Master's thesis

Master's Programme in Computer Science

Communication Between Quantum Networks

Xingchang Song

June 9, 2022

Faculty of Science

University of Helsinki

Contact information

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki,Finland

Email address: info@cs.helsinki.fi
URL: http://www.cs.helsinki.fi/

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty Faculty of Science	Koulutusohjelma — Utbildningsprogram — Study programme Master's Programme in Computer Science

Tekijä — Författare — Author

Xingchang Song

Työn nimi — Arbetets titel — Title

Communication Between Quantum Networks

Ohjaajat — Handledare — Supervisors

Prof. Valtteri. Niemi

Työn laji — Arbetets art — Level Master's thesis	Aika — Datum — Month and year June 9, 2022	Sivumäärä — Sidoantal — Number of pages 42 pages

Tiivistelmä — Referat — Abstract

Quantum networking is developing fast as an emerging research field. Distributing entangled qubits between any two locations in a quantum network is one of the goals of quantum networking, in which repeaters can be used to extend the length of entanglement. Although researchers focus extensively on problems inside one quantum network, further study on communication between quantum networks is necessary because the next possible evolution of quantum networking is the communication between two or more autonomous quantum networks.

In this thesis, we adapted a time slotted model from the literature to study the inter quantum network routing problem. Quantum routing problem can be split into path selection and request scheduling. We focus on the latter considering the previous one received considerable interest in the literature. Five request scheduling policies are proposed to study the impact of preference for certain request types on entanglement generation rate. Experiments also demonstrate other factors should be considered in context of entanglement rate in communication between quantum networks, e.g., the number and distribution of requests and inter-network distance.

ACM Computing Classification System (CCS)

Computer systems organization → Architectures → Other architectures → Quantum computing
Hardware → Emerging technologies → Quantum computation → Quantum communication and cryptography
Networks → Network protocols → Network layer protocols → Routing protocols

Avainsanat — Nyckelord — Keywords

Quantum Networking, Quantum Routing, Request Scheduling

Säilytyspaikka — Förvaringsställe — Where deposited

Helsinki University Library

Muita tietoja — övriga uppgifter — Additional information

Networking study track

Contents

1 Introduction

Recently, massive progress has been made in quantum computing since Google announced its 53-qubit quantum processor Sycamore had achieved quantum supremacy [4] in 2019. Shortly after, the competition of making more powerful quantum processor resulted in the great improvement in quantum computing power [46, 43, 47]. To date, IBM claims its 127-qubit quantum process IBM Eagle [9] to be the most powerful quantum computer ever made. Though quantum computers with larger computational capacity will surely become available in the future, a quantum network interconnecting multiple quantum processors is necessary to increase the overall computational capacity of a system.

Quantum networking is now a prevailing research area due to the efficiency of some applications such as Quantum Key Distribution (QKD) [5] in quantum internet compared to that in a classical network. More applications will be enabled as quantum networking keeps developing such as clock synchronization, timing, and positioning in quantum mechanics [16, 1, 22], metrology [20, 13], and online secure quantum computation [6].

Designing and implementing a quantum network is a popular research topic in academia. Unfortunately, experience in designing early-stage classical internet cannot be directly transfer to a quantum network due to the unique underlying quantum mechanics. No-Cloning Theorem [33] shows that generating an unknown state of a qubit cannot be done while reserving the original one, which makes it impossible to "copy-paste" quantum data (qubit) in the network. Quantum entanglement is vital to a quantum network, but it presents new challenges to the realization of quantum internet because qubit can be only teleported after knowing the entanglement information of the corresponding qubit and the location of the target [24].

Simply speaking, a quantum network is expected to realize the transmission and manipulation of qubits between far-away locations, in which quantum entanglement plays a significant role because remote locations are expected to share one or more entanglement pairs beforehand. Multiple strategies have been put forward to distribute entanglement in long distance, e.g., satellites [44] and drones [30] have been utilized as trusted entanglement emitters. More commonly seen is the case where entanglement generator such as Nitrogen-Vacancy centers in diamond [18, 29] distributes entanglement pairs to two neighboring locations via optical fibres.

Considering the direct entanglement via optical fibres between adjacent locations has length limit [25], two remote locations in larger range are expected to extend the direct entanglement length by “stitching” one or more repeaters along the way, who serve as relays in the longer end-to-end entanglement chain. The intermediate repeaters help to directly connect its entangled neighbor by performing entanglement swapping. A lot of work [8, 10, 11, 21, 45, 39] has been conducted in the repeater-based quantum network, most of which was focusing the routing problem (path selection [8] and request scheduling [10] included) within the network.

Although many works are related to problems within one quantum network, few attentions have been paid to the inter quantum network even though a natural evolution direction of quantum network is the communication between two autonomous quantum networks. An easy-to-understand example is that as seen in many works (e.g., [8, 36]), growing size of a quantum network downgrades the performance of routing algorithms drastically, which can be addressed by splitting a huge quantum network into two or more sub-networks.

In this thesis, inter quantum network problem is studied. Followed by the idea diving quantum routing problem into two sub-problems: path selection and request scheduling [10], we put emphasis on the latter because extensive research has been conducted on the previous one, while the latter received less interest. We study the impact of preference for a certain request types on the overall entanglement rate of the network. Further more, we study possible influential factors in the performance of the network, e.g., inter-network distance and the number and distribution of requests.

The remainder of the thesis is arranged as follows. Chapter 2 will briefly introduce the aforementioned quantum mechanics and repeater-based network in details. Chapter 3 will examine some of the related work. Chapter 4 will present an inter quantum network model, followed by the experiment in chapter 5 and performance evaluation in chapter 6. Finally, chapter 7 concludes this work.

2 Preliminaries

In this chapter, Quantum Mechanics and repeater-based quantum internet are introduced. Section 2.1 provides brief introduction to quantum mechanics for readers not familiar with the topic.

2.1 Quantum Mechanics

In this section, basic concepts in quantum mechanics are introduced. More details of linear algebra, bra-ket notation, Bell state, and Bell measurement can be found in [26, 33, 14].

1 Qubit. In a classical computing system, a bit is the smallest information unit that is either in state 0 or state 1. The counterpart in quantum computing system is a quantum bit, or qubit [7], which distinguishes itself by the capability to represent state 0 and state 1 simultaneously. Therefore, n bits in classical computing can only represent one of the 2^n states, whereas n qubits can represent at once all 2^n states. Mathematically, a quantum state can be regarded as the weighted combination of two base states 0 and 1, where state 0 can be written as $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ or $|0\rangle$ [14], and state 1 can be written as $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ or $|1\rangle$ [14].

2 Superposition. Technically and mathematically speaking, a qubit is allowed to randomly superpose two states to form $\begin{bmatrix} a \\ b \end{bmatrix}$, or $a|0\rangle + b|1\rangle$, in which $a^2 + b^2 = 1$, and such state is defined as superposition [7].

3 Measurement. Interestingly, unlike a bit can be read in a classical computing system due to the deterministic characteristic of bits, a qubit needs to be measured in a quantum computing system and it will collapse to a classical state after measurement [7], which means the destruction of its current state.

4 Entanglement. Now that a qubit has possible two base states, 2 qubits have possible $2^2 = 4$ base states $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$. Given a state of two qubits, some can be interpreted as the tensor product [26] of the two, for example, state $|01\rangle$ or $\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$ can be interpreted as the tensor product of $|0\rangle$ and $|1\rangle$. States that cannot

be interpreted as so, are called entangled states. For example, no tensor product of two qubits can be found to represent a Bell state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ [33] , which is also known as a maximally entangled state, or EPR pair. Experiments have shown quantum entanglement will not be influenced even if two entangled qubits are physically placed far apart [42]. And measuring one of the entangled qubits will instantaneously cause the collapse of the other.

5 No-Cloning Theorem. In quantum computing, No-Cloning theorem [7, 42] states that one cannot simply generate an exact copy of an unknown quantum state, which renders store-and-forwarding packets impossible to be applied in quantum networks.

6 Quantum Teleportation. However, the functionality of transferring a state from one end to the other is essential to the realization of a feasible quantum network. Quantum teleportation [42] is therefore introduced to tackle such issue. Specifically, as illustrated in Figure 2.1, quantum teleportation starts by two ends (Alice and Bob) having one qubit of an EPR pair (a, b) respectively. Alice needs to transfer her data qubit d to Bob with the help of qubit a. After a certain manipulation to qubit a and d, Alice is able to perform a Bell Measurement on them and sends the corresponding 2-bit result via classical network link to Bob, who is ensured to reproduce data qubit d at his side by the received classical bits. In this process, the entangled pair is consumed and there is no actual qubit travelling from Alice to Bob. More generally, $2n$ classical bits and n preallocated EPR pairs are required to "transport" n qubits.

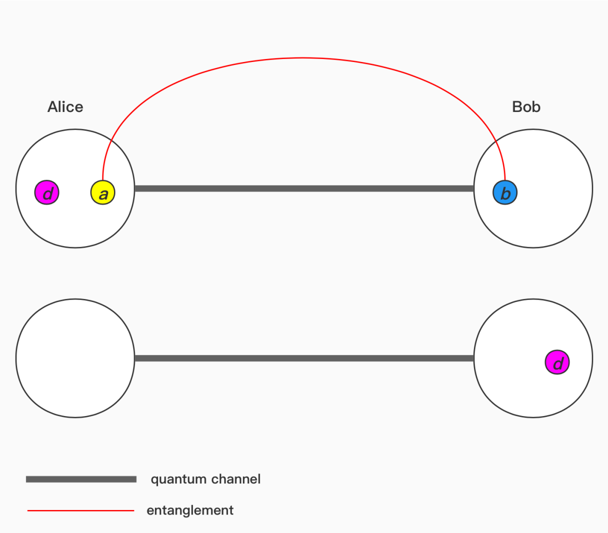

Figure 2.1: Quantum teleportation

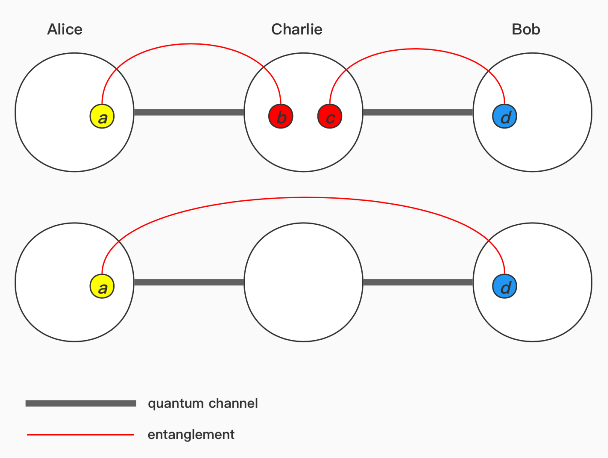

Figure 2.2: Entanglement swapping

7 Entanglement Swapping. As seen from the description of quantum teleportation above, quantum entanglement is a feasible means of building connection between two distant nodes in a quantum network. If two far-end nodes are to communicate via teleportation, they are required to create an entanglement pair regardless of their long distance. Directly establishing entanglement via long-distance quantum channels (e.g., optical fibres) between two ends is considered infeasible due to the exponential drop of success rate caused by the inherent "lossiness" of channels [39]. Entanglement swapping [42] is a technique that can “deliver” entanglement via intermediate nodes. For example, in Figure 2.2, if Alice wants to share an entanglement pair with Bob but there is Charlie between them, Alice and Bob can build an EPR pair with Charlie separately, so Alice has qubit a entangled with Charlie’s qubit b and Bob has qubit d entangled with Charlie’s qubit c. If Charlie performs a Bell Measurement on his qubit b and c, and sends the corresponding result to Bob. Bob is able to “correct” his qubit d so that a and d will form an entanglement. In this way, Charlie helps connect Alice and Bob as a “middleman”. In another way, it can be viewed that Charlie teleports his correlation with Alice to Bob, who reproduces that entanglement based on Charlie’s message.

8 Fidelity. Entanglement, as seen from the discussion above, is certainly the building block of quantum networks. However, generating a perfect entanglement pair in practice is hardly possible due to the noise and imperfect operations in the quantum system. Fidelity [11, 23] is a unique index that quantifies the closeness between a practical state and a theoretical state. It generally describes the quality of the generated entanglement. Numerically, fidelity falls within the range between 0 and

1 and higher value represents higher quality entanglement.

9 Decoherence. Decoherence [11, 23] is a process during which quantum information is gradually lost over time and one of its external manifestations is the decrease of fidelity. Decoherence requests better solutions for the utilization of generated EPR pairs before they decohere.

10 Entanglement Distillation. Entanglement distillation [11, 23] is a process that provides two nodes with a higher quality entanglement pair by consuming two or more low quality ones. Some considerations should be made when implementing entanglement distillation, one of which is the availability of it if the decoherence time is too short and another is the strategy of implementing it, e.g., should it be performed before or after (how many) entanglement swapping [23].

11 Quantum Error Correction. Quantum error correction, or QEC, first proposed by Shor [40], is a technique that can protect quantum information from decoherence or noises in the system. It is generally considered demanding in terms of required resources such as number of qubits and quality of states, and it is predicted not to be available in the near future [11, 23].

2.2 Repeater-based Quantum Internet

Long-distance communication between two far-away nodes in quantum network cannot be realized by directly transmitting qubits between them because qubit is fragile due to errors and decoherence when transmitting in lossy channels. Moreover, No-Cloning theorem prohibits intermediate nodes to read-and-copy quantum data.

Connection between two far-away end nodes can also be realized by a trusted intermediate node, e.g., a quantum satellite [44], to serve as an entanglement distributing agent. In 2016, China launched the world's first quantum communications satellite Micius [44]. It has successfully delivered simultaneous streams of entangled photons to its original ground stations, which are 1,200 kilometers away.

It is more commonly seen, however, that quantum repeaters serve as intermediate nodes to extend the length of direct communication of two nodes. Therefore, entanglement between two distant nodes can be created by chaining multiple repeater nodes along the path. More specifically, as illustrated in Figure 2.3, a repeater chain consists of multiple repeaters directly entangled with its predecessors and successors, source node entangled with the first repeater and destination node entangled with the last repeater. By performing Bell Measurement [33] on each of the repeaters and sending corresponding results to destination via classical links for final "correction", communication between source and destination can be successfully established.

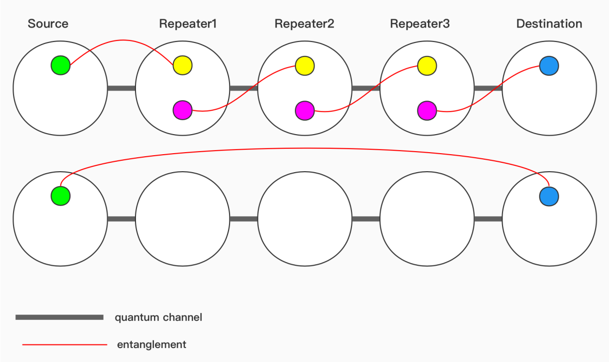

Figure 2.3: A 3-repeater based quantum network

Given the inevitable loss of quantum information caused by optical fibres, some had proposed quantum satellite networks (QSN) where quantum satellites on low earth orbit (LEO) are used as repeaters because quantum links in vacuum are hardly lossy and cause only negligible loss of photon. [36]

We discuss mainly repeater-based quantum network with optical fibres as quantum channels later in the thesis.

3 Related Work

Since no quantum network in practice exists for the time being, many different studies are conducted concentrating on different components needed for a feasible quantum internet. In this thesis, definitions of the four layers from the quantum network stack [11, 23] will be introduced and referred to later, and previous studies will be also examined under the quantum network stack framework.

Dahlberg et al. and Kozlowski et al. [11, 23] proposed a quantum network stack, in which layers are divided by the expected functionality, as an equivalent of classical TCP/IP network stack. More specifically, the quantum network stack is divided into four layers as shown in Table 3.1: physical layer, link layer, network layer and transport layer. The physical layer will conduct heralded entanglement generation on adjacent nodes and report the results of the attempts to the corresponding nodes. The link layer is expected to provide the upper layer (network layer) with service of robust entanglement generation so that the network layer will acknowledge whether adjacent nodes are entangled or not. The functionality of robust entanglement generation is achieved by link layer instructing physical layer to perform entanglement attempts multiple times until success or time-out. The network layer will build long-distance entanglement between nodes that are not directly connected. Many research interests have been shown on this layer, e.g., optimal routing, path selection, request scheduling. Finally, the transport layer will perform qubit transmission via teleportation according to the entanglement information from network layer.

Pirker and Dür [37] designed an abstract quantum network stack, in which the function-

Application	Utilization of the transmitted qubits
Transport	Qubit transmission
Network	Long distance entanglement
Link	Roubust entanglement generation
Physical	Attempt entanglement generation

Table 3.1: A quantum network stack. Adapted from [11].

alities of each layer are defined in technology-independent manner. The main difference between [37] and [11, 23] is that the former is designed for quantum network based on Greenberger–Horne–Zeilinger (GHZ) states entanglement [19] while the latter is aiming for network based on Bell pairs entanglement. Even though GHZ states establishment is more demanding and prone to noise [34, 28], Pirker and Dür believed GHZ states entanglement scheme requires less storage than complete bipartite approach [37].

The remainder of this section is divided into 4 subsections focusing on studies on different layers: studies on physical layer, on link layer, on network layer and on transport layer. Studies on network layer will be given more attention as it is strongly related to our study in this thesis.

3.1 Studies on Physical Layer

Wang et al. [41] reported a drastic improvement of estimated coherence time in a single ion qubit from around ten minutes to over an hour, which offers potential improvement of quantum repeaters' memory time. Their research can be further extended to develop multi-qubit memory with individual storage of single qubit given longer coherence capability. [41]

Lee et al. [27] designed a router architecture consisting of multiple repeater nodes to produce higher fidelity entanglement without extensively sacrificing entanglement generation rate. The advantage of "upgraded" repeaters is obtained from the superiority of local, low-loss channels compared to lossy photonic links of "router-less" ones.

3.2 Studies on Link Layer

Dahlberg et al. [11] proposed and implemented a link layer protocol named Entanglement Generation Protocol (EGP), which promised to provide robust entanglement between adjacent nodes. Technically speaking, lower physical layer is equipped with Midpoint Heralding Protocol (MHP), which polls link layer periodically to ask for its willingness of entanglement. When EGP receives entanglement creating instructions from upper network layer and computes necessary parameters (e.g., minimum fidelity, minimum waiting time, available quantum memory resources) to estimate corresponding feasibility of the request, it waits for the next MHP poll to trigger the request. Upon receiving a trigger signal from

EGP, MHP will make multiple entanglement attempts until the process succeeds or has timed out.

Pompili et al. [38] revised the link layer protocol proposed in [11] and pointed out three challenges in the realizations of EGP and MHP, which are the consistency of distributed queues between adjacent nodes, the possible synchronization errors of EGP triggering and entanglement request mismatches. Corresponding solutions regarding aforementioned challenges are put forward. A centralized scheduler with the knowledge of the network is assigned to replace distributed queues in adjacent nodes so that inconsistencies become very rare. The task of synchronization of emitting photons is left to physical layer (MHP) rather than being done by EGP, because physical layer can provide higher accuracy synchronization clocks. Normally, an entanglement establishment attempt between neighboring nodes comes with the result of success or failure. The result is given by the heralding station in the middle of two nodes. Pompili et al. [38] suggested that MHP peers check for entanglement request mismatches instead of heralding stations, because MHP peers have to communicate before entanglement anyway, and it can simplify the jobs of heralding stations. However, they did not implement their suggestion yet.

3.3 Studies on Network Layer

Kozlowski et al. [23] designed and presented a quantum network data plane protocol (Quantum Network Protocol or QNP) on top of link layer [11], and it defines the required information from control plane (e.g., routing, signaling) for single end-to-end communication on network layer. More specifically, a virtual circuit is initiated after routing and signaling are done and QNP demands intermediate nodes to perform entanglement swapping immediately when they share entanglements both with its predecessor and successor. It is beneficial because decoherence caused by qubits idling in memory while waiting to be used is identified to be one of the factors that lead to the reduction of end-to-end entanglement fidelity.

Pirker and Dür [37] focused on the reliability of a network when a node leaves. They suggested using symmetrization and shielding qubits to provide redundancy of network information. The regional (inter-network) routing problem where nodes from different regions (sub-networks) request communication is also discussed in their paper [37]. The proposed solution is to construct a Steiner tree to find the optimal route among routers that are the gateway nodes of regions.

Caleffi [8] designed a routing protocol aiming at finding the most optimal route given two arbitrary non-adjacent nodes. The optimality of a route is based on the metric of end-to-end entanglement rate, which is computed by the relevant derived mathematical closed-form expression. The time complexity of the computing process, in which end-to-end entanglement rate is computed, is correlated with the number of simple paths in the network, which needs to be addressed as quantum Internet continues to scale up. Researchers has put forward corresponding solutions, e.g., by utilizing genetic algorithms or zone-based routing [3, 2].

Li et al. [28] focused on addressing request scheduling problem under the situation of receiving multiple user requests simultaneously. More specifically, all requests in the network are directed to one central controller. One of the functionality of the central controller is to select multiple paths for the user in case the optimal path breaks due to potential network failures. The other is to coordinate simultaneous flows so that no edge will be overwhelmed. Three different resource allocation strategies are provided: proportional share, progressive filling and propagatory update [28]. Proportional share strategy results in smallest variance of edge capacity usage distribution but smallest throughput and edge usage rate, while progressive filling obtains greatest throughput and edge usage rate but highest distribution variance in edge capacity. Clearly small-scale lattice topology experimented in the paper simplified the request scheduling problem and reduced the time complexity of algorithms compared to that of bigger-scale topology. The authors have also identified the limitation of single scheduler in a large-size network and reckoned a queuing model will be beneficial to provide guidelines for queuing requests.

Picchi et al. [36] developed a Quantum SDN (Software-Defined Networking) architecture [35] to find a best end-to-end path in terms of entanglement generation rate in a Quantum Satellite Network (QSN). One of the strengths of SDN architecture is that it splits control plane apart from data plane so that the operation logic is clear. Picchi et al. made a comparison between three controller algorithms: Modified Random Walk, Ant Colony Optimization [15] and Dijkstra's algorithm. The former two are distributed algorithms while the last one is centralized. Surprisingly, Dijkstra's algorithm outperformed the other two in experiments and successfully achieved greater average end-to-end entanglement generation rate, which is contradictory to the conclusion in a previous study [8]. One of the potential reasons can be the impact of the free space environment on quantum links.

3.4 Studies on Transport Layer

Dai et al. [12] proposed a quantum queuing delay (QQD) model utilizing dynamic programming technique to shorten the average queuing length of a quantum processor. More specifically, their model considered the scenario of one receiver with multiple quantum data sources and managed to decrease the queuing delay exponentially by increasing the memory size of a node. However, the model is working under the assumption that entanglement can be reserved in quantum memory for indefinitely long time, which is technically not feasible at least in the foreseeable future. Moreover, it employs a dropping policy when the queue is not capable of holding more quantum data, which requires the re-transmission functionality implemented on the sender side due to the restriction of No-Cloning theorem. The dropping policy should be given more considerations given the difficulties of establishing connections between nodes far apart in the network layer. Improvements of proposed dropping policy can be done by assigning the priority of quantum data in queue, which balances the acceptable queue length and the factor of either the importance of quantum data or difficulties of transporting a qubit.

4 Design

In this chapter, a proposed system model from other work [10] will be introduced, to which we have made slight modifications. First, we will present the structure of the quantum network and its key components, whose implementation details will be further discussed. Then we will describe the event flows in the time-slotted model. Because the model adopted in this work originated from work of Cicconetti et al [10], interested readers are referred to their paper for better understanding.

4.1 Quantum Network Structure

This section presents the network structure from two angles: components and event flows. Components represent the physical logic while event flows gives transaction logic.

4.1.1 Components

A quantum processor or a quantum computer is a system on which quantum applications are running [10, 39]. It is similar to an end node or a host in classical network. In a quantum network, it is the one that make requests of establishing end-to-end entanglements with other processors. A quantum processor is usually connected with quantum repeaters or directly to other quantum processors via quantum and classical channels. It is equipped with limited quantum memory to hold a few qubits and capable of performing quantum teleportation and entanglement swapping.

A quantum repeater, as mentioned before in section 2.2, is a special quantum device that connects with quantum computers and other quantum repeaters via quantum links. Like a quantum processor, it is also equipped with delicate hardware to perform quantum swapping and has limited number of quantum memory slots to store qubits for a short period of time [10, 39]. Classical channels are used to deliver control messages and correction results to quantum repeaters. A quantum processor includes all necessary functions of a quantum repeater [39]. Generally, both quantum processors and quantum repeaters are referred as nodes in quantum networking. We will use nodes and quantum processors and quantum repeaters interchangeably hereafter in the thesis.

A quantum channel is physical connection between two adjacent nodes. It usually exists physically in the form of optical fibre that allows the transmission of qubits. A quantum channel is inherently lossy and the success rate of establishing entanglement between two neighboring nodes decreases exponentially with node distance [32]. More details of quantum channel will be discussed later in section 5.2.

A classical channel is also a physical connection between neighbors, but it transmits classical information in the way of classical Internet. In our model, classical channels are typically used to deliver control information, e.g., link states, and correction messages with propagation delay. The details of classical channels will be also introduced later in section 5.2.

A request collector will collect communication requests from a network periodically.

A quantum network controller is an individual device that controls the whole network but does not connect to any of the nodes in the network. The following list contains responsibilities of a quantum network controller [10].

1. Retrieving requests from the request collector. The controller will notify the source and destination nodes of each request whether it can be fulfilled or not.
2. Collecting entanglement link states (activated/ non-activated) from each node in the network.
3. Deciding which requests to serve first and serve each of them with the best path according to whichever metric selected in the system.
4. Notifying the intermediate nodes along the selected paths of the destination node as well as the specific links to perform entanglement swapping operations.

4.1.2 Event Flows

A repeater chain, as introduced before in section 2.2, exists when all nodes are entangled with their neighbors at the same time. This will not be possible if any of the links along the path is broken. These impose challenges to the working logic of quantum nodes in network. It is noticeable that such delicacy of the system requires certain level of time synchronization [39], which can be achieved by implementing a time-slotted model. Typically, in a time-slotted model, time is divided into small chunks and each one of these small chunks is called a time slot. In a single quantum network, perfect time synchronization of all nodes is not necessary [10, 39]. However, in a scenario of two quantum networks communicating, time synchronization should be done at an inter-network level, which means that one network needs to be synchronized with the other. More specifically, two networks should agree on issues such as the start and the ending time of the system and the duration of a time slot.

Generally speaking, a time slot in the utilized system model can be divided into two phases by the readiness of qubits to perform quantum teleportation. The first phase is the preparation phase, in which the system will try to satisfy as many requests as possible and produce corresponding end-to-end entanglements. The second phase is the ready phase, in which end-to-end entanglements are utilized in the manner depending on the requirements of specific applications.

Of these two phases, we are putting more emphasis on the first phase because the second phase is related to transport layer [11, 23], which is outside of our research interest. From the perspective of the proposed quantum network stack [11], the first phase is responsible for the functionalities in physical layer, link layer, and network layer.

The implementation of physical layer functionality in [11] requires multiple entanglement attempts. However, in our model, there will only be one entanglement attempt at the beginning of each time slot. Multiple local entanglement attempts can be considered as one of the possible improvements of the model, as it would increase the likelihood of robust quantum links.

We will present detailed event flows of the preparation phase in the following [10].

1. At the start of each time slot, as discussed above, each node attempts establishing local entanglement with its neighbors once and reports the corresponding results (success or failure) to the network controller over classical links [10]. After all nodes have reported their successful local links to the network controller, the network controller will have the knowledge of network topology at this time slot. Note that at each time slot, the network controller will likely have a different global view from the previous one. It is also worth noticing that the time required by the controller to build a global view of the network is not constant [10]. It is strongly influenced by the network size, the distance between each node and the controller, the communication methods, and the information processing speed of the controller [10].

2. Now that the controller has the global topology, the next step is quantum routing in

which the controller determines which qubits in which quantum memory slots need to be measured to form end-to-end entanglements. Note that the controller may or may not fulfill all requests in a time slot and the decision of serving a request or not is affected by multiple factors, as will be introduced later in section 4.2.1. The result of this step is that the corresponding paths of the fulfilled requests, and related nodes on the paths [10].

3. Based on the steps above, the controller notifies all intermediate nodes on a repeater chain to perform entanglement swapping operation. It is done by indicating each of them the quantum memory slots in which the qubits to be measured lie.

4. After measurement on qubits, all nodes (except the destination node) of a path send correction messages via classical network to the destination node. Note that details of the underlying classical network are not necessarily known to all nodes. A node is able to send the measurement results to the destination node with the help of classical routing protocols, discussion of which is out of the scope of the thesis.

5. After the destination node of a request performs corrections on its local qubits utilizing the correction messages from all nodes along the path, the end-to-end entanglement is eventually established and is ready for use by distant nodes according to upper-layer applications. This happens in the ready phase.

4.2 Quantum Routing

Quantum routing is different from classical routing because of the underlying quantum mechanics that was discussed earlier. One of the most noticeable features of a quantum network is that a quantum link will vanish physically in a short time because it is more fragile than a classical link. Another difference is that a classical link can be shared by multiple nodes at the same time while a quantum link can only be shared by two nodes because of the monogamy property [17] of an entanglement.

As pointed out in previous work [10], a quantum routing problem can be divided into two smaller problems: path selection and request scheduling. Following similar analysis, an inter quantum network routing problem can also be divided into path selection and request scheduling.

In terms of path selection, the prior knowledge is still applicable to inter quantum network routing. In our work, we simply use Dijkstra's algorithm for path selection. An inter-network path between two sub-networks has usually three parts: two intra-network paths in two sub-networks, separately, and a bridge that connects these two sub-networks. Considering that bridges are known to both sub-networks, finding an optimal inter-network path can be converted into finding two intra-network paths whose connection gives best performance. More specifically, if there are two sub-networks A and B, and a source node in sub-network A wishes to communicate with a destination node in sub-network B, the path selection algorithm should give the optimal path with minimum cost. This path can be divided into two sub-paths. One is the sub-path from the source node to the an end of a bridge in sub-network B. The other is the sub-path from the end node of a bridge in sub-network B to the destination node. Note that there can be multiple bridges at the same time, which means there can be multiple combinations of two sub-paths, the optimal path should be the one that minimize the overall cost of these two sub-paths.

An inter quantum network problem differs from intra quantum network routing problem mainly because of adding inter-network requests. How to coordinate intra- and inter- network requests is a problem that has not been explored before. We give detailed discussion on the topic later in section 4.2.1 and 5.3.3.

Here we assume the routing problem is handled by one centralized network controller and one centralized request collector that collects requests from both sub-networks. They can also be implemented in a decentralized manner, as we will discuss later in section 4.2.2.

We give now the description of inter quantum network routing problem based on the system model.

Input of the inter quantum network routing problem is the following:

1. A graph $G_1 = (V_1, E_1)$ where V_1 contains all vertices in sub-network A and E_1 contains all edges and their corresponding physical distances between any two nodes in logical topology of network A.
2. $G_2 = (V_2, E_2)$, recording same information in sub-network B as G_1 does for the sub-network A.
3. A set S_1 consisting of successful entanglements from the preparation phase of sub-network A; a corresponding reduced graph $G_1' = (V_1, E_1')$, where E_1' contains all elements in S_1.
4. S_2 and G_2' containing same information as S_1 and G_1' but now for sub-network B
5. E_3 containing all the bridges that connect sub-network A and sub-network B.

6. A list of pairs P of end nodes from both request collectors R_1 and R_2 and the request list from previous time slots. A pair of end nodes contains a source node and a destination node that request building connections in the whole network.

The routing algorithm is expected to give an ordered set of source-destination paths as the output. Each path should contain the following relevant information [10].

1. A list of intermediate nodes of the path; and for each intermediate node, the quantum memory slots, in which the qubits to be measured are located.
2. The quantum memory slots, in which the qubits lie that needs to be entangled, one on the source node and another on the destination node.
3. A path ID, unique in a time slot, to differentiate this path from others.
4. The node ID of the destination, to which the intermediate nodes send correction messages.

Let us assume that inter-requests and intra-requests are considered equally important Then the inter quantum network routing problem can be viewed as an intra quantum network routing problem because one can view the two smaller networks as one bigger network and routes are found inside the bigger network. Therefore, maximizing the entanglement generation rate in the system is the main goal of the network controller [10].

The remainder of the section contains request scheduling and possible implementations of network controller and request collector.

4.2.1 Request Scheduling

The purpose of request scheduling is to decide which requests to serve and in which order [10]. A possible realization of scheduling is called skipping [10]. The skipping mechanism indicates that a request can be skipped when it has not been skipped before. However, if a request has been skipped once, it cannot be skipped in the future. We call a request skippable when it has not been skipped before and a request is unskippable when it has been skipped once. The working flow, in the form of instructions, of the network controller based on scheduling is as follows [10].

1. At the beginning of each time slot, make a temporary waiting queue by copying from the system's waiting queue.

2. Append new requests to the end of the temporary waiting queue.

3. Read the request from the temporary queue until the time slot ends or the temporary queue is empty.

   3.1 If the request is skippable and the request is instructed to be skipped, skip the request and mark it unskippable in system's waiting queue, go to step 3.

   3.2 If the request is not skippable or the request is instructed not to be skipped :

      3.2.1 If the request can be served with a path, serve the request, and update the reduced graph by removing edges on the path from the graph, delete the request from system's waiting queue, go to step 3.

      3.2.2 If the request cannot be served because no path can found from the source node to the destination in the current reduced graph, go to step 3.

4. When the current time slot ends, empty the temporary waiting queue and go to step 1.

In steps 3.1 and 3.2 above, there needs to be some request selection guidance to decide if a request should be served. If a request is served, the controller will try to find a path in the current reduced graph. Depending on the preference for request types, we propose five different request scheduling policies for comparison.

1. Inter-first: it will always serve inter-network requests and always skip an intra-network request if skippable.

2. Inter-favored: it will always serve inter-network requests and skip an intra-network request if skippable with a 50% probability

3. None: it will favor no specific requests, i.e., no request is skipped.

4. Intra-favored: it will always serve intra-network requests and skip an inter-network request if skippable with a 50% probability

5. Intra-first: it will always serve intra-network requests and always skip an inter-network request if skippable.

These policies are proposed to find out how request scheduling policy will change network behavior by showing preference for some types of requests.

4.2.2 Implementation Aspects

An autonomous quantum network should have its own request collector and network controller. However, if two or more of networks are to communicate with each other, coordination among multiple request collectors and network controllers should be done to reach consensus, e.g., timing synchronization, among them.

This section discusses two ways of implementing network controller and request collector from the angle of path selection and request scheduling.

From the perspective of path selection, many previous implementations can be applied here as the topic has been widely studied [8, 34, 10]. The common path selection algorithm has non-negligible growth in time complexity with network size. Considering the size of quantum network is still very small at the time of writing, using a centralized network controller will not necessarily deteriorate network performance. However, if each sub-network has one network controller, then this controller is able to freely choose path selection algorithm based on its sub-network configuration and this controller is capable of hiding intra-network details from outsiders. More importantly, such arrangement of network controllers and request collectors can speed up the path selection algorithm if network controllers are running in parallel.

In terms of request scheduling, implementation of decentralized network controllers will introduce the problem of reaching consensus. As shown in [10], the order of processing requests significantly affects the distribution of network resources. In comparison, it is easier to implement a centralized network controller processing requests from a centralized request collector because such implementation avoids the consideration of difficult scenarios to reach consensus.

5 Experiments

In this chapter, we explain the design details of experiments.

5.1 Simulator and Experiment Environment

We run all our experiments on NetSquid [31], a quantum network simulation platform combined with a discrete event simulator. It is capable of representing accurately underlying quantum hardware with corresponding quantum properties, e.g., noises, fibre losses and channel delay. Some experiments reported in earlier works, e.g., in [10, 11, 23], are also run on this simulator.

All results from simulations are obtained with Python 3.8.3 on MacOS Monterey. All experiments run on a 2020 MacBook Pro with 2 GHz Quad-Core Intel Core i5 Processor and 16 GB 3733 MHz LPDDR4X Memory.

5.2 Model assumptions

If we zoom in the structure of a quantum network in the model, we will find the key composition is two nodes connected by both a quantum channel and a classical channel. There is also a quantum source in between the two nodes. Details of the composition are illustrated in Figure 5.1.

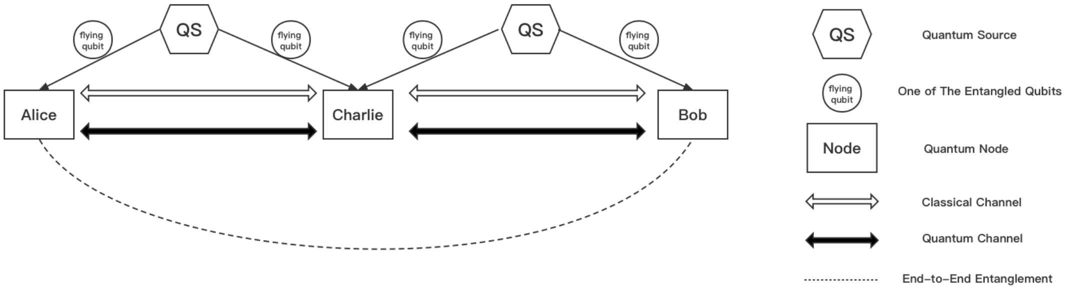

Figure 5.1: The key composition in the model. Adapted from [10].

Features of the key composition are the following:

1. A quantum source is located in the middle of two nodes to distribute an entangled pair of qubits at the beginning of each time slot.
2. A qubit may be lost in a quantum channel with a certain probability.
3. A classical channel imposes delay but not error to the delivery of classical messages.

Detailed introduction to error models and delay models can be found in the API description pages on NetSquid website [32]. In our system model, we are using “FibreLoss” model in quantum channels, “DephaseNoise” model in quantum memory, “DepolarNoise” model in quantum gates and “FibreDelay” model in classical channels.

A quantum channel modelled via fibre loss will cause the flying qubit to get lost in an optic fibre channel probabilistically. The probability p_{loss} is related to the initial loss probability p_{init} and survival probability of a qubit per channel length $p_{loss-per-length}$ [32]. In the model, $p_{init} = 0.1$, and $p_{loss-per-length} = 0.1$.

A quantum memory modelled via dephasing noise will put its stored qubits on decoherence by imposing dephasing noise on them. Decoherence is the main reason for the drop of the fidelity of quantum state [23]. The dephasing probability grows exponentially with dephasing rate and existence time of qubits [32]. In the model, we set dephasing rate as 1 MHz.

An implementation of quantum gate modelled via a depolarizing noise introduces different kind of noise to qubits. It happens in the operation of measurement on qubits and making a qubit correction. Like dephasing probability, depolarizing probability grows exponentially with depolarizing rate and the gate operation time. In the model, a gate operation time is 10 ns, and the adopted depolarizing rate is 5 KHz.

A classical channel modelled via a timing delay only transmits classical messages without error but with delay. The delay is related to channel length and speed of light. In the model, we set light speed 200000 km/s.

Therefore, from the perspective of qubits, an end-to-end entanglement in the model can not be established if one of the following happens:

1. If any of the qubits in the repeater chain fails to arrive from quantum source to the correct neighboring node in the first place.
2. If the fidelity of the final entanglement fails to meet the system requirement. Noise has two possible sources according to our assumptions. One is depolarizing noise by

imperfect implementation of quantum gates, which happens when measuring qubits or performing qubit correction. The other is dephasing noise that causes qubits to decohere over time in quantum memory.

The above error models, delay model and their corresponding parameters are the same as the ones used in the experiments of [10] in order to explore the probability of inter-connecting two quantum networks in the state-of-the-art technology.

5.3 Implementation Details

In this section, we discuss implementation details. More specifically, we discuss topology and configurations in our experiments.

5.3.1 Topology

We are using lattice topology in our experiments as it is easy to implement in the early stage of quantum network development and it might be one of the potential implementations in the near future if someone tries to cover an area using multiple nodes in an evenly distributed manner [10].

We are fully connecting two sub-networks of same size as shown in Figure 5.2. The reason for full connection rather than partial connection is that by doing so, we will have a reasonable estimate of the best performance of the system by assigning to it as many resources as possible.

5.3.2 Configurations

For each set of simulations, we have three configuration variables: size of sub-network, length of inter-network distance and offered requests. Size of sub-network falls in the range of 3 to 5 (both ends included); length of inter-network distance varies between 1.8 km to 7.2 km with increment of 0.6 km; and offered requests are one of the entries in the Table 5.1. The distribution of requests is one of the four types below, depending on the location of the source node and the destination node.

1. A -> A: Both are in sub-network A.

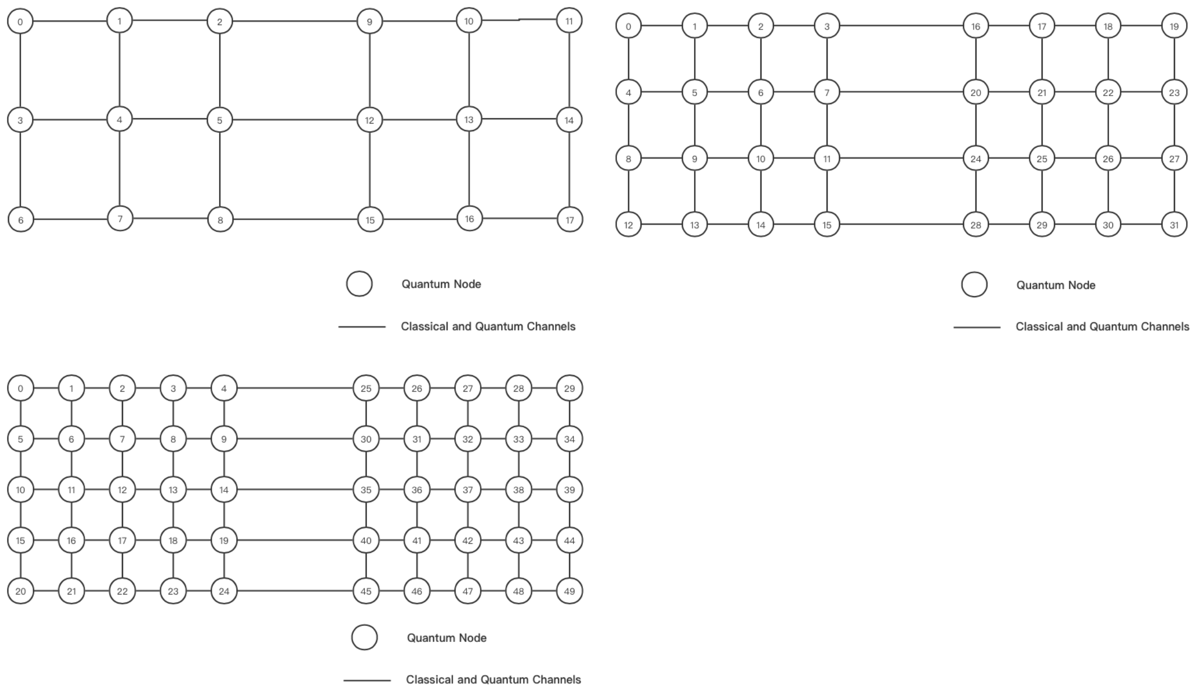

Figure 5.2: A network that fully connects two 3*3, 4*4, 5*5 sub-networks, respectively.

2. B -> B: Both are in sub-network B.
3. A -> B: Source node in sub-network A and destination node in sub-network B.
4. B -> A: Source node in sub-network B and destination node in sub-network A.

In total, there are 270 sets of simulation runs with different configurations and for each of them, we simulated 10000 time slots.

We notice that in the regular physical grid experiment from [10], a 5*5 network is used to cover $7.2 * 7.2km^2$ area, so in our experiments, we are setting the following question: given two sub-networks of regular grid covering two $7.2 * 7.2km^2$ area, what is the longest feasible inter-network distance? In terms of request scheduling policy, a good question to ask is that given the same inter-network distance and same size of sub-network, what is the best request scheduling policy according to entanglement generation rate.

As for request sampling, in each time slot, we have four types of requests to be generated as discussed before. For any request type, the generated requests in a time slot are randomly chosen from a complete set of requests of that type. Considering that the number of short-distance requests is larger than that of long-distance requests, e.g., in a 4*4 lattice whose inner-node distances are all one unit, the number of requests of length one unit

Number of requests	Distribution of requests			
	A ->A	B ->B	A ->B	B ->A
4	1	1	1	1
6	2	2	1	1
6	1	1	2	2
8	3	3	1	1
8	1	1	3	3
10	4	4	1	1
10	3	3	2	2
10	2	2	3	3
10	1	1	4	4

Table 5.1: Number and distribution of requests.

are much larger than the number of requests of length six units. Hereby, we ensure, for all possible requests within the complete set, they have the same uniform probability of getting selected.

Now we discuss all possible results of an individual request in the system.

1. It is served eventually after being skipped.
2. It is served without being skipped.
3. It is dropped because the request has been waiting too long in the queue. The maximum waiting time is 10 time slots in the model.
4. It is not served because the simulation ends before it has the chance to get served.

It can be derived that the maximum number of unserved requests in the model is 10 times the number of offered requests in each time slot.

For a served request, we define the end-to-end entanglement establishment success if the final fidelity $F \geq 0.5$, otherwise it is a failure. This is because an entanglement is not usable if $F < 0.5$ [23].

5.3.3 Metrics

There are two types of end-to-end entanglements for each successful request. An inter-network request will build an end-to-end entanglement path between two networks and an intra-network request will establish an entanglement path inside one network.

For the brevity of expressions, we will use the term inter-request for inter-network request and the term intra-request for intra-network request in the remainder of the thesis. A successful inter-request is called an inter-success. Similarly, an intra-success stands for a successful establishment of an intra-network end-to-end path based on an intra-request.

Normally, entanglement generation rate is used as a metric to measure the quantum network performance. We use the same metric called synthetic entanglement rate, or abbreviated as SER in the experiment as well. It is noticeable that SER treats inter-requests and intra-requests as equivalence according to our assumptions.

Note that in our system, a success can be classified into two categories: an inter-success or an intra-success. A reasonable new metric for evaluation should consider simultaneously both inter-success rate and intra-success rate. Now that there is no standard metric for assessment, one can simply come up with a weighted average metric. The corresponding formula should be $R_{entanglement} = w_{inter-request} * R_{inter-success} + w_{intra-request} * R_{intra-success}$, where $R_{entanglement}$ is the proposed metric for evaluation, $w_{inter-request}$ and $w_{intra-request}$ are the weights of inter-request and intra-request in evaluation, respectively; $R_{inter-success}$ and $R_{intra-success}$ are inter-success rate and intra-success rate. Considering no related research has been conducted in this area, it is hard to determine $w_{inter-request}$ and $w_{intra-request}$ without any insight, we will leave this work as a future research topic.

6 Performance Evaluation

This chapter gives performance evaluation of various network systems realized in the model and discusses influential factors in network performance.

6.1 Overview

Figure 6.1 presents the relation between SER and inter-network distance when four new requests are offered in each time slot and no request policy is specified. It shows the best network performance under the light-load mode, as will be introduced in later in 6.1.1. For communication between two grids, entanglement rate drops with inter-network distance for all three types of grid. More specifically, communication between two 5*5 lattices gives the largest reduction of SER, followed by 4*4 lattices and finally 3*3 lattices. When inter-network distance is 1.8 km, all three types of grid produce SER slightly over 0.5 with 5*5 coming first, 4*4 second and 3*3 last. However, when inter-network distance reaches 2.4 km, 4*4 grid gives the best performance while 5*5 grid gives the worst. Situation changes after inter-network distance hits 3.0 km, the network performance is dominated by inter-network distance instead of inner-node distance. Grid size of 3 gives the best performance afterwards, followed by grid size of 4 and 5.

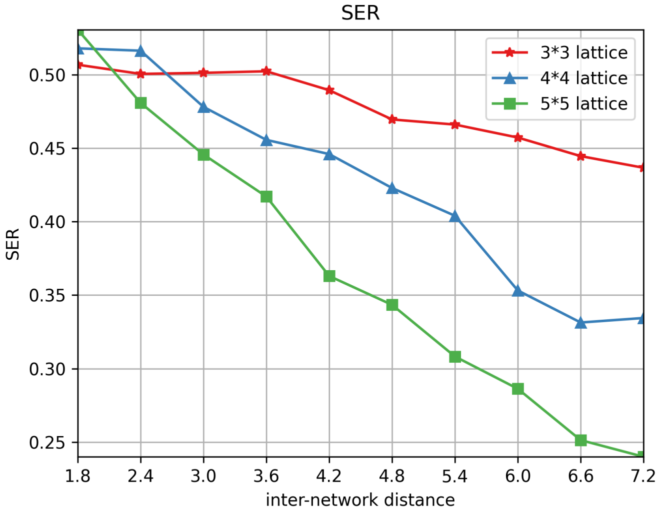

Figure 6.1: Request policy: none; number of requests: 4.

Limited by current quantum hardware, the network performance might be disappointing with the best SER being slightly over 0.5 as shown in the Figure 6.1, quantum networking is still in its infant stage so the state-of-art technology cannot guarantee high performance quantum hardware.

Even though the state-of-art hardware performance has its limitations, we believe the study of influential factors in inter quantum network routing under current technology will provide valuable experience and knowledge.

In the following, we will study the impact of different configuration variables on network system, namely number and distribution of requests, inter-network distance, scheduling policy and inter-network distance.

6.1.1 Number and Distribution of Requests

Figure 6.2 gives grouped data bar charts of both request unserved rate and SER with request distribution. For each request distribution, five bars are labeled to represent corresponding request unserved rate and SER under different scheduling policies. Note that request unserved rate is a negative index of network performance while SER is a positive one. The bigger the request unserved rate, the worse is the network performance because more requests are not served.

As seen from the upper plot of Figure 6.2, the number of requests has significant impact on requests unserved rate. More specifically, as suggested by [10], the system is indeed slightly loaded when the number of requests is four in a time slot. The system gradually becomes “heavy” as more requests get unserved.

Another observation from Figure 6.2 is that the distribution of requests influence both entanglement rate and unservablity of the system.

When given the same number of requests, offering more inter-requests in a time slot will drastically downgrade network performance, in terms of SER and request unserved rate. The statement holds when the number of requests is 6, 8 or 10.

More surprisingly, in the bottom plot of Figure 6.2, the bar heights of all policies of 8(1133) are very close to the ones of 10(2233), higher than both 10(4411) and (3322), which indicates that inter-requests are more demanding and add more pressure to the system than intra-requests.

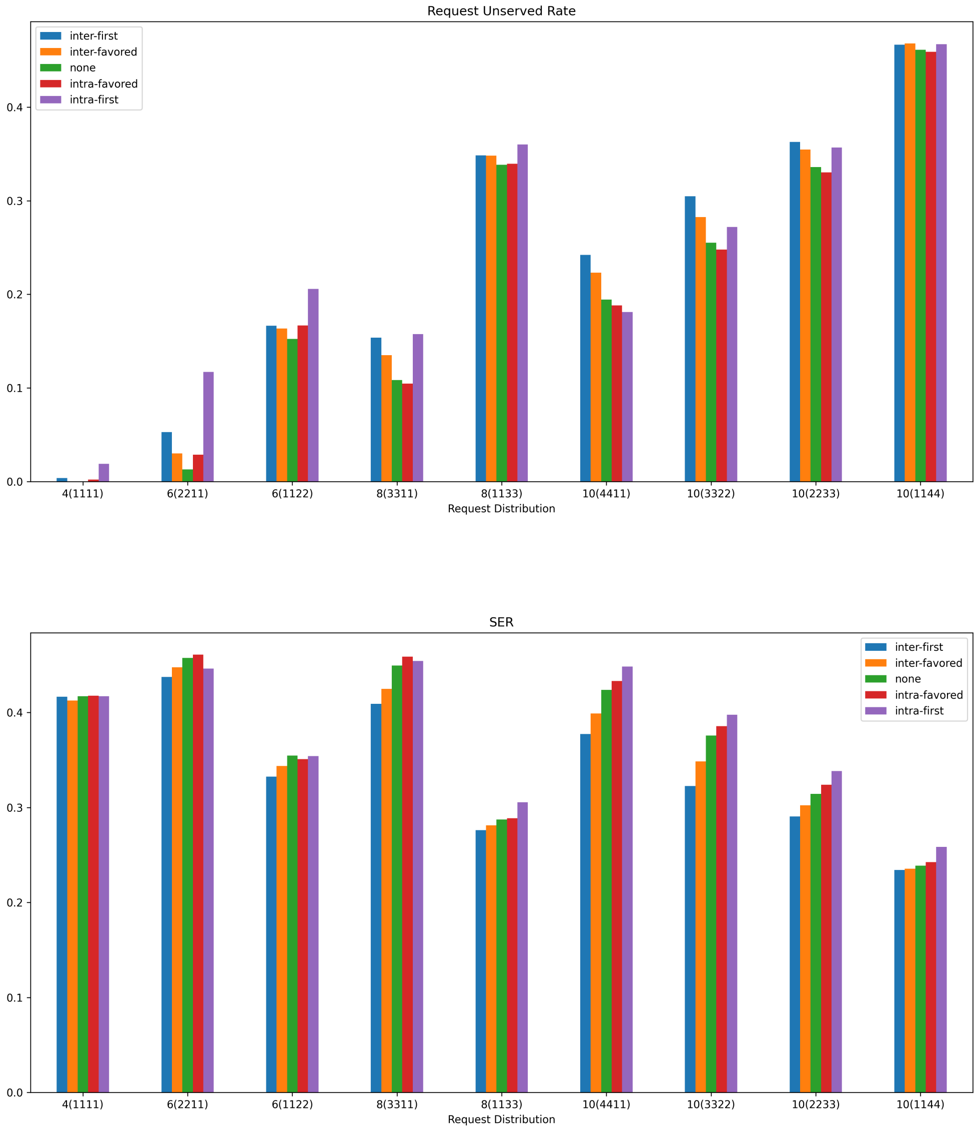

Figure 6.2: Grid size: 5, inter-network distance: 3.6.

6.1.2 Scheduling Policy

Figure 6.3 gives the inter-success rate and intra-success rate, respectively, with inter-network distance given grid size of 5 and request type 10(3322). In the left plot, SER curves down smoothly before inter-network distance reaches 4.8 km, and after this point, SER steps down instead. Similar step-wise downfall can be seen from the right plot. Before inter-network distance hits 3.6 km, minor changes are seen with all five scheduling policies. After 3.6 km, more pronounced decline is shown in terms of intra-success rate.

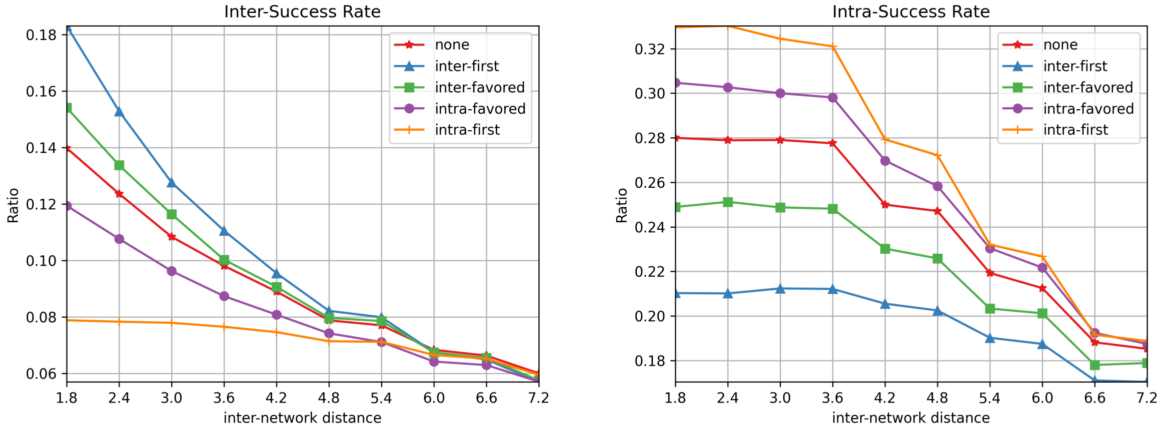

Figure 6.3: Grid size: 5, request type: 10(3322)

From the left plot of Figure 6.3, it can be seen that, for most of inter-network distance, inter-first policy results in highest inter-success rate, followed by inter-favored policy, none policy, intra-favored policy and finally intra-first policy.

Similarly, from the right plot of Figure 6.3, it can be seen that, intra-first policy gives best intra-success rate and inter-first policy gives worst performance in intra-success rate, with the curves of intra-favored, none, and inter-favored policies in between.

From the above observations, we can tell that scheduling policy will change system performance by instructing controller to favor some requests over others.

Now that we have agreed to treat intra-request and inter-request equivalently in evaluation, it is easy to understand that in terms of SER, intra-first policy in general gives the best performance in all inter-network distances. The reasons are the following.

1. Intra-first policy will maximize intra-success rate and minimize inter-success rate.
2. Intra-success rate is inherently higher than inter-success rate and the gain on SER from serving more intra-requests outweighs the loss from not serving more inter-requests.

From another angle, as pointed out in [10], the system model prefers short-distance requests over longer ones. Generally, intra-requests require shorter paths than inter-requests. Another reason that intra-requests are preferred compared to inter-requests can be that they generally require less entanglements while intra-requests need both bridges and local entanglements from both sub-networks.

6.1.3 Inter-Network Distance

Figure 6.4 gives the progression of SER with inter-network distance. For inter-network distance from 2.4 to 7.2, intra-first policy gives the best SER followed by intra-favored policy, none, inter-favored policy and inter-first policy. The patterns of all five curves in Figure 6.4, from another perspective, are the combinations of the curves of inter-success rate and intra-success rate in Figure 6.3.

It is clear, as seen from Figure 6.4 below, that the SER curves of all policies go down roughly linearly with the inter-network distance, which resembles the pattern in Figure 6.1. However, as seen from Figure 6.3, inter-network distance affects inter-success rate and intra-success rate differently.

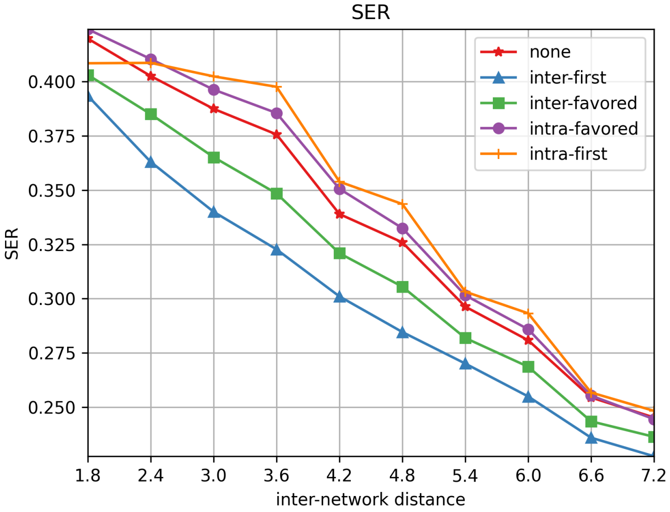

Figure 6.4: Grid size: 5, request type: 10(3322)

It is clear that increasing inter-network distance will have significant impact on inter-success rate, and average fidelity of inter-requests. However, we notice that increasing inter-network distance also has interesting influence on intra-requests. Compared to the “waterfall” pattern in the right of Figure 6.3, the curve in the left goes down smoothly.

Similar phenomenon can also be seen when grid size is 4*4, as seen in Figure 6.5 below.

Figure 6.5 gives three plots of SER, inter-success rate and intra-success rate, repectively given grid size of 4 and the same request type 10(3322). The patterns of decline in success rate of inter-requests and intra-requests are similar to those in Figure 6.3. More specifically, the curve of dropping in inter-success rate in Figure 6.5 is similar to the dropping pattern in the left plot of Figure 6.3. And the drastic reduction of intra-success rate from 3.6 km to 4.2 km in the right plot of Figure 6.3 can also be discovered from 5.4 km to 6.0 km in the third plot of Figure 6.5.

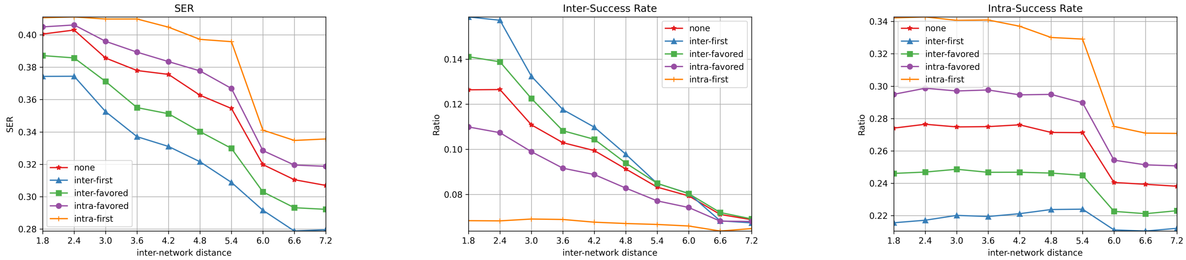

Figure 6.5: Grid size: 4, request type: 10(3322)

We hereby propose possible explanations for such phenomenon below.

Inter-network Distance and Inter-Requests

First, we discuss the impact of inter-network distance on inter-requests.

Figure 6.6 gives more information of what happens in the network as the inter-network distance increases. The top left plot gives the average number of hops of inter-success, while the bottom left gives the average number of hops of intra-success. The top right one shows the average fidelity of inter-success while the bottom right shows the average fidelity of intra-success.

As seen from Figure 6.6, when inter-network distance increases, the average number of hops of inter-requests declines. That is understandable because longer inter-node distance means bridges will break with higher probability, and the original long-distance inter-requests will likely be unservable and be replaced by shorter ones. Serving shorter inter-requests as inter-network distance increases also means the uptrend of average fidelity of inter-requests. The uptrend comes mainly from the decrease of servable inter-requests.

Note that if increasing inter-network distance infinitely, average number of hops will continue to drop to zero. As for the curve of average fidelity of inter-success, it will gradually become the curve of average fidelity of bridges.

Inter-Network Distance and Intra-Requests

Next, we discuss the impact of inter-network distance on intra-requests.

1. If the length of inter-network distance is so large that it is impossible to establish any

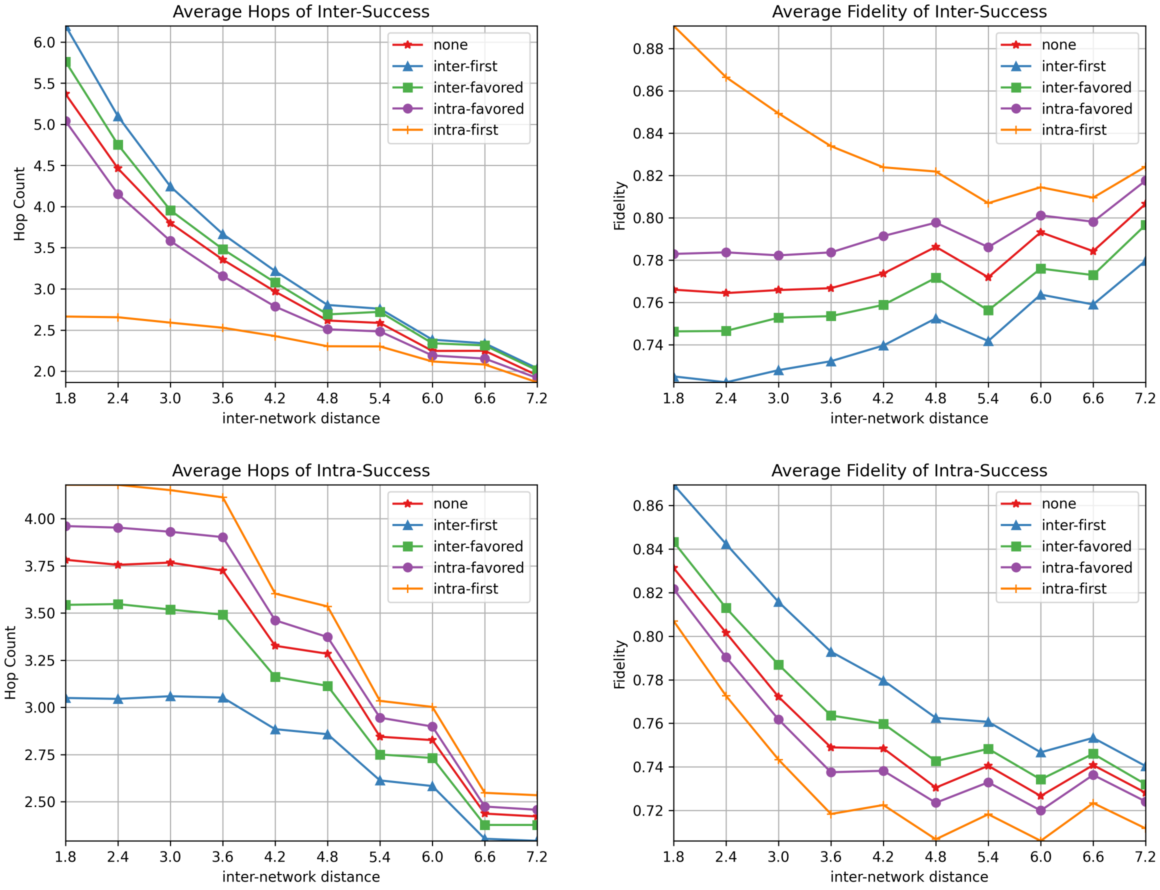

Figure 6.6: Grid size: 5, request type: 10(3322)

local entanglements between two sub-networks, in such case, the network degrades to two separate networks and all inter-network requests are unservable. Each sub-network will only serve intra-network requests. The topic of serving intra-network requests has been widely investigated before, e.g., in [10]. This scenario has no presence in the implication of the figures above, so we do not go into the topic any further.

2. Before the inter-network distance hits the point after which all bridges are always down in the start of a time slot, the contention for intra-network resources between intra-network requests and inter-network ones will become more intense. It is obvious that an inter-network request consumes some intra-network resources from both subnetworks. When inter-network distance becomes larger in this case, the expected number of bridges will gradually go down inevitably, as inferred from the top left plot of Figure 6.6. It is unclear if this is the reason for the step-wise downfall of average hop count of intra-success. However, the fact that the expected number of

bridges will gradually go down with inter-network distance will potentially cause the controller to find a longer path in which a bridge exists because the bridge in the shorter path breaks down. Therefore, more local entanglements in both subnetworks will be taken up by inter-network requests, which results in the preference for shorter intra-network requests, as shown in the bottom left plot of Figure 6.6.

The reason behind the "bumping" phenomenon cannot be fully explained at the time of writing. It can be a consequence of the underlying quantum mechanics working or caused by some unknown features of the model. Therefore, more in-depth research is needed to gain better insight into the phenomenon.

6.2 Discussion

This section presents the influential factors in network performance and discusses possible solutions to mitigate the effect of the factors.

1. Inter-network distance: Essentially, this is part of the length of the repeater chain. Longer inter-network distance increases the probability of qubit loss in our model, which indicates smaller expected number of bridges. It is a neutral factor. To mitigate the effect of it, one can either adopt better equipment, e.g., better optical fibre with smaller probability of photon loss, or use multiple repeaters in the middle to shorten the length.

2. Attribute of the time-slotted model: In the model, requests in the front of the queue will get served with higher probability of success. The main reason is the function of decoherence to qubits while they are idling in the quantum memory when controller is performing path selection. Front requests have shorter waiting time while rear ones wait longer. To address decoherence, researchers are expected to improve coherence time in quantum memory. Another solution is to reduce the execution time of the routing algorithm, e.g., by using high performance computer.

3. Network load: As discussed before, the number and distribution of new requests have non-negligible influence on network performance. Unservability severely deteriorates user experience. The unserved requests come from two parts: 1. the remaining requests from previous time slots; 2. the unserved requests from this time slot. A feasible solution to this issue is to experiment beforehand to find the suitable network

load if the new offered load is fixed in each time slot, i.e., to find the maximum feasible fixed load of each time slot for a system before the actual running of it. Another solution is to adopt unfixed network load to limit the new incoming network load when the system is heavily loaded in a time slot.

4. The skipping mechanism of scheduling: In the current system, front requests have highest priority of getting served to avoid starvation. However, such arrangement might somehow counteract the effect of scheduling policy because the remaining requests in the queue affect the future decisions on new requests. Future design can consider optimizing the prioritization mechanism. One possible solution is to serve the previous requests periodically in exclusive time slots so that starvation can be prevented, and the scheduling policy will function normally in other time slots. Another possibility is to utilize a priority queue model in which the most profitable requests will get served in the current time slot. The priority of a request can be provided by a function, which considers multiple factors, e.g., waiting time and scheduling policy preference.
5. Scheduling policy: The current scheduling policy is designed to favor certain type of requests over others. However, the granularity might be too coarse if one only considers the request type. Possible scheduling policy in the future can take into account more specific indices, e.g., path length threshold.

7 Conclusions

This thesis has attempted to inter-connect two quantum networks, by extending a time-slotted model from predecessors' work. Following the idea of splitting quantum routing problem into path selection and request scheduling, we put focus on the latter. Five scheduling policies are proposed to explore the impact of scheduling policies on inter quantum network routing problem. Generally, our scheduling policy changes network behavior and performance by favoring certain requests over others. We also analyzed how inter-network distance and the number and distribution of requests affect network performance with experimentation. Summary of influential factors in network performance in the current model has also been presented.

For future work, one can consider the following: more topology (e.g., random topology and hexagon) for a sub-network; partial connections between two sub-networks; separated implementations of request collector and network controller, e.g., in a decentralized manner; different scheduling policies or scheduling mechanisms, e.g., the ones mentioned above in Chapter 5.

Bibliography

[1] Y. Aharonov, J. Oppenheim, S. Popescu, B. Reznik, and W. G. Unruh. "Measurement of time of arrival in quantum mechanics". In: *Physical Review A* 57.6 (1998), pp. 4130–4139. DOI: `10.1103/PhysRevA.57.4130`. URL: `https://link.aps.org/doi/10.1103/PhysRevA.57.4130`.

[2] D. Alanis, P. Botsinis, Z. Babar, S. X. Ng, and L. Hanzo. "Non-Dominated Quantum Iterative Routing Optimization for Wireless Multihop Networks". In: *IEEE Access* 3 (2015), pp. 1704–1728. ISSN: 2169-3536. DOI: `10.1109/access.2015.2478793`.

[3] D. Alanis, P. Botsinis, S. X. Ng, and L. Hanzo. "Quantum-Assisted Routing Optimization for Self-Organizing Networks". In: *IEEE Access* 2 (2014), pp. 614–632. ISSN: 2169-3536. DOI: `10.1109/access.2014.2327596`.

[4] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, C. Gidney, M. Giustina, R. Graff, K. Guerin, S. Habegger, M. P. Harrigan, M. J. Hartmann, A. Ho, M. Hoffmann, T. Huang, T. S. Humble, S. V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, J. Kelly, P. V. Klimov, S. Knysh, A. Korotkov, F. Kostritsa, D. Landhuis, M. Lindmark, E. Lucero, D. Lyakh, S. Mandrà, J. R. McClean, M. McEwen, A. Megrant, X. Mi, K. Michielsen, M. Mohseni, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Y. Niu, E. Ostby, A. Petukhov, J. C. Platt, C. Quintana, E. G. Rieffel, P. Roushan, N. C. Rubin, D. Sank, K. J. Satzinger, V. Smelyanskiy, K. J. Sung, M. D. Trevithick, A. Vainsencher, B. Villalonga, T. White, Z. J. Yao, P. Yeh, A. Zalcman, H. Neven, and J. M. Martinis. "Quantum supremacy using a programmable superconducting processor". In: *Nature* 574.7779 (2019), pp. 505–510. ISSN: 1476-4687. DOI: `10.1038/s41586-019-1666-5`. URL: `https://doi.org/10.1038/s41586-019-1666-5`.

[5] C. H. Bennett and G. Brassard. "Quantum cryptography: Public key distribution and coin tossing". In: *Theor. Comput. Sci.* 560 (2014), pp. 7–11.

[6] A. Broadbent, J. Fitzsimons, and E. Kashefi. "Universal Blind Quantum Computation". In: *2009 50th Annual IEEE Symposium on Foundations of Computer Science*

(Oct. 2009). DOI: 10.1109/focs.2009.36. URL: http://dx.doi.org/10.1109/FOCS.2009.36.

[7] A. S. Cacciapuoti, M. Caleffi, F. Tafuri, F. S. Cataliotti, S. Gherardini, and G. Bianchi. "Quantum Internet: Networking Challenges in Distributed Quantum Computing". In: *IEEE Network* 34 (2020), pp. 137–143.

[8] M. Caleffi. "Optimal Routing for Quantum Networks". In: *IEEE Access* 5 (2017), pp. 22299–22312.

[9] J. Chow, O. Dial, and J. Gambetta. *IBM Unveils Breakthrough 127-Qubit Quantum Processor*. Nov. 2021. URL: https://research.ibm.com/blog/127-qubit-quantum-processor-eagle.

[10] C. Cicconetti, M. Conti, and A. Passarella. "Request Scheduling in Quantum Networks". In: *IEEE Transactions on Quantum Engineering* 2 (2021), pp. 2–17.

[11] A. Dahlberg, M. Skrzypczyk, T. Coopmans, L. Wubben, F. Rozq dek, M. Pompili, A. Stolk, P. PawfBczak, R. Knegjens, J. A. d. O. Filho, R. Hanson, and S. Wehner. "A link layer protocol for quantum networks". In: *Proceedings of the ACM Special Interest Group on Data Communication* (2019).

[12] W. Dai, T. Peng, and M. Z. Win. "Quantum Queuing Delay". In: *IEEE Journal on Selected Areas in Communications* 38 (2020), pp. 605–618.

[13] R. Demkowicz-Dobrzański and L. Maccone. "Using Entanglement Against Noise in Quantum Metrology". In: *Physical Review Letters* 113.25 (2014), p. 250801. DOI: 10.1103/PhysRevLett.113.250801. URL: https://link.aps.org/doi/10.1103/PhysRevLett.113.250801.

[14] P. A. M. Dirac. "A new notation for quantum mechanics". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 35.3 (1939), pp. 416–418. DOI: 10.1017/S0305004100021162.

[15] M. Dorigo, V. Maniezzo, and A. Colorni. "Ant system: optimization by a colony of cooperating agents". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 26.1 (1996), pp. 29–41. DOI: 10.1109/3477.484436.

[16] V. Giovannetti, S. Lloyd, and L. Maccone. "Quantum-enhanced positioning and clock synchronization". In: *Nature* 412.6845 (2001), pp. 417–419. ISSN: 1476-4687. DOI: 10.1038/35086525. URL: https://doi.org/10.1038/35086525.

[17] Y. Guo and G. Gour. "Monogamy of the entanglement of formation". In: *Phys. Rev. A* 99 (4 Apr. 2019), p. 042305. DOI: 10.1103/PhysRevA.99.042305. URL: https://link.aps.org/doi/10.1103/PhysRevA.99.042305.

[18] B. Hensen, H. Bernien, A. E. Dréau, A. Reiserer, N. Kalb, M. S. Blok, J. Ruitenberg, R. F. L. Vermeulen, R. N. Schouten, C. Abellán, W. Amaya, V. Pruneri, M. W. Mitchell, M. Markham, D. J. Twitchen, D. Elkouss, S. Wehner, T. H. Taminiau, and R. Hanson. "Loophole-free Bell inequality violation using electron spins separated by 1.3 kilometres". In: *Nature* 526.7575 (2015), pp. 682–686. ISSN: 1476-4687. DOI: 10.1038/nature15759. URL: https://doi.org/10.1038/nature15759.

[19] X.-B. Huang, Y.-H. Chen, and Z. Wang. "Fast generation of three-qubit Greenberger-Horne-Zeilinger state based on the Lewis-Riesenfeld invariants in coupled cavities". In: *Scientific Reports* 6.1 (2016), p. 25707. ISSN: 2045-2322. DOI: 10.1038/srep25707. URL: https://doi.org/10.1038/srep25707.

[20] Z. Huang, C. Macchiavello, and L. Maccone. "Usefulness of entanglement-assisted quantum metrology". In: *Physical Review A* 94.1 (2016), p. 012101. DOI: 10.1103/PhysRevA.94.012101. URL: https://link.aps.org/doi/10.1103/PhysRevA.94.012101.

[21] P. C. Humphreys, N. Kalb, J. P. J. Morits, R. N. Schouten, R. F. L. Vermeulen, D. J. Twitchen, M. L. Markham, and R. Hanson. "Deterministic delivery of remote entanglement on a quantum network". In: *Nature* 558 (2018), pp. 268–273.

[22] P. Kómár, E. M. Kessler, M. Bishof, L. Jiang, A. S. Sørensen, J. Ye, and M. D. Lukin. "A quantum network of clocks". In: *Nature Physics* 10.8 (2014), pp. 582–587. ISSN: 1745-2481. DOI: 10.1038/nphys3000. URL: https://doi.org/10.1038/nphys3000.

[23] W. Kozlowski, A. Dahlberg, and S. Wehner. "Designing a quantum network protocol". In: *Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies* (2020).

[24] W. Kozlowski and S. Wehner. "Towards Large-Scale Quantum Networks". In: *Proceedings of the Sixth Annual ACM International Conference on Nanoscale Computing and Communication*. NANOCOM '19. Dublin, Ireland: Association for Computing Machinery, 2019. ISBN: 9781450368971. DOI: 10.1145/3345312.3345497. URL: https://doi.org/10.1145/3345312.3345497.

[25] V. Krutyanskiy, M. Meraner, J. Schupp, V. Krcmarsky, H. Hainzer, and B. P. Lanyon. “Light-matter entanglement over 50km of optical fibre”. In: *npj Quantum Information* 5.1 (2019), p. 72. ISSN: 2056-6387. DOI: 10.1038/s41534-019-0186-3. URL: https://doi.org/10.1038/s41534-019-0186-3.

[26] S. Lang. “The Tensor Product”. In: *Algebra*. New York, NY: Springer New York, 2002, pp. 601–640. ISBN: 978-1-4613-0041-0. DOI: 10.1007/978-1-4613-0041-0_16. URL: https://doi.org/10.1007/978-1-4613-0041-0_16.

[27] Y.-H. Lee, E. Bersin, A. Dahlberg, S. Wehner, and D. R. Englund. “A Quantum Router Architecture for High-Fidelity Entanglement Flows in Multi-User Quantum Networks”. In: *arXiv: Quantum Physics* (2020).

[28] C. Li, T. Li, Y.-X. Liu, and P. Cappellaro. “Effective routing design for remote entanglement generation on quantum networks”. In: *npj Quantum Information* 7 (2020), pp. 1–12.

[29] G.-Q. Liu and X.-Y. Pan. “Quantum information processing with nitrogen–vacancy centers in diamond”. In: *Chinese Physics B* 27.2 (2018), p. 020304. ISSN: 1674-1056. DOI: 10.1088/1674-1056/27/2/020304. URL: http://dx.doi.org/10.1088/1674-1056/27/2/020304.

[30] H.-Y. Liu, X.-H. Tian, C. Gu, P. Fan, X. Ni, R. Yang, J.-N. Zhang, M. Hu, J. Guo, X. Cao, X. Hu, G. Zhao, Y.-Q. Lu, Y.-X. Gong, Z. Xie, and S.-N. Zhu. “Drone-based entanglement distribution towards mobile quantum networks”. In: *National Science Review* 7.5 (2020), pp. 921–928. ISSN: 2095-5138. DOI: 10.1093/nsr/nwz227. URL: https://doi.org/10.1093/nsr/nwz227.

[31] *NetSquid*. https://netsquid.org. Accessed: 2022-05-25.

[32] *Netsquid.Components.Models*. https://docs.netsquid.org/latest-release/netsquid.components.models.html. Accessed: 2022-05-25.

[33] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. DOI: 10.1017/CBO9780511976667.

[34] M. Pant, H. Krovi, D. F. Towsley, L. Tassiulas, L. Jiang, P. Basu, D. Englund, and S. Guha. “Routing entanglement in the quantum internet”. In: *npj Quantum Information* 5 (2019), pp. 1–9.

[35] G. Petropoulos, F. Sardis, S. Spirou, and T. Mahmoodi. “Software-defined internetworking: Enabling coordinated QoS control across the internet”. In: *2016 23rd International Conference on Telecommunications (ICT)*. 2016, pp. 1–5. DOI: 10.1109/ICT.2016.7500361.

[36] R. Picchi, F. Chiti, R. Fantacci, and L. Pierucci. “Towards Quantum Satellite Internetworking: A Software-Defined Networking Perspective”. In: *IEEE Access* 8 (2020), pp. 210370–210381.

[37] A. Pirker and W. Dür. “A quantum network stack and protocols for reliable entanglement-based networks”. In: *ArXiv* abs/1810.03556 (2019).

[38] M. Pompili, C. D. Donne, I. t. Raa, B. v. d. Vecht, M. Skrzypczyk, G. Ferreira, L. d. Kluijver, A. Stolk, S. L. N. Hermans, P. PaweBczak, W. Kozlowski, R. Hanson, and S. Wehner. “Experimental demonstration of entanglement delivery using a quantum network stack”. In: *ArXiv* abs/2111.11332 (2021).

[39] S. Shi and C. Qian. “Concurrent Entanglement Routing for Quantum Networks: Model and Designs”. In: *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication* (2020).

[40] P. W. Shor. “Scheme for reducing decoherence in quantum computer memory”. In: *Physical Review A* 52.4 (1995), R2493–R2496. DOI: 10.1103/PhysRevA.52.R2493. URL: https://link.aps.org/doi/10.1103/PhysRevA.52.R2493.

[41] P. Wang, C.-Y. Luan, M. Qiao, M. Um, J. Zhang, Y. Wang, X. Yuan, M. Gu, J. Zhang, and K. Kim. “Single ion qubit with estimated coherence time exceeding one hour”. In: *Nature Communications* 12.1 (2021), p. 233. ISSN: 2041-1723. DOI: 10.1038/s41467-020-20330-w. URL: https://doi.org/10.1038/s41467-020-20330-w.

[42] S. Wehner, D. Elkouss, and R. Hanson. “Quantum internet: A vision for the road ahead”. In: *Science* 362.6412 (2018), eaam9288. DOI: 10.1126/science.aam9288. eprint: https://www.science.org/doi/pdf/10.1126/science.aam9288. URL: https://www.science.org/doi/abs/10.1126/science.aam9288.

[43] Y. Wu, W.-S. Bao, S. Cao, F. Chen, M.-C. Chen, X. Chen, T.-H. Chung, H. Deng, Y. Du, D. Fan, M. Gong, C. Guo, C. Guo, S. Guo, L. Han, L. Hong, H.-L. Huang, Y.-H. Huo, L. Li, N. Li, S. Li, Y. Li, F. Liang, C. Lin, J. Lin, H. Qian, D. Qiao, H. Rong, H. Su, L. Sun, L. Wang, S. Wang, D. Wu, Y. Xu, K. Yan, W. Yang,

Y. Yang, Y. Ye, J. Yin, C. Ying, J. Yu, C. Zha, C. Zhang, H. Zhang, K. Zhang, Y. Zhang, H. Zhao, Y. Zhao, L. Zhou, Q. Zhu, C.-Y. Lu, C.-Z. Peng, X. Zhu, and J.-W. Pan. “Strong Quantum Computational Advantage Using a Superconducting Quantum Processor”. In: *Physical Review Letters* 127.18 (2021), p. 180501. DOI: `10.1103/PhysRevLett.127.180501`. URL: `https://link.aps.org/doi/10.1103/PhysRevLett.127.180501`.

[44] J. Yin, Y. Cao, Y.-H. Li, S.-K. Liao, L. Zhang, J.-G. Ren, W.-Q. Cai, W.-Y. Liu, B. Li, H. Dai, G.-B. Li, Q.-M. Lu, Y.-H. Gong, Y. Xu, S.-L. Li, F.-Z. Li, Y.-Y. Yin, Z.-Q. Jiang, M. Li, J.-J. Jia, G. Ren, D. He, Y.-L. Zhou, X.-X. Zhang, N. Wang, X. Chang, Z.-C. Zhu, N.-L. Liu, Y.-A. Chen, C.-Y. Lu, R. Shu, C.-Z. Peng, J.-Y. Wang, and J.-W. Pan. “Satellite-based entanglement distribution over 1200 kilometers”. In: *Science* 356.6343 (2017), pp. 1140–1144. DOI: `10.1126/science.aan3211`. eprint: `https://www.science.org/doi/pdf/10.1126/science.aan3211`. URL: `https://www.science.org/doi/abs/10.1126/science.aan3211`.

[45] S. Zhang, S. Shi, C. Qian, and K. L. Yeung. “Fragmentation-Aware Entanglement Routing for Quantum Networks”. In: *Journal of Lightwave Technology* 39 (2021), pp. 4584–4591.

[46] H.-S. Zhong, Y.-H. Deng, J. Qin, H. Wang, M.-C. Chen, L.-C. Peng, Y.-H. Luo, D. Wu, S.-Q. Gong, H. Su, Y. Hu, P. Hu, X.-Y. Yang, W.-J. Zhang, H. Li, Y. Li, X. Jiang, L. Gan, G. Yang, L. You, Z. Wang, L. Li, N.-L. Liu, J. J. Renema, C.-Y. Lu, and J.-W. Pan. “Phase-Programmable Gaussian Boson Sampling Using Stimulated Squeezed Light”. In: *Physical Review Letters* 127.18 (2021), p. 180502. DOI: `10.1103/PhysRevLett.127.180502`. URL: `https://link.aps.org/doi/10.1103/PhysRevLett.127.180502`.

[47] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu, P. Hu, X.-Y. Yang, W.-J. Zhang, H. Li, Y. Li, X. Jiang, L. Gan, G. Yang, L. You, Z. Wang, L. Li, N.-L. Liu, C.-Y. Lu, and J.-W. Pan. “Quantum computational advantage using photons”. In: *Science* 370.6523 (Dec. 2020), pp. 1460–1463. DOI: `10.1126/science.abe8770`. URL: `https://doi.org/10.1126%2Fscience.abe8770`.