporate an underlying physical graph.

Definition 5.2.7. *Generation time*
Given a quantum network and an edge with $id \in \mathcal{E}$, the generation time of the edge id is $T_{gen} = argmin_i\{\xi(id, t + 1 + i) = True\}$, where $t \in \mathbb{N} : \xi(id, t) = True$, $\xi(id, t + 1) = False$ and $i \in \mathbb{N}$.

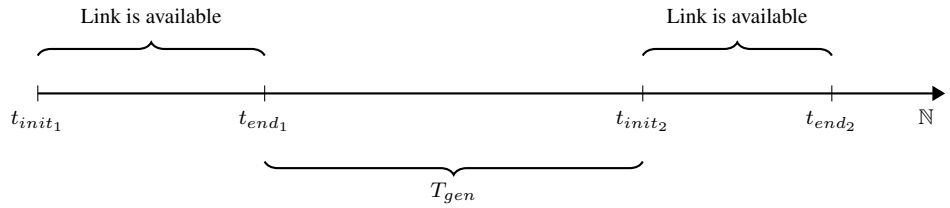


Figure 5.2: Lifecycle of a quantum link. The respective t_{init} and t_{end} with the same index represent the same time windows upper bounded by a T_{th}value (see defintion 5.2.2). T_{gen} denotes the time required for successful link generation.

5.3 Routing in a quantum network

Definition 5.3.1. *Path finding in a quantum network*
Given a quantum network $\mathcal{G} = (V, \mathcal{E}, L, \Gamma, \xi^T, \lambda^T)$*,* s*,* $e \in V$ *and* $t \in \mathbb{N}$ *the problem of path finding is providing a* $t < t'$ *such that* $path^T(s, e) = \{id_1, id_2, ...id_{n-1}\}$ *is a time-dependent path between* s *and* e *at time* t'*.*

We call the ordered four of $(s, e, \#EPR \ \ pairs, \ t)$ a *demand*. Solving the pathfinding problem in a quantum network is one of the main goals of quantum network routing. However, in quantum networks, there are several metrics which would prove to be important for a given time-dependent path.

On one hand, if a time-dependent path considered as a simple path at $t \in \mathbb{N}$ (when it is time-dependent) the fidelity of the end-to-end entanglement that can be created using this path decreases with its distance. In other words, it is worth having small distance paths as time-dependent paths.

On the other hand, we would like nodes not to wait too long after they have requested a link. What this means is that we would like to secure a relative low latency for each request. For this reason, serving demands fast is key, hence we would like to find a time-dependent path as soon as possible after the request arrived. In our analysis, we choose the second metric as our main focus. Therefore, we introduce the concept of fastest time-dependent path.

Definition 5.3.2. *Fastest time-dependent path*
Given a quantum network $\mathcal{G} = (V, \mathcal{E}, L, \Gamma, \xi^T, \lambda^T)$ *and a demand of* $(s, e, \#EPR \ pairs, t)$*a* $path^T(s, e) = \{id_1, id_2, ...id_{n-1}\}$ *that is a time-dependent path between* s *and* e *at time* t' *is called the fastest time-dependent path, if* $t' = min\{t \in \mathbb{N} | \forall j \in [1..n-1] \exists [t_{init}, t_{end}] \in T_{id_j} : t \in [t_{init}, t_{end}]$

5.3.1 Concepts in quantum networks

In the previous section, we have defined quantum networks, now we would like to have a look what each of the concepts from temporal graphs mean in quantum networks.

Time-dependent paths will determine whether an end-to-end entanglement can be created between two nodes in a quantum network. A solution to the path finding problem (such as finding the fastest time-dependent path) will determine the time when the entanglement swap and quantum teleportation operations are going to take place.

The problem of *routing in a quantum network* can be worded as considering the time-dependent paths for a source and destination pair and choosing the one with a minimum cost in the temporal graph. What a minimum cost is for a given time-dependent path depends on what constraints we would like to pose on the time-dependent paths that we are choosing. Several approaches will be used in the later chapters.

5.3.2 Time evolution of the quantum network

Given a quantum network, we have seen that various events affect the connectivity of the network. Through observing and analysing a series of such events, we can conclude how the network evolves. Our goal is to (with some heuristic) predict how it is going to change, and solve the problem of routing in a quantum network accordingly. In this section, we provide an example of the evolution of a quantum network over time.

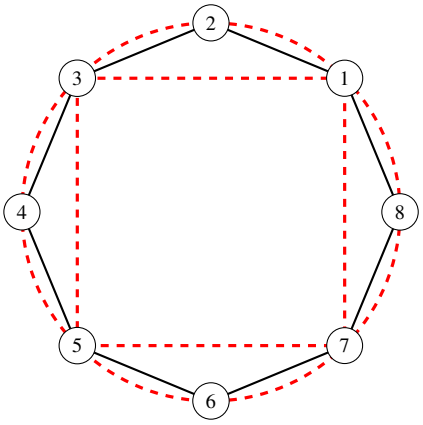
Example

Our focus will mainly be examining quantum networks with virtual graphs that are connected at $T = 0$ (i.e. there is a time-dependent path between any two nodes at $T = 0$). Therefore, we consider topologies that are constructed in ways as described in section 4.6.

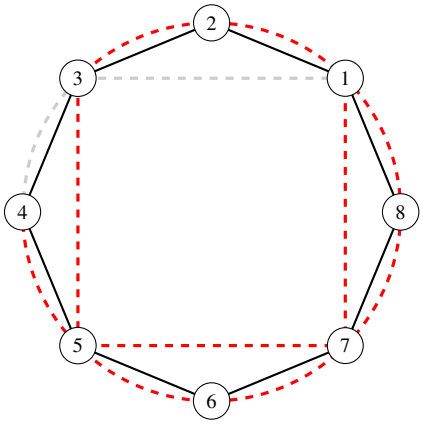
In our example, we assume to have a quantum network of 8 nodes. We look at a time window, where at t_{init} point in time, the availability of edges in the network corresponds to a deterministic virtual graph of $d_{th} = 2$ and $maxd_{th} = 2$ (see figure 5.3f). After this point in time, requests are received.

Upon receipt, a route is determined for each of the requests. We assume that in each case this route is the shortest path in the virtual graph between source and destination. The entangled links along the route are used to create end-to-end entanglement and hence they are consumed (see subfigures 5.3b, 5.3c and 5.3d). The requests are served in the network at t_1, t_2 and t_3 points in time. Eventually, at a certain t_{end} the links decohere, and right at the next point in time, the links are not available (see figure 5.4).

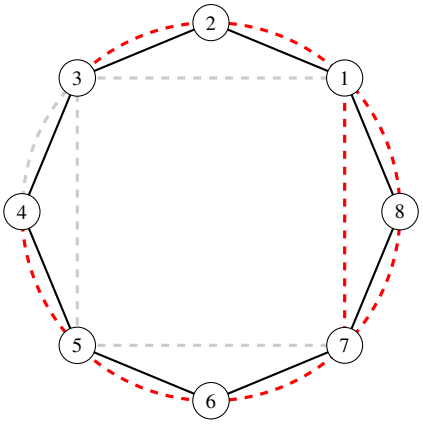
As it can be concluded, after such a time window, a new set of entangled links need to be generated. This allows us to serve the next set of requests coming into the network.



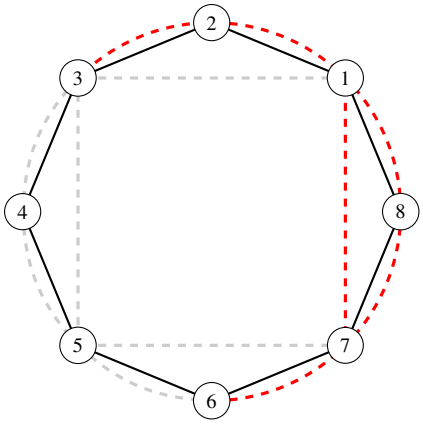
(a) Quantum network wtih pre-shared entangled links at t_{init}.



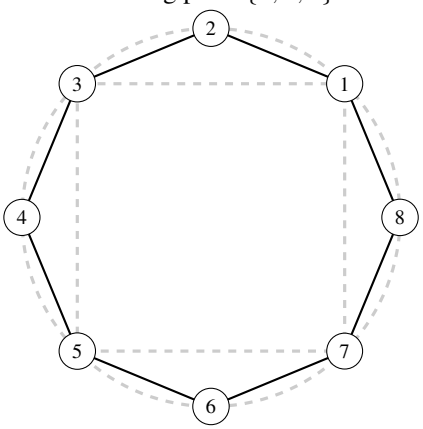
(b) 1. demand from node 4 to node 1 arrives at time t_1, entangled links are consumed along path: $\{4, 3, 1\}$.



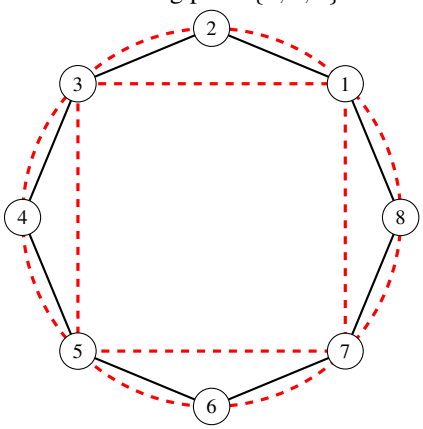
(c) 2. demand from node 7 to node 3 arrives at t_2, entangled links are consumed along path: $\{7, 5, 3\}$.



(d) 3. demand from node 4 to node 6 arrives at t_3, entangled links are consumed along path: $\{4, 5, 6\}$.



(e) Quantum network wtih pre-shared entangled links at t_{end}.



(f) Quantum network wtih pre-shared entangled links at t_{init2}.

Figure 5.3: Time evolution of a quantum network of $|V| = 8$, $d_{th} = 2$ and $maxd_{th} = 2$.

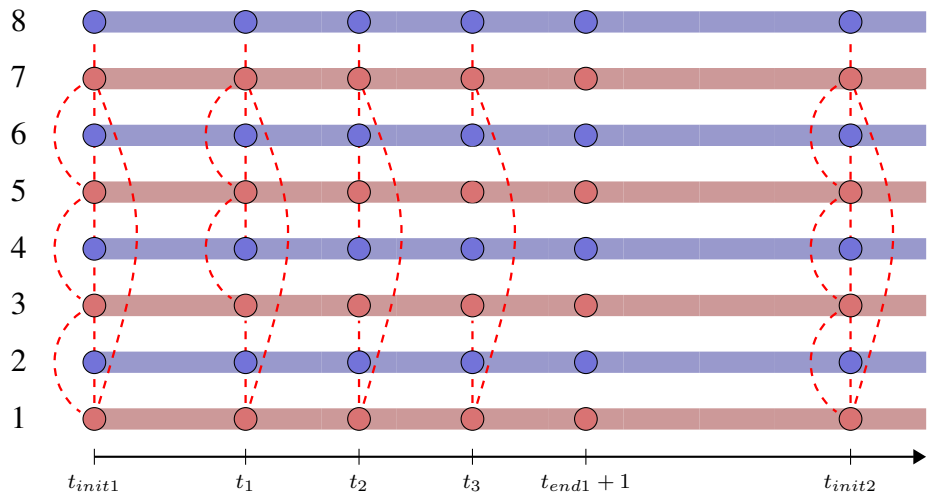


Figure 5.4: Time evolution of a quantum network of $|V| = 8$, $d_{th} = 2$ and $maxd_{th} = 2$ serving demands in a temporal network representation. t_1, t_2 and t_3 are the points in time when a demand is served in the network.

5.4 Challenges of entanglement routing in a quantum network

As previously mentioned, we are interested in the time it takes for a demand to be served in the quantum network. This time is equal to the time it takes to create end-to-end entanglement between source and destination nodes. The created EPR pair is then used to teleport a qubit.

Let us look at the latency for each of these operations (a summary can be found in table 5.1). Considering the first phase of end-to-end entanglement creation, path discovery (section 2.3.1), we have two opportunities: centralised or distributed routing. In this thesis, we focus on the first approach that is a path can be computed from source to destination in a centralised fashion. In this case, the source node determines the path based on its knowledge about the network. With the current capabilities of computers and CPUs, we can assume that the time such a computation would require is negligible.

When it comes to entanglement reservation (section 2.3.2), the source node needs to notify every node along the path to reserve entanglement capabilities. One approach to this is that these nodes will receive a classical message about the demand and send an acknowledgement to the request. The time for this communication scales linearly with the diameter of the classical infrastructure.

In the entanglement distribution phase (section 2.3.3) let l denote the number of entangled pairs to be generated along the reserved path. This number is equal to $d + 1$ where d is the hop distance, the number of "hops" a message does between source and destination using routers. Taking the model used in [CRDW19], the

latency for link generation will depend on several factors and scale exponentially with d, as described in section 2.2.3.

Once all the entangled links have been generated pairwise through the reserved path, entanglement swap operations can commence creating the end-to-end entanglement.

Finally, the latency for quantum teleportation is determined by the classical communication needed for the operation between source and destination.

As can be observed, the latency for the creation of end-to-end entanglement is significantly determined by the number of entangled links that need to be generated. If almost all entangled links are available, then the time entanglement generation takes is negligible. According to our model, in this case, the latency for end-to-end entanglement creation is mostly determined by classical communication and the time certain operations like entanglement swap, quantum teleportation take.

Once, however, a significant number of entangled links are missing along the reserved path, the latency is going to be very high. Thus, optimising path discovery to decrease the number of entangled links to be generated proves to be key to keep the latency for demands low. In subsequent sections and chapters, we would like to delve into details on what techniques can be used for this purpose.

Procedure	Time-consuming operation	Latency
Path discovery	Computing the path	$O(1)$
Path reservation	Sending messages to the nodes along the path	$O(diam_{G_{Cl}})$
Entanglement generation	Successful elementary link generation along the path	$O((\frac{1}{P_0})^l)$
Entanglement swap	Sending messages to perform the swap	$O(diam_{G_{Cl}})$
Quantum teleportation	Sending a message to perform the measurement	$O(diam_{G_{Cl}})$

Table 5.1: Latency for end-to-end entanglement creation (section 2.2.3) and quantum teleportation (section 2.1.3). We denote the diameter of the classical communication infrastructure with $diam_{G_{Cl}}$. P_0 is the probability of successful elementary link creation whereas l is the number of elementary links that need to be generated. Denoting d as the hop distance, $l = d + 1$.

5.5 Summary

In this chapter we have formulated a new model for entanglement routing in a quantum network using temporal graphs. First, we have identified that the notion of time and the change of the network over time is still to be included in our models. Then, we have used the temporal graph concepts to define quantum networks and routing in a quantum network. At the end of the chapter we provided an example for the evolution of the network.

Chapter 6

Information propagation in quantum networks

In the previous chapter we have discussed a dynamic (non-static) model for quantum networks. In this chapter we use this formalism to present empirical results coming from simulations. In order to do this, we first introduce the concept of knowledge about the network topology. Then, we describe how it plays a key role in quantum network routing.

In our work we make the assumption that nodes know about the network topology for the underlying physical graph. They can also make use of the classical communication infrastructure between them.

We have taken three different approaches based on the level of knowledge a node has about the network topology. This knowledge is important, as nodes perform Path Discovery 2.3.1 based on this knowledge.

In this chapter we introduce the following approaches:

1. initial knowledge;
2. local knowledge and
3. global knowledge.

First, by taking the initial knowledge approach, we assume that each node has a static knowledge about the topology throughout the lifetime of the network. With the local knowledge approach, information is propagated about the change in the topology within a certain radius. By increasing the level of information propagation in the network to its maximum, we achieve global knowledge. As later explained, global knowledge means that the level of information propagation guarantees that each node is always up to date with the current state of the network.

In this chapter, centralised routing is used to compare preliminary intuitions with observations based on empirical results from simulations.

6.1 The method of observation

Our aim is to observe how the latency is for a demand in the quantum network. For a given demand $(s, e, \#EPR\ \ pairs, t)$, latency is the time difference between t and point in time when the end-to-end entanglement has been created between s and e.

Our simulations consist of executing the phases of end-to-end entanglement creation for several demands (see section 2.3 for a description on the phases) and keeping track of the average latency.

Path Discovery takes place in a centralised fashion. The node requesting entangled links performs the Shortest-Path algorithm (algorithm 4) to find a path towards the destination based on the local knowledge it has about the network. This means that it uses a locally stored graph representing the network as an input to the algorithm. It then, sends out classical communication messages to execute the Entanglement Reservation phase (see section 2.3.2).

As hinted on in section 2.3.2, the Entanglement Reservation phase takes place sequentially for entire paths, and so our simulations are devised accordingly. This implies the following. Given two demands, (s_1, e_1, epr_1, t_1) and (s_2, e_2, epr_2, t_2), let us denote the paths returned by Path Discovery as:

$$path_1^T(s_1, e_1) = \{id_1, id_2, ...id_{n-1}\},$$
$$path_2^T(s_2, e_2) = \{id_1, id_2, ...id_{n-1}\}$$

respectively.

Let us then denote:

$$JointSet = (path_1^T(s_1, e_1) \cap path_2^T(s_2, e_2)).$$

In this case, if $t_1 < t_2$, then $\forall id \in JointSet$ will be utilised by the first demand. The second demand will then either utilise parallel links between the all (v_1, v_2) $(v_1, v_2) = \Gamma(id)_{id \in JointSet}$ (see 5.2.3 for definition) pairs of nodes or reserve capacities for generating entangled links between these nodes.

Once a path has been reserved for a demand, Entanglement Distribution (see section 2.3.3) takes place and the entangled links used for end-to-end entanglement creation are consumed. If there are not enough entangled links available along the discovered path, a generation process takes place.

We use the discrete time model described in section 4.3. In section 5.4, we have established the fact that the procedure that greatly influences the latency of end-to-end entanglement creation is entanglement generation. Consequently, in our simulations we are computing the time entanglement generation takes as latency while neglecting the time other procedures would take. In our work we do not impose constraints on the minimum fidelity needed along the path. Should we want to extend these observation methods by also incorporating the requirement for fidelities, we would need to include it in our demand specification. As far as the topology goes, we can simply assign fidelities as a property to each edge and posing an extra constraint on the fidelity of the edges that are used in a path.

6.2 Knowledge in quantum network routing

As explained in section 5.4, the time of end-to-end entanglement creation is highly dependent on the number of entangled links that need to be generated. This number is, however, determined by the path along which the entangled links are used to create the end-to-end entanglement. Looking at the phases of end-to-end entanglement generation phases as described in section 2.3, the crucial phase for this is the first one, namely the Path Discovery phase, as this is when the path is determined.

The Path Discovery phase can be reduced to solving the shortest path problem. This is so because the purpose of the Path Discovery phase would be to find the shortest path between a source and destination along which we can reserve the entangled links adhering to the requirements set by the demand.

It is, therefore, needed to introduce what we mean by a certain node using its knowledge about the network to perform Path Discovery.

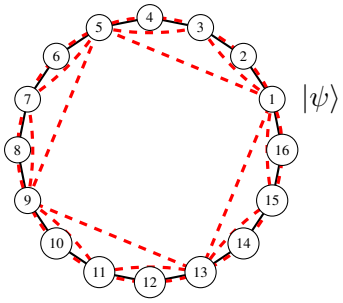
Definition 6.2.1. *Knowledge about the network*
Given a quantum network $\mathcal{G} = (V, \mathcal{E}, L, \Gamma, \xi^T, \lambda^T)$*, let us denote the set of edges that are available at a* $t \in \mathbb{N}$ *point in time as* $\mathcal{E}_t$*, that is:* $\mathcal{E}_t = \{e | e \in \mathcal{E} \wedge \xi(e,t) = True\}$*. A node* $v \in V$ *has knowledge about the quantum network at time* t *if it locally stores a* $G_v = (V, E_v)$ *for which* $\exists t \in \mathbb{N} : E_v \subset \mathcal{E}_t$*.*

6.2.1 Why is knowledge important?

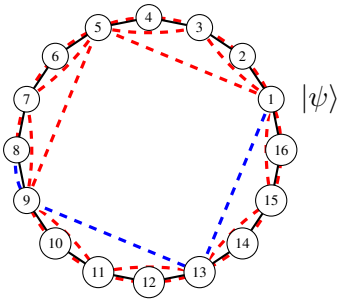
Let us take an example of what role knowledge plays once a quantum network is used for communication. Assume that we have a quantum network with $N = 16$ and pre-shared entangled links constructed as shown in figure 6.1 (similar to the ones described in section 4.6). Let us take the case when there is a demand from node 1 to node 8. A potential path to create end-to-end entanglement could be $path = 1, 13, 9, 8$. As it is demonstrated in subfigures 6.1a- 6.1e, the entangled links along this path are consumed.

Let us now assume, that node 2 wants to teleport a qubit to node 12. It uses the topology shown on subfigure 6.1a to discover the path. Hence, it determines the $path = \{2, 1, 13, 12\}$ as a path. However, the entangled link between nodes 1 and 13 is consumed and needs to be regenerated. The real state of the network is represented on 6.1e. Should node 2 have had the knowledge about the real state of the quantum network, it could have discovered another path such as $path = \{2, 1, 15, 13, 12\}$. Instead, the time for the demand to be served is increased by the creation of the long link between nodes 1 and 13.

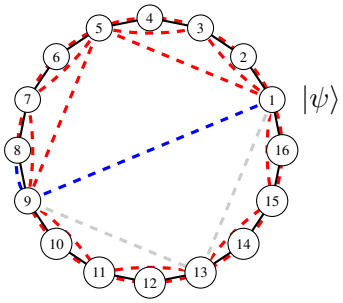
This example hints on the fact that the knowledge stored locally by each node plays a key role in Path Discovery. In order to find a good path available, information about the change in the network needs to be propagated to nodes. In later sections, we will examine how much the knowledge of the nodes affect the time it takes to serve demands.



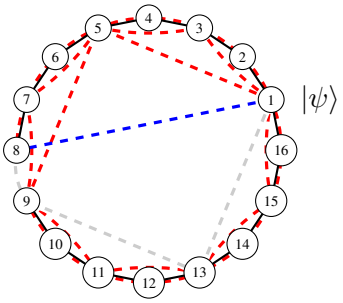
(a) Quantum network with pre-shared links.



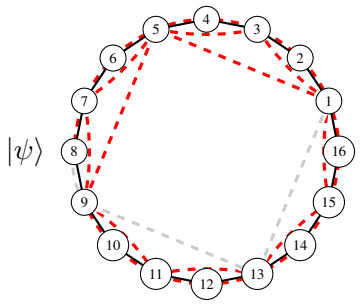
(b) Entanglement swap at node 13.



(c) Entanglement swap at node 9.



(d) End-to-end entanglement between 1 and 8.



(e) Quantum teleportation using the end-to-end entangled link.

Figure 6.1: The creation of end-to-end entanglement between nodes 1 and 8. Once created, the end-to-end entanglement is used for the quantum teleportation of the $|\psi\rangle$ quantum state from node 1 to node 8.

6.3 Simulation setup

In the following sections, we will present results for Python simulations that were carried out based on the level of knowledge each of the nodes had. Before we delve into the description of the approaches used, in this section, we discuss the simulation setup. This was standard for each of the scenarios described later, or else, the change of parameters will be mentioned.

The topology chosen as a simulation model is a one-dimensional lattice (ring topology) of 32 nodes. A physical graph is created by connecting nodes along the ring (see figure 4.2 in section 4.5). Virtual graphs are created based on the rules discussed in sections 4.6 and 4.6.2. We carry out simulations in the on-demand model as well as the continuous model. Simulations based on the continuous model start from a virtual graph with pre-shared entangled links.

Demands are sampled uniformly at random, which means that an evenly distributed traffic is simulated. The number of demands is equal to 50 in most of our simulations (else marked otherwise). In our models, we assume that there are no limits to generate capacities and that generation attempts to start taking place immediately after the reservation attempt. The parameters are used according to the values presented in [CRDW19], so the threshold value for time windows is set as $T_{th} = 1,000$ time units. If the generation time of entangled links along the discovered path would take more than this value, then the generation process does not take place. Instead, we assume that the demand will be served in the next time window. The value of T_{th} is regarded as generation time and so as latency.

First, we fix the number of demands and sample them. Then we use Monte Carlo simulations with at least 1000 rounds to determine what the average latency for these demands could be. In the process, we also determine the following properties:

- average number of links used for each demand;
- number of links remaining in the quantum network (multiple links are possible between the same pair of nodes);
- average number of links remaining (counting multiple links between the same pair of nodes once).

If the quantum network is constructed based on a randomly generated network topology, we average 10 separate simulation results.

6.4 Initial knowledge

6.4.1 Approach taken

We start from a quantum network $\mathcal{G} = (V, \mathcal{E}, L, \Gamma, \xi^T, \lambda^T)$. With the initial knowledge approach, we assume that each node $v \in V$ has knowledge $G_v = (V, E_v)$ about the network. According to definition 6.5.2 this means, that $\exists t_{init} \in \mathbb{N}$:

$E_v = \mathcal{E}_{t_{init}}$. We further make the assumption, that this t_{init} is the starting point of a time window with a specific T_{th} threshold value (see definition 5.2.2). We then carry out simulations as explained prior.

6.4.2 Results and discussion

Having previously introduced the simulation setting and the approach that is going to be taken, now we discuss the intuition behind how the initial knowledge approach would perform. We apply this approach to different topologies introduced earlier in chapter 4.6. Our intuition based on works like [Kle00][CRDW19] is that the algorithm performs better in topologies where longer links are available. That is, if we keep the same number of edges, but increase the value of the distance threshold, then we effectively decrease the diameter of the network. A smaller diameter implies a decrease in average latency.

Deterministic virtual graphs

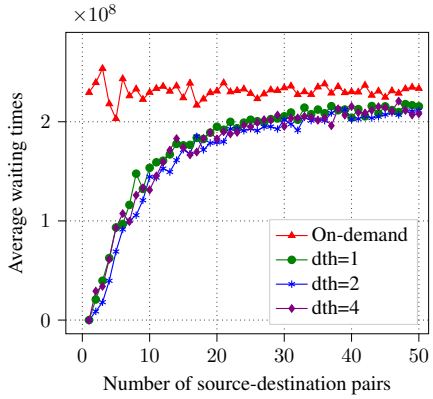
The first simulation result for the initial knowledge approach was carried out in a quantum network of a deterministic virtual graph with 32 nodes, 56 links and the parameter $maxd_{th} = 4$ (see definition of distance threshold 4.6.3).

Subfigure 6.2a shows the values for the average latency for each number of demands. What we can observe is that the previously described intuition does not hold. First of all, the on-demand model performs as expected, and sets a nearly constant average latency regardless of the number of demands. However, there seems to be no correlation between increasing the distance threshold and a decreased average latency. Proof for that is the fact that $d_{th} = 2$ topology outperforms $d_{th} = 4$ quite clearly up until 30 demands. After that, each of the topologies with pre-shared links starts to tend to the on-demand model. What could be the reason for such behaviour? If the topology contains shorter paths, "shortcuts" for the demand, how could such a construction of the network still result in higher average latency?

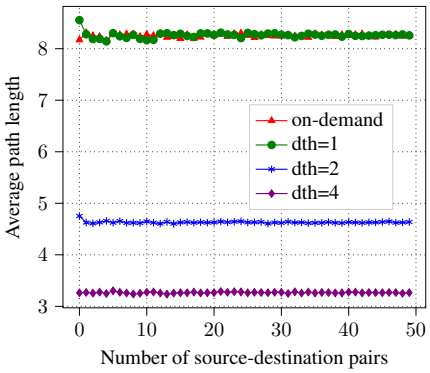
The intuitive answer is that these "shortcuts" are available once, but traffic is routed towards them based on initial knowledge after their consumption as well. Let us consider figure 6.2. From the second demand using the same long link, they need to be regenerated, which introduces the exponential scaling with physical distance mentioned in section 5.4. Examining subfigure 6.2c and the curves for d_{th}= 2 and d_{th}= 4, we can observe that their decrease rate is the same at around when the number of demands reaches 10. This seems to indicate the fact that in the d_{th}= 2 topology as well, the longer, length 2 links are utilised for prior demands. Once there are more than 10 however, the regeneration of the long links need to take place.

Potential new approaches in the implementation of quantum networks have the prospect of changing the correlations between our parameters. Therefore, we have further carried out a simulation where we assume, that given a chain of routers, the

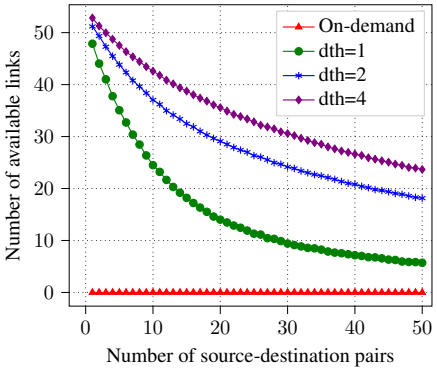
time for generating entanglement in a quantum router chain scales polynomially with the hop distance (see figure 6.3). This behaviour is contrary to the exponential scaling used in our other simulation rounds.



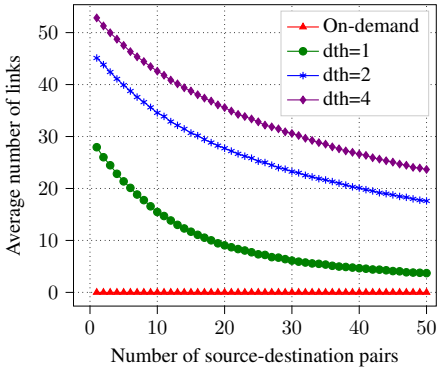
(a) Average latency.



(b) Average path length.



(c) Average number of links remaining.



(d) Average number of links (counting multiple links between the same pair of nodes once).

Figure 6.2: Initial knowledge approach in a quantum network of a deterministic virtual graph of $|V| = 32$, $|\mathcal{E}| = 56$ and $maxd_{th} = 4$.

The example of polynomial scaling is fairly similar to the exponential result previously presented when it comes to comparing each of the topologies. It can be observed that advantage is not gained in the $d_{th} = 4$ case in this setting either. The difference between $d_{th} = 2$ and the other two topologies of the continuous model is proportionally larger, as it was the case in figure 6.2.

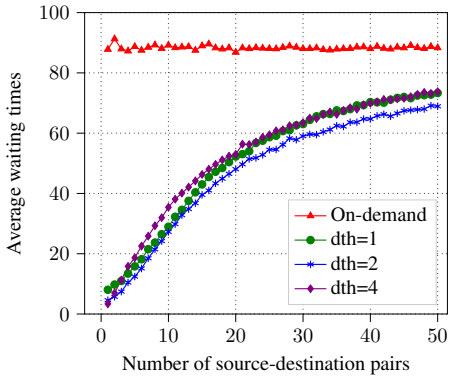


Figure 6.3: Initial knowledge approach in a quantum network of a deterministic virtual graph of $|V| = 32$, $|\mathcal{E}| = 56$ and $maxd_{th} = 4$. In this case, the time for generating entanglement in a quantum router chain is a polynomial function of the hop distance between source and destination.

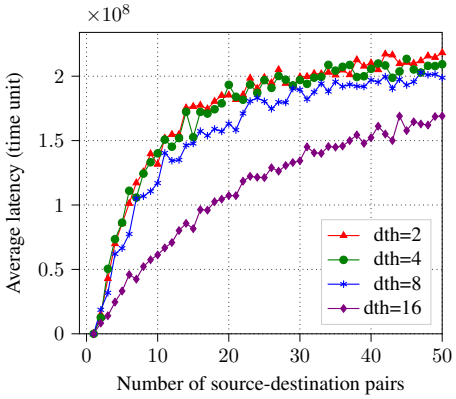
Random virtual graphs

After simulations in topologies of deterministic virtual graphs, we proceed to quantum networks with random virtual graphs. The construction of these topologies is detailed in section 4.6.2.

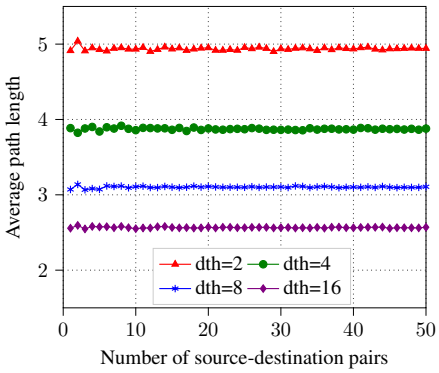
The first example clearly proves our basic intuition is figure 6.4. Constructed based on the rules described in section 4.6.2, it shows quite a considerable advantage, when increasing the distance threshold. This property could be because the long links in the topology are generated at random following a power-law distribution. Consequently, there are no main clusters formed, towards which traffic would be routed to again and again. For this reason, it is more probable that a link that has not been used before is utilised.

At last, we transit to a random virtual graph where $k = 4$ (see figure 6.5. What we observe is that apart from $d_{th} = 2$, an increased value of the distance threshold results in a decreased average latency. This proves our preliminary intuition. Why are then the values for average latency low up until 15 demands in the $d_{th} = 2$ topology?

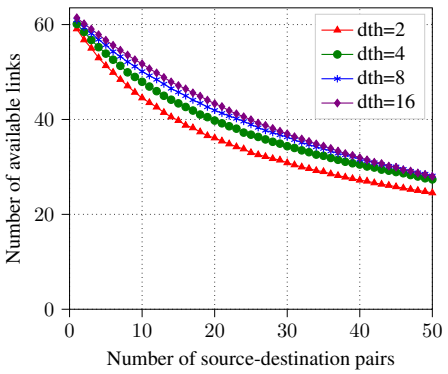
An intuitive answer to that question would be that for demands sampled uniformly at random, there are available links to be used. The way this is possible, is that in this topology $|V| = 32$ and $k = 4$, so there are 120 links of length 2. These links provide very good connectivity for the topology. However, at least one more link needs to be used on average for each demand as compared to topologies with higher distance thresholds (see subfigure 6.5b). Based on this fact, the links are consumed way faster (see figure 6.5c, and this advantage is lost as the number of demands further increases. An example that can be observed at 40 demands, when the approach starts to perform better in the $d_{th} = 16$ topology.



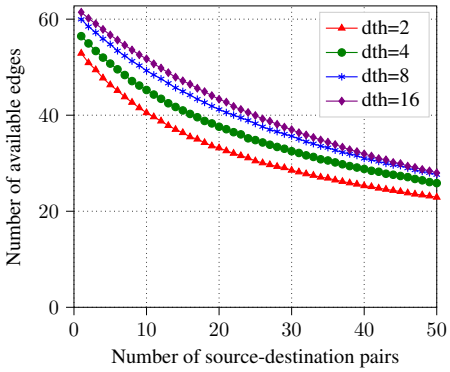
(a) Average latency.



(b) Average path length.

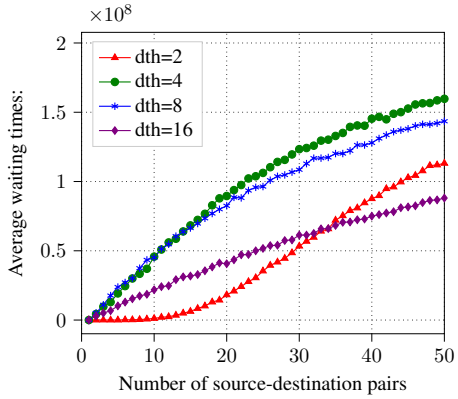


(c) Average number of links remaining.

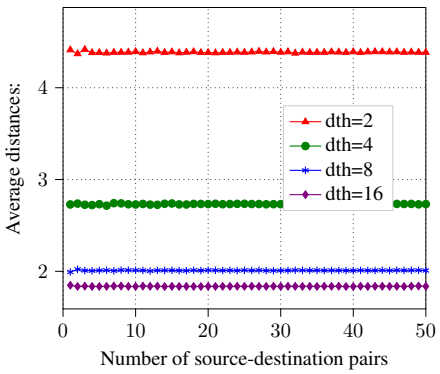


(d) Average number of links (counting multiple links between the same pair of nodes once).

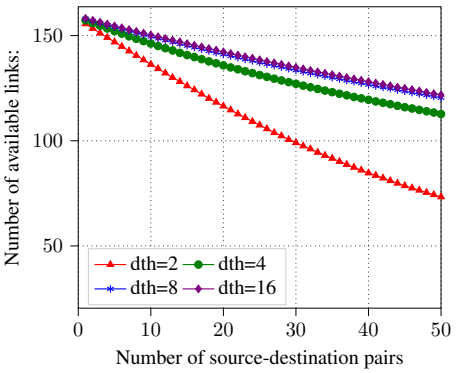
Figure 6.4: Initial knowledge approach in a quantum network with 32 nodes and 64 links. Each node chose one ($k = 1$) long virtual link randomly within the distance threshold of 16.



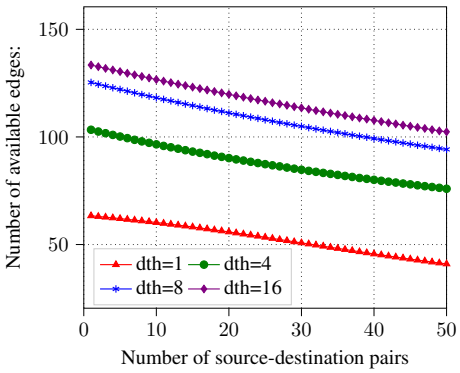
(a) Average latency.



(b) Average distance.



(c) Average number of links remaining.



(d) Average number of links (counting multiple links between the same pair of nodes once).

Figure 6.5: Initial knowledge approach in a quantum network with 32 nodes and 120 links. Each node chose one ($k = 4$) long virtual link randomly within the distance threshold of 16.

6.5 Local knowledge

6.5.1 Approach taken

Having seen the rather static behaviour of the initial knowledge, when it comes to knowledge about the network, we now look at an approach that includes updating the knowledge stored. With the local knowledge approach, we assume an existing quantum network $\mathcal{G} = (V, \mathcal{E}, L, \Gamma, \xi^T, \lambda^T)$ and that each node $v \in V$ has knowledge $G_{init} = (V, E_{init})$ about the network. As an addition, each time Path Discovery takes place, the information about the change in topology is propagated to certain nodes. The nodes to which information is propagated is determined by a pre-defined radius.

Definition 6.5.1. *Information propagation*
Given a quantum network $\mathcal{G} = (V, \mathcal{E}, L, \Gamma, \xi^T, \lambda^T)$*, a demand* $(s, e, \#EPR\ pairs, t)$*and a path* $path(s, e)$ *to serve the demand, information propagation is sending classical messages to certain nodes of the network about* $path(s, e)$*.*

Assume that a node had a locally stored $G_v = (V, E_v)$ knowledge about the network at time t. Once the information has been propagated to the node, its locally stored knowledge will be G updated with the consumed virtual links along $path(s, e)$. Information propagation takes place after the Path Reservation. As mentioned previously, in our simulations we neglect the time of the classical communication. Consequently, upon Path Reservation, nodes receiving classical messages update their local knowledge with negligible delay.

Definition 6.5.2. *Propagation radius*
Given a quantum network $\mathcal{G} = (V, \mathcal{E}, L, \Gamma, \xi^T, \lambda^T)$*, a demand* (s, e, t) *and a path* $path^T(s, e)$*, the propagation radius is* $r \in \mathbb{N}$ *such that information is propagated to all nodes* v *for which* $dist_{G_{Ph}}(u, v) \leq r$ *such that* $u \in path^T(s, e)$*.*

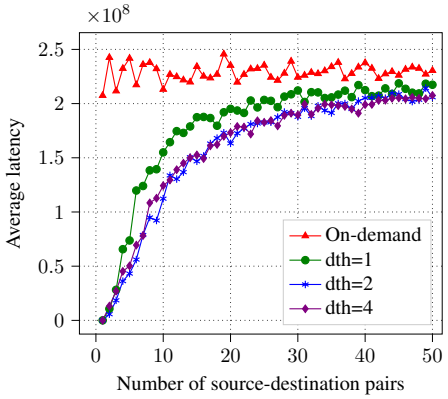
In this chapter, whenever we apply the local knowledge approach, we also specify the propagation radius.

6.5.2 Results and discussion

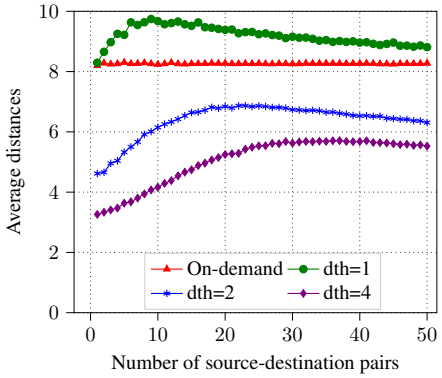
We now consider how the proposed approach performs. Simulating the local knowledge approach was first carried out in a quantum network of a deterministic virtual graph with 32 nodes, 56 links and the maximum distance threshold of 4 (see figure 6.6). We set the propagation radius to $r = 0$. This implies that only the nodes along the discovered path update their local knowledge about the quantum network.

First, let us compare the initial knowledge with the local knowledge approach. The graph showing the average latency for local knowledge differs from the initial knowledge case presented in figure 6.2. What can be observed is that the local knowledge approach performs better in more connected topologies. The difference in average latency between the $d_{th} = 1$ and $d_{th} = 2$ topologies is more significant than it was for the initial knowledge approach. With $d_{th} = 1$ the approach performs worse than the initial knowledge. A similar behaviour can be observedin the $d_{th} = 2$ case. For 10 demands, there are around $3 * 10^7$ time units of difference between the two approaches. One reason for this could be that after a demand has been served, nodes within the propagation radius update their locally stored state about the network. As the distance threshold is equal to one, the virtual graph is a ring with multiple edges between certain physical neighbours. Therefore, once knowledge has been propagated, the traffic may be routed in the opposite direction along the ring. This would ensure the utilisation of the remaining available links, yielding a more advantageous route for certain demands. However, as more demands are served, links along such routes become unavailable. Taking longer paths results in an increased latency when generating unavailable links. This fact

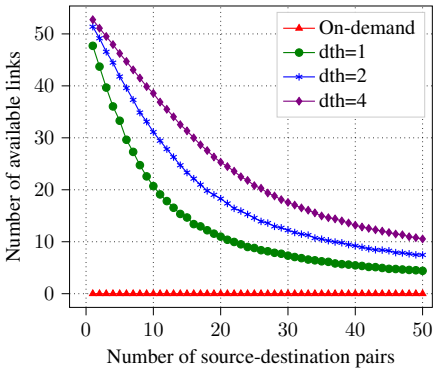
serves as an explanation for the increased overall average latency. Subfigure 6.6b indicates that the average path length for $d_{th} = 1$ exceeds the on-demand average. Considering topologies with a different distance threshold, this approach performs slightly better in the virtual graph with $d_{th} = 4$.



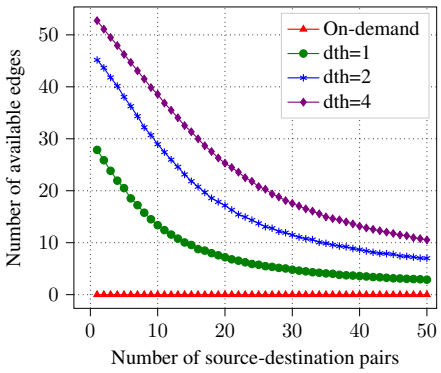
(a) Average latency.



(b) Average path length.



(c) Average number of links remaining.

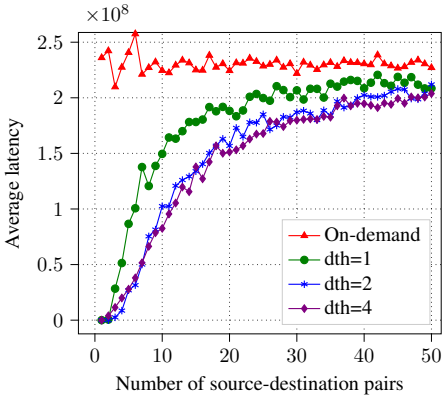


(d) Average number of links (counting multiple links between the same pair of nodes once).

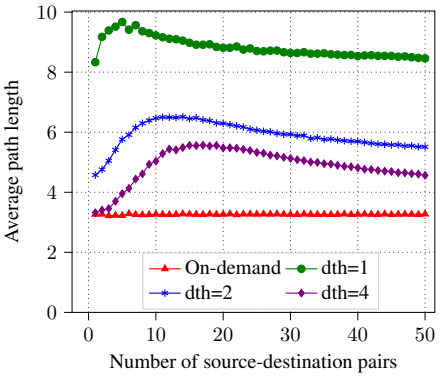
Figure 6.6: Local knowledge approach of $r = 0$ in a quantum network of a deterministic virtual graph of $|V| = 32$, $|\mathcal{E}| = 56$ and $maxd_{th} = 4$.

Figure 6.7 summarizes results for information propagation with radius 5. This type of information propagation seems to be the most beneficial in the $d_{th} = 4$ case. The curve of the average latency for this topology reaches the curve for $d_{th} = 2$ after around 10 demands. For later demands, the average latency is lower in the $d_{th} = 4$ case. The reason for this is that once the long links are utilised, the knowledge about their consumption is propagated to the nodes in the network. The effect of the increased propagation radius can be also observed by comparing subfigures 6.6b and 6.7b. For each of the topologies, the curves of average path

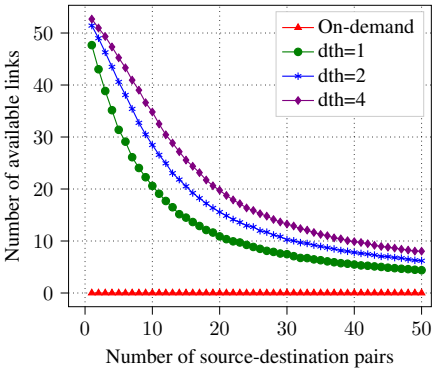
length reach their peak after a fewer number of demands, when the propagation radius is equal to 5. For consequent demands this average tends to the level observed in 6.2b. This is an indicator that demands are not routed towards longer paths once information has been propagated about the change in topology.



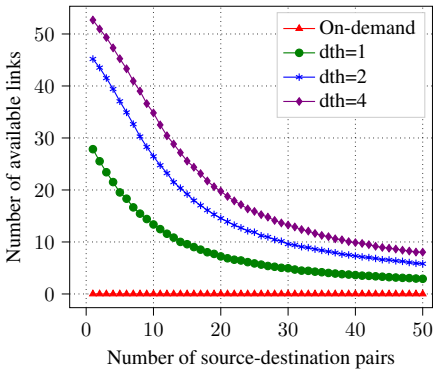
(a) Average latency.



(b) Average path length.



(c) Average number of links remaining.



(d) Average number of links (counting multiple links between the same pair of nodes once).

Figure 6.7: Local knowledge approach of $r = 5$ in a quantum network of a deterministic virtual graph of $|V| = 32$, $|\mathcal{E}| = 56$ and $maxd_{th} = 4$.

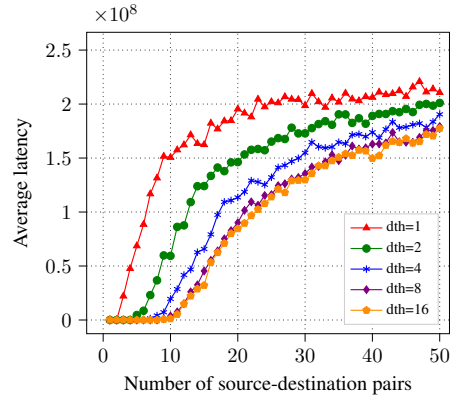
6.6 Global knowledge approach

6.6.1 Approach taken

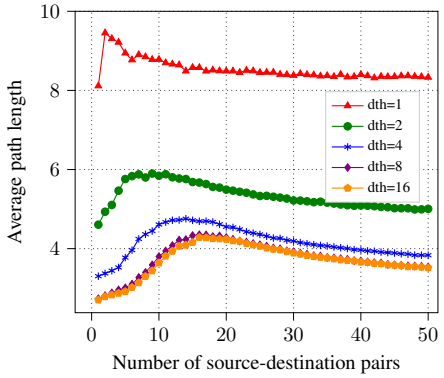
Once again, we start from an existing quantum network $\mathcal{G} = (V, \mathcal{E}, L, \Gamma, \xi^T, \lambda^T)$ and we also assume that each node $v \in V$ has knowledge $G_v = (V, E_v)$ about the network. With the global knowledge approach, the main assumption we make is that information propagation happens by flooding to every node. That is, every

node is informed about the change in topology. This means that following the notation of a node having G_v knowledge about the network at time t, $\forall t \in \mathbb{N}$, $G_v = \mathcal{E}_t$ at all times.

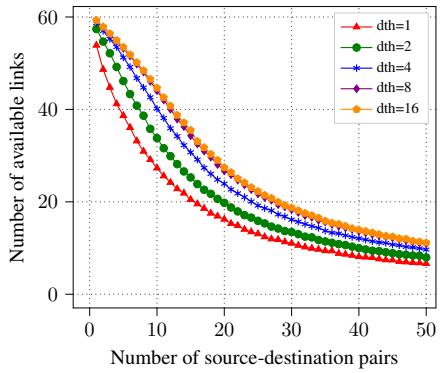
6.6.2 Results and discussion



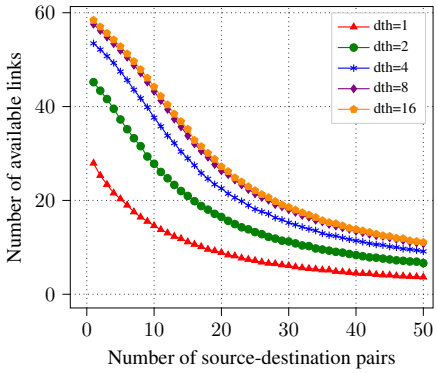
(a) Average latency.



(b) Average path length.



(c) Average number of links remaining.

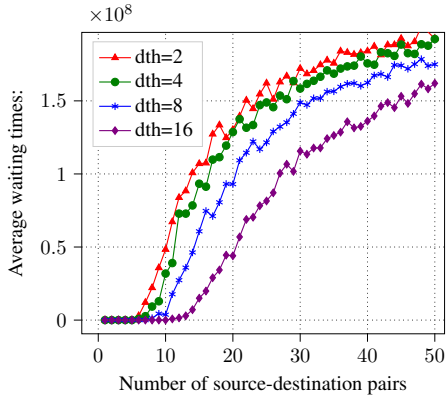


(d) Average number of links (counting multiple links between the same pair of nodes once).

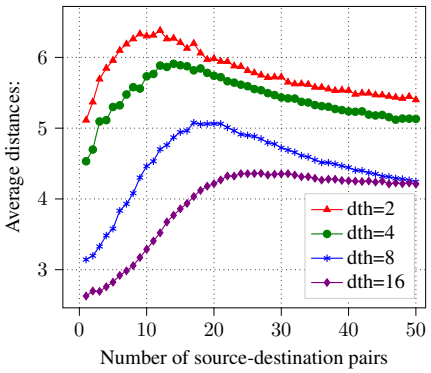
Figure 6.8: Global knowledge approach in a quantum network of a deterministic virtual graph of $|V| = 32$, $|\mathcal{E}| = 62$ and $maxd_{th} = 16$.

The global knowledge approach was first carried out in a quantum network of a deterministic virtual graph of $|V| = 32$, $|\mathcal{E}| = 62$ and $maxd_{th} = 16$ and in a random virtual graph with $k = 1$, $|V| = 32$ and $maxd_{th} = 16$.

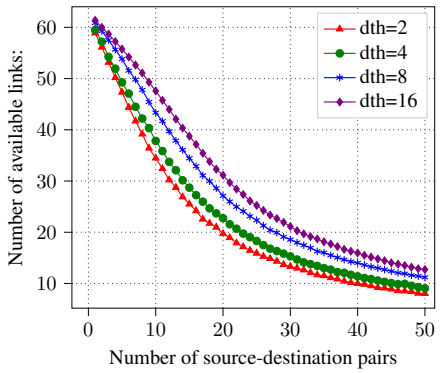
What can be observed from the measurement results (figures 6.8 and 6.9) for the global knowledge approach, is that the preliminary intuition influenced by [Kle00] holds once information is propagated to all nodes. Decreasing the diameter of the topology results in lower average latency for several demands. This means that a topology with a bigger distance threshold gives an advantage. Comparing these



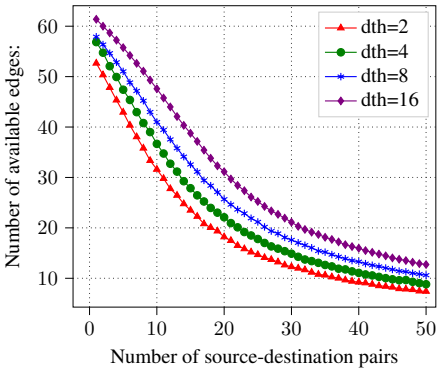
(a) Average latency.



(b) Average path length.



(c) Average number of links remaining.

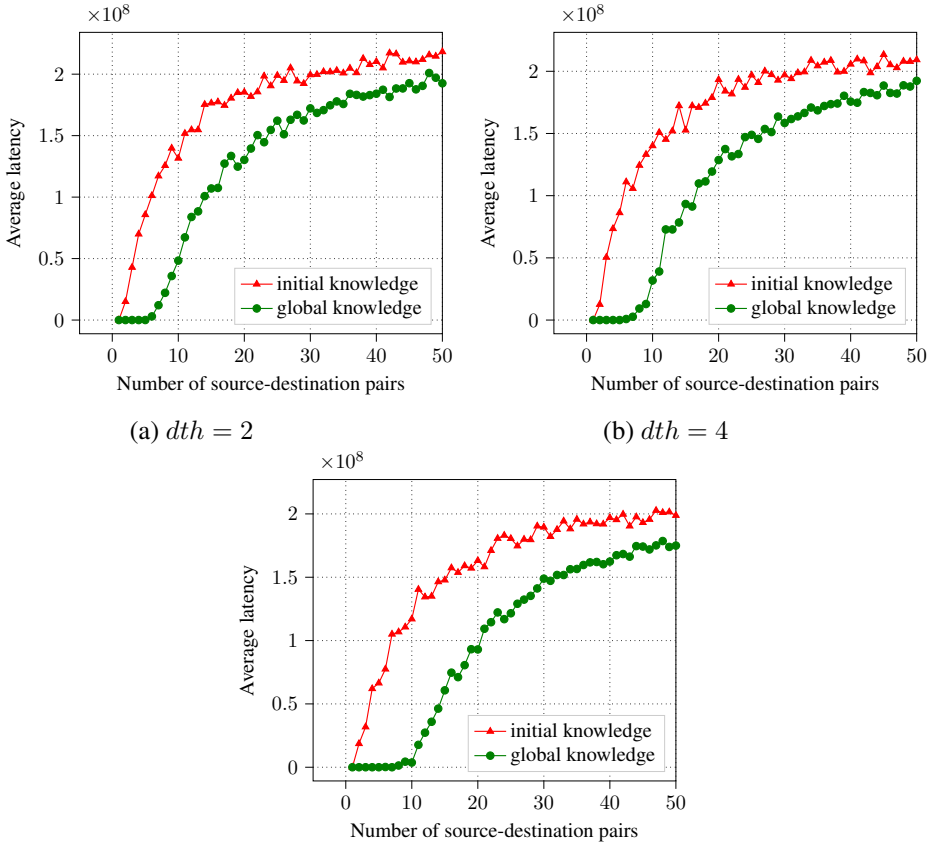


(d) Average number of links (counting multiple links between the same pair of nodes once).

Figure 6.9: Global knowledge approach in a quantum network of a random virtual graph with $k = 1$, $|V| = 32$ and $maxd_{th} = 16$.

results with figures 6.6 and 6.7, this advantage is clearly distinguishable between the different values of distance threshold.

The length of paths is determined by the information propagation. In the global knowledge case, alternative routes with bigger path length are used. Subfigure 6.8b demonstrates how the path lengths increase with the number of demands. The peaks of each curve indicate when links in these alternative routes are consumed. After this point link generation commences along the path with minimum distance in the underlying physical graph. As can be observed, the peak of each curve is determined by the distance threshold.



(a) $dth = 2$ (b) $dth = 4$

(c) $dth = 8$

Figure 6.10: Comparison of the initial knowledge and the global knowledge approach in a quantum network with a random graph of $|V| = 32$, $k = 1$ as virtual graph.

6.7 Summary of results

In this chapter, we looked at the concept of a node having knowledge about the quantum network. We have established the conclusion that knowledge plays an important role in routing in a quantum network (in figure 6.10, this fact is demonstrated). Later, we introduced the concept of information propagation, a technique with which the knowledge of nodes can be updated.

We have looked at simulation results for three knowledge-based approaches:

- initial knowledge;
- local knowledge and
- global knowledge.

We observed that topologies with a low diameter can utilise their advantage and yield low average latency for demands. Information propagation about the change in topology plays a key role to utilise this advantage. Therefore, in this chapter we have examined different levels of information propagation. If information is not propagated to the nodes of the network and they have only an initial knowledge, our results showed no advantage for low diameter topologies. Once information is propagated to the nodes in the network, topologies with a higher distance threshold have a clear advantage. The magnitude of the advantage gained depends on the size of the radius, within which information is propagated.

Chapter 7

Link prediction in quantum networks

7.1 Introduction

In the previous chapter, we have looked at how the knowledge of a certain node in a quantum network affects the average latency for demands. We have also concluded that given a demand $(s, e, \#EPR\ pairs, t)$, there is an inverse correlation between the level of knowledge of node s performing Path Discovery and the average latency for the demand $(s, e, \#EPR\ pairs, t)$.

In this chapter, we present a technique called link prediction with which low average latency can be achieved with non-global (e.g. initial) knowledge without introducing additional latency or message complexity. First, we present the motivation for introducing this technique. Then we look at how a measure related to complex network analysis can be used for link prediction. At the end of the chapter, we present simulation results through comparison with previous results from the previous chapter.

7.2 Motivation for link prediction

In chapter 6 we have seen that by increasing the level of knowledge for each node (see sections 6.4 -6.6), the average latency can be decreased. Following this intuition, once each node has global knowledge about the current state of the network, they can perform Path Discovery and discover the best route available (best in terms of latency for the demand).

How realistic is it, however, that nodes have global knowledge about the network at all times? Achieving global knowledge would entail sending classical messages to all the nodes about the change in topology (broadcasting/flooding). Two factors to be considered in such a case: the introduced time and message complexity. Given a quantum network, in our model, we have assumed that one-time step is equal to the maximum classical communication time between any two

nodes in the physical graph (same as in [CRDW19]). Therefore, the introduced additional latency for the information propagation would be equal to the diameter of the physical graph. Regarding the message complexity, let us denote the classical communication infrastructure by $G_{Cl} = (V_{Cl}, E_{Cl})$ average distance of the underlying physical graph as $\langle d \rangle$. Given a demand $(s, e, \#EPR\ \ pairs, t)$, once node s has performed Path Discovery, it will start a flooding procedure making use of the classical communication infrastructure to propagate information about the change of topology in the quantum network. In this procedure, once a node receives a message, it sends it forward to all neighbours it has, except for the node it received the message from. This means that the overall number of messages sent during this algorithm is $2 * |E_{Cl}| - |V_{Cl}| + 1$. Considering the numbers presented in [Bar16] for the classical Internet, $|V_{Cl}| = 192,244$ and $|E_{Cl}| = 609,066$ resulting in hundreds of thousands of messages sent for a single demand in the quantum network. Altogether, this means that achieving global knowledge in a real-world network is infeasible.

What happens then, if the nodes do not have global knowledge about the state of the network? We have seen in chapter 6.10, that in such cases, the average latency after a certain number of demands is significantly greater than in the global knowledge case. This comes from the fact that given a demand $(s, e, \#EPR\ \ pairs, t)$, if s has initial knowledge about the network, then it might discover a path with more unavailable links than it would have discovered with global knowledge.

The problem of a significantly increased latency described in subsection 6.4.2 comes from the fact that in most connected networks, a given link is part of several shortest paths. That is, given a $\mathcal{G} = (V, \mathcal{E}, L, \Gamma, \xi^T, \lambda^T)$ quantum network and a virtual link at time t such that $edge \in \mathcal{E}, \Gamma(edge, t) = True$, the following statement is true:

$$\exists s_1, e_1, s_2, e_2 \in V, s_1 \neq e_1 \neq s_2 \neq e_2 : edge \in path_{(s_1,e_1)} \wedge edge \in path_{(s_2,e_2)}$$

where $path_{(s_1,e_1)}$ and $path_{(s_2,e_2)}$ are paths discovered by nodes s_1 and s_2 respectively using $\mathcal{E}_t$.

What this means essentially, is that considering each pair of vertices from V, and the shortest path between them, $edge$ is part of several such shortest paths. We could then, characterise any edge in a graph by the measure of how many shortest paths it is part of. This will be a key idea used for the technique presented in this chapter.

7.3 Approach taken

Previously, we have assumed that the traffic of the network is uniformly distributed. That is, given a demand $(s, e, \#EPR\ \ pairs, t)$, s and e are sampled from the set of vertices following a uniform distribution. For that reason, it is equally likely to seek the shortest path between any two nodes in the network. We can make use of

this fact and perform the link prediction technique. Link prediction means in our case that nodes can use their

- initial knowledge about the network topology for a given time window and
- knowledge about the traffic (demands)

to "predict" whether or not a link is still available in the network topology. For this reason, they need these two types of input data to perform link prediction. As far as the first input data is concerned, for real-world networks, each node can use historic data to predict the distribution of traffic over time. What this would imply, is that these nodes can observe the traffic from previous days to refine what is the expected traffic at each times. Then, they can see what can be accounted for each time window when it comes to incoming demands. In the second case, we can assume that nodes in the network perform some form of clock synchronisation and know the starting time of certain time windows. They would also have knowledge about the initial topology of the network at the starting time of such time windows.

So after how many demands will an edge be consumed? Let p_{edge} be the probability that an edge is consumed for a demand.

Let $path(s, e)$ be the shortest path between two vertices. Then we consider the following indicator function which determines whether or not an edge is in the shortest path: $I_{path(s,e)}(edge) : E \mapsto \{0, 1\}$.

$$I_{path(s,e)}(edge) = \begin{cases} 1 & edge \in path(s, e) \\ 0 & otherwise \end{cases}$$

Using it, we can define p_{edge}:

$$p_{edge} = \sum_{s,e \in V, s \neq e} \frac{I_{path(s,e)}(edge)}{\frac{N*(N-1)}{2}} \tag{7.1}$$

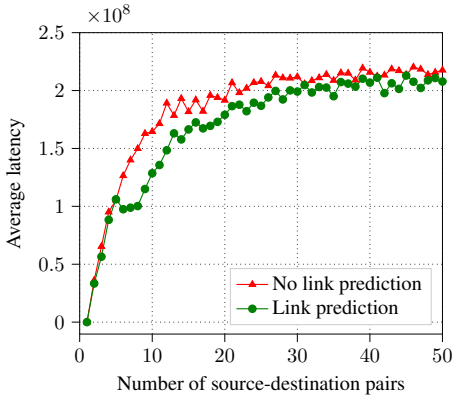
We can now consider the probability that it is consumed by the kth demand and this yields a geometric distribution with the success probability of $Pr_{edge}(X = k)$. The expectation value for this distribution is $\frac{1}{p_{edge}}$. This is the expected number of demands after which an edge will be consumed.

So what each node does when performing link prediction, is that they compute this expectation value for all of the edges in the graph. Based on the time elapsed from the start of the time window, they can estimate how many demands have been served already. If certain edges are expected to be consumed, then they perform Path Discovery in a graph where such edges are unavailable. Hence, each node stores a collection of link consumption times for the edges in the topology.

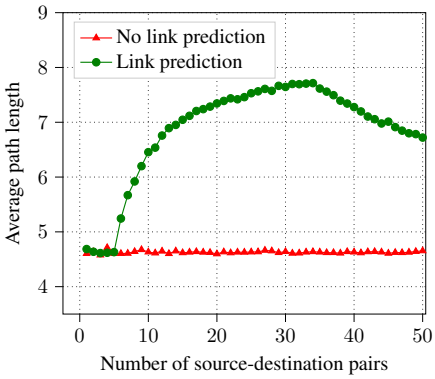
7.4 Results and discussion

In this section, we compare the proposed link prediction approach with the classical initial knowledge approach. As mentioned, link prediction uses local information

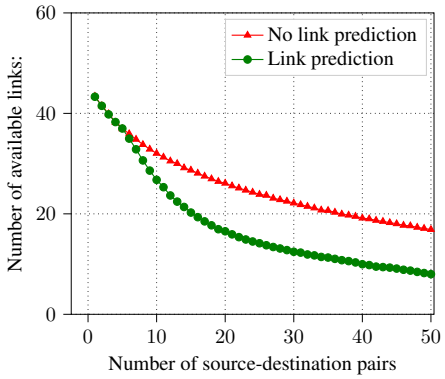
about the time window to estimate the number of demands and knowledge about the distribution of traffic.



(a) Average latency.



(b) Average path length.



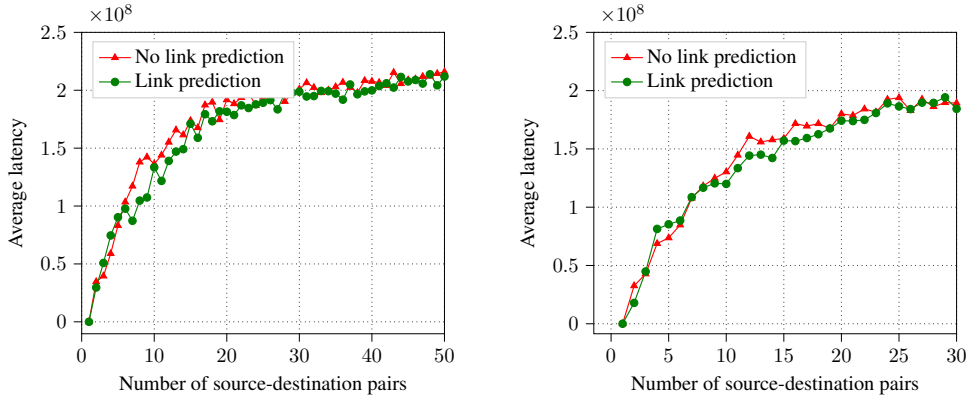
(c) Average number of links remaining.

Figure 7.1: Link prediction starting from an initial knowledge approach in a quantum network with 32 nodes. The virtual graph with 48 links was constructed deterministically with parameters $dth = 2$ and $maxdth = 2$.

The first results for link prediction in a deterministically created quantum network are shown in figure 7.1. The quantum network had 32 nodes. The underlying virtual graph with 56 links was constructed deterministically with parameters $dth = 2$ and $maxdth = 2$.

What can be observed in subfigure 7.1a is that from 6 demands the link prediction approach starts to outperform the initial knowledge approach. The magnitude of difference in average latency is higher for earlier demands (until around 30). However, link prediction yields a lower average latency overall for this topology.

As mentioned in section 7.3, Path Discovery is performed in an updated graph based on link consumption times. In subfigure 7.1b from 6 demands the curve for



(a) The virtual graph with 56 links was constructed deterministically with parameters $dth = 4$ and $maxdth = 4$.

(b) The virtual graph with 60 links was constructed deterministically with parameters $dth = 8$ and $maxdth = 8$.

Figure 7.2: Link prediction starting from an initial knowledge approach in a quantum network with 32 nodes.

link prediction increases and reaches a path length more than 7 at 30 demands. The reason for this change is the Path Discovery performed in an updated graph. Once certain links are expected to be consumed, alternative paths are discovered. These paths are longer on average than those discovered with the classical initial knowledge approach.

In subfigure 7.1c the difference in link consumption rate can be observed. Link prediction uses existing links faster than the classical initial knowledge approach. This is related to the previous observation, that it chooses longer paths on average.

After the $dth = 2$ case we have considered a quantum network with a virtual graph constructed with parameters $dth = 4$ and $maxdth = 4$. As can be observed, we obtain the advantage in this case as well. The link prediction algorithm, similar to the $dth = 2$ case starts to take effect from 6 demands and gives considerable advantage for around 10 more demands. For later demands, an advantage is still achieved with link prediction, although there is a smaller difference between the two approaches.

At last, we conducted the simulation in a quantum network with 60 virtual links constructed deterministically. The parameters $dth = 8$ and $maxdth = 8$ were used. Figure shows that the performance advantage gained by link prediction still holds for a certain number of demands in this topology as well.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

Realising a quantum internet in practice would allow a perfectly secure way of communication. Quantum networks would be provisioning quantum entanglement between participating nodes. These nodes can then use shared entanglement for quantum communication. The iterative use of the operation called entanglement swap allows nodes to share entanglement even if there is no direct physical quantum communication channel between them. With this approach, far away nodes can also share entanglement and be able to transfer quantum information. Entanglement is, however, a finite resource that needs to be generated and can be consumed. This characteristic results in a dynamically changing network topology.

Modelling a quantum network becomes difficult with current network representations due to its dynamic nature. Therefore, in this thesis, we define a quantum network as a temporal network. This definition allows us to describe the dynamically changing nature of the network. We can also incorporate time windows in which entangled links can be used for quantum communication. This model is useful for describing the lifecycle of quantum networks. In practice, we envision that traffic engineering will play a key role in quantum network routing. Our proposed model serves as a perfect basis for the mathematical and statistical analysis of the network.

We have then considered a centralized routing approach for the Path Discovery. Nodes that would like to share an entangled link with another remote node discover a path based on their local state of the network topology. Our main focus lay on the novel problem of determining the average latency in a quantum network based on information propagation approaches. Determining the average latency for multiple demands proves to be difficult with analytical approaches. Therefore, we have carried out numerical simulations to determine these values for a fixed number of demands. These simulations were carried out in network models with different distance threshold values. We have further defined the initial knowledge, local knowledge and global knowledge approaches. Each represents a certain level

of information propagation. Our results showed that with the initial knowledge approach we do not gain an advantage with a higher value of distance threshold in the deterministically created topology. Based on our results, as the level of information propagation is increased, the expected advantage of the distance threshold is gradually observed. With the global knowledge approach, our results show a clear advantage in the topologies created with higher distance threshold values. Finally, the obtained simulation results have further shown that increasing the level of information propagation (e.g. comparing the initial knowledge and global knowledge approaches) results in a decreased average latency.

Our proposed *link prediction* algorithm can be used to considerably decrease the average latency of demands. One key factor for nodes to determine is the current point in time in the time window. Historic data on the distribution of demands in time windows can be used to estimate this value. Nodes can then in practice use this estimate and their locally stored knowledge about the topology to perform link prediction. It allows them to predict which entangled links have likely been already consumed in the network since the start of the time window (although their locally stored knowledge might still include them as available entangled links). Link prediction is useful when nodes do not have a global knowledge about the network topology. In such a case, the existing level of information propagation and link prediction can be used together to achieve lower average latency.

8.2 Future Work

8.2.1 Fidelity constraints

In the models that we have used, we were focusing on the latency of demands. The quality of entanglement described by its fidelity, however, affects the noise involved in quantum communication. In our models, a minimum value for the fidelity of entangled links was included (see [CRDW19]). This value affected the choice of the value of the threshold time. However, nodes of the network may request entangled links with a fidelity higher than this minimum value. Thus, nodes could specify the requested minimum fidelity for their demand. Incorporating this feature would make the simulation model more realistic.

8.2.2 Refined link prediction

Our proposed link prediction algorithm uses a statistical measure to compute the expected time after which links are going to be consumed. This measure is similar to the *betweenness centrality* measure in networks. It would be worth exploring, what advantage can be gained by performing link prediction by using other centrality measures.

8.2.3 Netsquid

The quantum network simulator called Netsquid [QuT18] allows simulating the decay of quantum information over time. It further incorporates modelling the noisy operations of quantum communication. Netsquid allows simulations with specific implementations of quantum networks such as nitrogen-vacancy centers, atomic ensembles, trapped ions, etc.. Currently under development, Netsquid could be used to devise routing algorithms in the physically-realistic setting.

8.2.4 Load-balanced centralized shortest path finding

Our approach was using Dijkstra's algorithm for finding the shortest path between source and destination. A node might discover the same path each time, however, when there are several possible shortest paths towards the same destination. This results in the traffic flowing the same direction. A simple theoretical solution to such a problem would be sampling a path from all the shortest paths uniformly at random each time. However, further, more sophisticated strategies could be devised based on the properties of the topology. We further expect that similarly to the classical internet, such decisions will be mostly determined by performing traffic engineering. Thus, what could be a potentially general way of effectively balancing the load between alternative paths, remains to be future work.

Bibliography

[AD] Tim Coopmans Leon Wubben Filip Rozpędek Matteo Pompili Arian Stolk Przemysław Pawełczak Robert Knegjens Julio de Oliveira Filho Ronald Hanson Stephanie Wehner Axel Dahlberg, Matthew Skrzypczyk. A link layer protocol for quantum networks.

[Bar16] Albert-László Barabási. *Network science*. Cambridge university press, 2016.

[BB84] Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing, booktitle = Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, volume = "560", pages = "7 - 11", address = "new york",. 1984.

[BBC+93] Charles H. Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K. Wootters. Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels. *Phys. Rev. Lett.*, 70:1895–1899, Mar 1993.

[BDCZ98] 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.*, 81:5932–5935, Dec 1998.

[Ber13] Hensen B. Pfaff W. Koolstra G. Blok M. S. Robledo L. Taminiau T. H. Markham M. Twitchen D. J. Childress L. Hanson R. Bernien, H. Heralded entanglement between solid-state qubits separated by three metres. *Nature*, 2013.

[Bol01] B. Bollobás. *Random Graphs*. Cambridge University Press, 2001.

[Cal17] M. Caleffi. Optimal routing for quantum networks. *IEEE Access*, 5:22299–22312, 2017.

[CLRS09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.

[CRDW19] Kaushik Chakraborty, Filip Rozpedek, Axel Dahlberg, and Stephanie Wehner. Distributed routing in a quantum internet. 07 2019.

[DJW98] Steven H. Strogatz Duncan J. Watts. Collective dynamics of 'small-world' networks. *Nature*, 1998.

[DP04] Ellie D'Hondt and Prakash Panangaden. Leader election and distributed consensus with quantum resources. 2004.

[Eke91] A. K. Ekert. Quantum cryptography based on Bell's theorem. *Physical Review Letters*, 67:661–663, August 1991.

[ESW16] T. Islam I. Kerenidis E. Schoute, L. Mancinska and S. Wehner. Shortcuts to quantum network routing. 2016.

[GI18] Laszlo Gyongyosi and Sandor Imre. Decentralized base-graph routing for the quantum internet. *Phys. Rev. A*, 98:022310, Aug 2018.

[Gil59] E. N. Gilbert. Random graphs. *Ann. Math. Statist.*, 30(4):1141–1144, 12 1959.

[GR01] Chris Godsil and Gordon Royle. *Algebraic Graph Theory*, volume 207. 01 2001.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 212–219, New York, NY, USA, 1996. ACM.

[GWZ+08] Alexander Goebel, Claudia Wagenknecht, Qiang Zhang, Yu-Ao Chen, Kai Chen, Jörg Schmiedmayer, and Jian-Wei Pan. Multistage entanglement swapping. *Physical review letters*, 101:080403, 09 2008.

[Hen15] Bernien H. Dréau A. E. Reiserer A. Kalb N. Blok M. S. Ruitenberg J. Vermeulen R. F. L. Schouten R. N. Abellán C. Amaya W. Pruneri V. Mitchell M. W. Markham M. Twitchen D. J. Elkouss D. Wehner S. Taminiau T. H. Hanson R. Hensen, B. Loophole-free bell inequality violation using electron spins separated by 1.3 kilometres. *Nature*, 2015.

[HJB99] R.Albert H. Jeong and A.-L. Barabási. Internet: Diameter of the world-wide web. *Nature*, pages 401:130–131, 1999.

[HKO+12] Julian Hofmann, Michael Krug, Norbert Ortegel, Lea Gérard, Markus Weber, Wenjamin Rosenfeld, and Harald Weinfurter. Heralded entanglement between widely separated atoms. *Science*, 337(6090):72–75, 2012.

[HS12] Petter Holme and Jari Saramäki. Temporal networks. *Physics Reports*, 519(3):97–125, Oct 2012.

[IdSP78] M. Kochen I. de Sola Pool. Contacts and influence. *Social Networks*, pages 1: 5–51, 1978.

[JCM99] SF Huelga JI Cirac, AK Ekert and Chiara Macchiavello. Distributed quantum computation over noisy channels. *Physical Review A*, 1999.

[Kim08] H. J. Kimble. The quantum internet. *Nature*, 2008.

[Kle00] Jon Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing*, STOC '00, pages 163–170, New York, NY, USA, 2000. ACM.

[LNK03] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, CIKM '03, pages 556–559, New York, NY, USA, 2003. ACM.

[LSW+04] Seth Lloyd, Jeffrey H. Shapiro, Franco N. C. Wong, Prem Kumar, Selim M. Shahriar, and Horace P. Yuen. Infrastructure for the quantum internet. *SIGCOMM Comput. Commun. Rev.*, 34(5):9–20, October 2004.

[Met14] R. Van Meter. *Quantum networking*. John Wiley & Sons, 2014.

[MFF99] P. Faloutsos M. Faloutsos and C. Faloutsos. On power-law relationships of the internet topology. In *Proceedings of SIGCOMM. Comput. Commun. Rev*, pages 29: 251–262, 1999.

[Mil67] S. Milgram. The Small World Problem. *Psychology Today*, 1967.

[Moe07] Maunz P. Olmschenk S. Younge K. C. Matsukevich D. N. Duan L.-M. Monroe C. Moehring, D. L. Entanglement of single-atom quantum bits at a distance. *Nature*, 2007.

[MS17] Vera Zaychik Moffitt and Julia Stoyanovich. Temporal graph algebra. In *Proceedings of The 16th International Symposium on Database Programming Languages*, DBPL '17, pages 10:1–10:12, New York, NY, USA, 2017. ACM.

[NC10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[PE59] A. Rényi P. Erdős. On random graphs. *Publicationes Mathematicae*, 1959.

[QuT18] QuTech. A simulation platform for quantum networks. 2018.

[Rit12] Nölleke Christian Hahn Carolin Reiserer Andreas Neuzner Andreas Uphoff Manuel Mücke Martin Figueroa-Eden Bochmann Joerg Rempe Gerhard Ritter, Stephan. An elementary quantum network of single atoms in optical cavities. *Nature*, 2012.

[Sho94] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, Nov 1994.

[TBZG98] W. Tittel, J. Brendel, H. Zbinden, and N. Gisin. Violation of bell inequalities by photons more than 10 km apart. *Phys. Rev. Lett.*, 81:3563–3566, Oct 1998.

[TKM05] Seiichiro Tani, Hirotada Kobayashi, and Keiji Matsumoto. Exact quantum algorithms for the leader election problem. In Volker Diekert and Bruno Durand, editors, *STACS 2005*, pages 581–592, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[WEH18] Stephanie Wehner, David Elkouss, and Ronald Hanson. Quantum internet: A vision for the road ahead. *Science*, 362(6412), 2018.

[WHHL09] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103:150502, 10 2009.

[ZZHE94] Marek Zukowski, A Zeilinger, MA Horne, and Artur Ekert. Event-ready-detectors Bell experiment via entanglement swapping. *Physical review letters*, 71:4287–4290, 01 1994.