Protocols for Quantum Networks

Md. Tanvirul Islam
(B.Sc.(Hons.), BUET)

Department of Computer Science
School of Computing

National University of Singapore

A THESIS SUBMITTED FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

COMPUTER SCIENCE, SCHOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE

2016

.

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis. This thesis has also not been submitted for any degree in any university previously.

Md. Tanvirul Islam

18 January, 2016

Acknowledgements

I would like to thank my thesis supervisor Stephanie Wehner for guiding me through this journey and for giving me the freedom to pursue my own interests. I want to thank Hugh Anderson for being my academic supervisor for the last year of my PhD program. And I would also like to thank Tan Tiow Seng for giving me the opportunity and for encouraging me to pursue my PhD in the School of Computing, NUS.

I would like to thank my co-authors Iordanis Kerenidis, Loïck Magnin, Laura Mančinska, Eddie Schoute and Brandon Sorg for all the interesting collaborations.

I would also like to thank my examiners Frank Stephan and Seth Gilbert. Their reviews and valuable suggestions helped me improve the thesis.

I am very grateful to Alexander Ling and Tan Yue Chuan for the hands-on experience in experimental physics that I had in their lab.

Thanks also to my group mates Nelly Ng Huei Ying, Corsin Pfister, Jed Kaniewski, Esther Haenggi, Patric Coles, Marco Tomamichel, Mischa Woods and Ciara Morgan for all the things I have learned from them.

I want to thank Sambit Bikas Pal, Supartha Podder and Raghav Kulkarni for the countless interesting discussions we had.

I thank CQT (NUS, Singapore) and QuTech (TUDelft, Netherlands) for hosting me during my time as a PhD student.

Finally, I would like to express my infinite gratitude to my father Md. Manirul Islam and mother Fatima Islam. They have taught me to dream big and supported me throughout my life. I would

also like to thank my wife and sisters for their constant support and encouragement.

Md. Tanvirul Islam

Singapore
January 2016

Abstract

Recent scientific breakthroughs and technological advancements have demonstrated the feasibility of various quantum computing and quantum cryptographic tasks. Most of these works are focused on computation involving up to two parties where the parties are connected via a direct quantum link. However, for computations involving more than two parties the nodes have to be connected in a network. Because of the quantum nature of the communication involved, the architecture of these networks and protocols to operate them are completely different from the classical networks. Therefore, many building blocks of the classical networks do not translate to the quantum networks and require novel solutions of their own. Moreover, since the field is relatively new, these building blocks have mostly remained unaddressed so far.

In this thesis we study how our existing knowledge of the two party quantum protocols can be extended and used to build scalable multiparty quantum networks. To be more specific, we give the first fault tolerant protocols for reference frame agreement among $n > 2$ nodes in both synchronous and asynchronous quantum networks. We also study quantum routing using entanglement swapping and design efficient routing protocols for this architecture. The design and analysis techniques developed during our study of these problems provide us with valuable insights and practical tools for further advancements towards implementing a quantum Internet.

Parts of this report are based on materials contained in the following papers:

- **Spatial reference frame agreement in quantum networks**
  Tanvirul Islam, Loïck Magnin, Brandon Sorg and Stephanie Wehner
  New J. Phys. 16 063040 (2014)
  (Chapter 3)

- **Asynchronous reference frame agreement in a quantum network**
  Tanvirul Islam and Stephanie Wehner
  (Submitted, arXiv:1505.02565)
  (Chapter 4)

- **Routing in a quantum network**
  Eddie Schoute, Tanvirul Islam, Laura Mančinska, Iordanis Kerenidis and Stephanie Wehner
  (In preperation)
  (Chapter 6)

Other papers to which the author has contributed during his candidature:

- **Computability limits nonlocal correlations**
  Tanvirul Islam and Stephanie Wehner
  Physical Review A, 86, 042109 (2012)

Contents

List of Figures

LIST OF FIGURES

1

Introduction

A quantum network is a telecommunication network that allows spatially separated quantum systems to exchange quantum information. In this thesis we study problems related to the construction and operation of such quantum networks. We start this chapter describing our motivation and goals. We briefly introduce some useful concepts related to quantum information and quantum computation in Section 1.2. We describe quantum networks in general terms and discuss some of the ongoing efforts to implement them in Section 1.3. In Section 1.4 we give a brief introduction to Byzantine consensus problem. Finally, we end this chapter with a detailed outline of the thesis.

1.1 Motivation and goals

Quantum networks are gaining importance [1–5] for a variety of tasks such as quantum distributed computing [6–13], quantum cloud computing [14–16] and quantum cryptography [17–21]. There are significant advantages in using a quantum network over a classical network, especially in secure communication. We know that most of the important encryption protocols that are currently used in the internet infrastructure, such as RSA [22] and ElGamal [23], are vulnerable to attacks using quantum computers. These classical encryption protocols use computational assumptions, for example hardness of problems such as factorisation and the discrete logarithm problem to guarantee security. Since a quantum computer can efficiently solve these problems [24], these protocols are

not *quantum safe*. There are efforts in designing quantum safe protocols that have spurred the field of post-quantum cryptography [25]. There have been limited successes in developing classical protocols for which quantum attacks are not currently known [26]. However, it is not proven that these protocols are safe against any quantum adversary; whereas we know that quantum cryptographic protocols are safe against any quantum adversary [20, 27]. Moreover, implementing many quantum cryptographic protocols does not require a full-fledged universal quantum computer [28–30]. That is, quantum cryptography is feasible using current technologies and will remain secure even when quantum computers become available.

There are quantum cloud computing protocols that allow universal blind quantum computation [14]. In these interactive protocols a server carries out a quantum computation for a client such that the clients input, output and the computation remain completely private from the server. Any scalable implementation of this technology where many clients share a central server would require a quantum network.

The idea of quantum computers was originally conceived by Feynman [31] for performing quantum simulations where quantum computers provide significant advantages in solving various physical problems [32–34]. In such applications, a quantum network will allow separate quantum computers with limited resources to cooperate in solving more resource-intensive problems.

From the current architecture of the Internet one can predict that any general purpose quantum network will contain a large number of nodes that are distributed over widespread geographical locations on earth [35–37] or on satellites [38–43] and connected via quantum and classical communication channels [44].

Our study of quantum networks in this thesis can be grouped roughly in two categories. In one category we study problems related to initialising a quantum network. This involves the study of fault tolerant multiparty synchronisation problems such as reference frame agreement in synchronous and asynchronous networks, which are not only important for quantum communication but also have many important non-quantum applications. We use the term 'reference frame' in a broad sense, because depending on which quantum systems and which degrees of freedom are used to carry quantum information the meaning of reference frame

varies. It might mean for example, spatial Cartesian reference frame, or phase reference, or synchronised clocks [45]. In Chapter 3 we study the spatial reference frame agreement problem in a synchronous multiparty setting, in Chapter 4 we study the asynchronous reference frame agreement problem.

In the other category, we study network architectures that uses entanglement swapping to distribute entanglement over a quantum network. For such a network to be effective, we need efficient routing protocols. We develop the concept of *routing graphs* that allows efficient quantum routing over different network topologies and give a routing protocol using it.

Before going into the details we first introduce some preliminaries.

1.2 Preliminaries

We give an informal introduction to quantum information and briefly define the quantum primitives that we later use in this thesis. For example, the concepts of *unspeakable information* (Section 1.2.15) and using qubits to carry *direction information* (Section 1.2.13) in space are used in our reference frame agreement related works in Chapter 2, 3 and 4. And the concept of *teleportation* (Section 1.2.11) and *entanglement swapping* (Section 1.2.12) are used as primitives in the description of quantum routing networks and for designing routing protocols in Chapter 5 and 6. For an in-depth introduction to quantum computing and quantum information theory we refer to, for example [46–48].

1.2.1 Quantum information

When we talk about information processing we often think of information as an abstract sequence of symbols or bits that we control and use by encoding them in physical systems. For example, in digital computers bits are represented as states of the flip-flop circuits [49] in a register, or charge in capacitors in a memory unit, in optical fibre transmission lines bits are encoded in light pulses, in DVD's they are encoded as tiny dents, on a reflective surface. However, there is another way of looking at information. That is, we could think of information as something that describes the state of a physical system. We know that all the properties of

information emerge from the logical operations that we are able to perform on them. Without operations, a sequence of symbols is completely useless. However, what do we mean by 'performing an operation'? Since bits are descriptions of states of a system, an operation means a physical process that evolves the initial state of that system into the final state. This final state would correspond to the output of the operation. This implies, there cannot be operations that corresponds to any physically impossible evolution of such systems. This also implies, that for every possible evolution of physical systems, there will be corresponding logical operations that can be performed on the bits of information [50]. When quantum systems are considered, this later phenomenon allows us to generalise the notion of information to quantum information and to use quantum systems to perform various information processing tasks that were thought to be impossible classically.

It is known that a classical computer can in principle simulate a quantum computer with an exponential blowup in running time and memory requirements [50]. However, non-local correlations, and intrinsic randomness available in quantum information allows us to perform tasks (such as randomness expansion [51]) that are *impossible* in classical information theory. On the other hand, quantum information can be used to perform any classical information processing and communication related task. This is why quantum information is considered as a generalisation of classical information.

1.2.2 Bits and qubits

To make our previous discussion more concrete, let us imagine a two state system like the flip-flop of a computer register. It can have two perfectly distinguishable voltage levels corresponding to two different states. We arbitrarily denote one state as 0 and the other as 1. By measuring this voltage, one could retrieve the information it contains and write it down using the symbols in $\{0, 1\}$. This is called a bit. If we have a register with n such flip-flops, then a description of the register's state would be a string in $\{0, 1\}^n$, that is a concatenation of n bits. For example, when $n = 2$, the set of possible states is, $\{00, 01, 01, 11\}$. Any operation that we can perform electronically on the register, would accordingly change these bits. A sequence of such operations is called a computation. Instead

of using flip-flops we could write down the bit symbols on paper, and perform the sequence of logical operations by hand, and we would still have performed the same computation.

However, the situation can change drastically if we use a different type of system to encode our bits, for example the spin direction of a spin-1/2 particle, such as an electron. This system can also be in perfectly distinguishable states 'spin-up' and 'spin-down' and all the previous logical operations could still be performed on them [52]. Moreover, when isolated from the environment, quantum mechanics allows the system to be in a superposition of such 'up' and 'down' states. This means, the system can, to some extent, be in both states at the same time. To describe this phenomenon mathematically, let us first arbitrarily denote the 'up' state by a symbol $|0\rangle$ and the 'down' state by $|1\rangle$. According to quantum mechanics, a complete description of all possible pure states of this system would look like, $\alpha|0\rangle + \beta|1\rangle$. Where, α and β, called *amplitude*, are complex numbers satisfying $|\alpha|^2 + |\beta|^2 = 1$. Here, one can interpret $|\alpha|^2$ as the probability of the system to be in the state $|0\rangle$ and similarly $|\beta|^2$ as the probability of the system to be in the state $|1\rangle$. So, an 'up' state is where $|\alpha|^2 = 1$ and $|\beta|^2 = 0$. The choice $(1, 0)$ for 'up' and $(0, 1)$ for 'down' allows us to define a simple natural isomorphism from $\{|0\rangle, |1\rangle\}$ to $\{(1, 0), (0, 1)\}$ and to write,

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{1.1}$$

They form a two dimensional orthonormal basis[1], which we call the *computational basis*. This allows us to write their all possible superpositions as,

$$\alpha|0\rangle + \beta|1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}. \tag{1.2}$$

This shows, according to quantum mechanics, the state of such a two level system lies in a two dimensional complex space in $\mathbb{C}^2$. We call the quantum information represented by this system a *qubit*.

[1]Since, their vector inner product is 0. And they have unit lengths.

However, one might still ask, why do the two level systems like the flip-flops used in our computer's registers not show such quantum behaviour? Note that, we said, to observe quantum behaviour the system must be well isolated from the environment. This is not the case with our computer flip-flops. Even though they are governed by quantum mechanics, the constant interaction with the environment destroys their superposition [53, 54]. That is why the state of a flip-flop can be described using a single bit.

1.2.3 Two qubits

We have seen how a single two level quantum system or a qubit is mathematically described. Let us consider what happens if we have two such systems. Intuitively they can be in any of the four distinct states $|0\rangle|0\rangle, |0\rangle|1\rangle, |1\rangle|0\rangle, |1\rangle|1\rangle$, where we juxtapose the state of each qubit by arbitrarily fixing one as the first qubit. However, remember the system is quantum mechanical. So, the whole system can also be in any arbitrary superposition of these four distinct states. Formally, the system can be in a state,

$$\alpha\,|0\rangle|0\rangle + \beta\,|0\rangle|1\rangle + \gamma\,|1\rangle|0\rangle + \zeta\,|1\rangle|1\rangle\,, \tag{1.3}$$

where, α, β, γ and ζ are complex numbers (amplitudes) satisfying,

$$|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\zeta|^2 = 1. \tag{1.4}$$

Similar to the single qubit case, we get a vector representation of this combined state as well. For example, we define $|0\rangle|0\rangle$ (often written $|00\rangle$) to be the tensor product of their vector representations from (1.1). That is,

$$|0\rangle|0\rangle \equiv |00\rangle \equiv |0\rangle \otimes |0\rangle = \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}1 \times \begin{bmatrix}1\\0\end{bmatrix}\\ \\ 0 \times \begin{bmatrix}1\\0\end{bmatrix}\end{bmatrix} = \begin{bmatrix}1\\0\\0\\0\end{bmatrix}. \tag{1.5}$$

$|0\rangle|0\rangle$ Similarly, $|0\rangle|1\rangle$, $|1\rangle|0\rangle$ and $|1\rangle|1\rangle$ are also defined using tensor products.

Using this notation, the 2-qubits state of the Term (1.3) becomes,

$$\alpha\,|0\rangle|0\rangle + \beta\,|0\rangle|1\rangle + \gamma\,|1\rangle|0\rangle + \zeta\,|1\rangle|1\rangle = \begin{bmatrix}\alpha\\ \beta\\ \gamma\\ \zeta\end{bmatrix}. \tag{1.6}$$

If there are n qubits in a system, then each qubit can be in one of the two distinct orthonormal computational basis states as in Equation (1.1). Their combined state can be in 2^n distinct computational basis state as defined by the tensor product. So, the whole system acts as a multilevel quantum system (defined in Section 1.2.5) with states in $\mathbb{C}^{2^n}$.

1.2.4 Product states and entangled states

If two distinct qubits are in state $|\psi\rangle$ and $|\phi\rangle$ where, in computational basis the first qubit is,

$$|\psi\rangle = \alpha_1\,|0\rangle + \beta_1\,|1\rangle = \begin{bmatrix}\alpha_1\\ \beta_1\end{bmatrix}, \tag{1.7}$$

and the second qubit is,

$$|\phi\rangle = \alpha_2\,|0\rangle + \beta_2\,|1\rangle = \begin{bmatrix}\alpha_2\\ \beta_2\end{bmatrix}, \tag{1.8}$$

then their combined system is said to be in a *product* state,

$$|\psi\rangle|\phi\rangle = |\psi\rangle \otimes |\phi\rangle = \begin{bmatrix}\alpha_1\\ \beta_1\end{bmatrix} \otimes \begin{bmatrix}\alpha_2\\ \beta_2\end{bmatrix} = \begin{bmatrix}\alpha_1 \times \begin{bmatrix}\alpha_2\\ \beta_2\end{bmatrix}\\ \\ \beta_1 \times \begin{bmatrix}\alpha_2\\ \beta_2\end{bmatrix}\end{bmatrix} = \begin{bmatrix}\alpha_1\alpha_2\\ \alpha_1\beta_2\\ \beta_1\alpha_2\\ \beta_1\beta_2\end{bmatrix}, \tag{1.9}$$

$$= \alpha_1\alpha_2\,|0\rangle|0\rangle + \alpha_1\beta_2\,|0\rangle|1\rangle + \beta_1\alpha_2\,|1\rangle|0\rangle + \beta_1\beta_2\,|1\rangle|1\rangle\,. \tag{1.10}$$

It means this combined state can be factorised as,

$$|\psi\rangle|\phi\rangle = (\alpha_1\,|0\rangle + \beta_1\,|1\rangle) \otimes (\alpha_2\,|0\rangle + \beta_2\,|1\rangle). \tag{1.11}$$

However not all 2 qubits quantum states can be factorised like this. Such unfactorable states are called *entangled* states. For example, $|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$ is an entangled state. Quantum entanglement plays a key role in many of the most interesting applications of quantum computation and quantum information. It is a uniquely quantum mechanical phenomenon that allows us to perform operations such as teleportation (Section 1.2.11), which has no classical analog. Before we can fully appreciate this idea we would need to introduce a few more concepts.

1.2.5 Multilevel systems

A d level closed quantum system can be described as a vector in $\mathbb{C}^d$ as,

$$|\Psi\rangle = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_d \end{bmatrix}, \tag{1.12}$$

where $|\Psi\rangle$ is a unit vector. That is $\sum_i |\alpha_i|^2 = 1$

We denote $\langle\Psi|$ to be the conjugate transpose of the vector $|\Psi\rangle$, which is defined to be,

$$\langle\Psi| = \begin{bmatrix} \alpha_1^* & \alpha_2^* & \cdots & \alpha_d^* \end{bmatrix}. \tag{1.13}$$

1.2.5.1 Inner product

If another d level state $|\Phi\rangle$ is described as,

$$|\Phi\rangle = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_d \end{bmatrix}. \tag{1.14}$$

then the *inner product* of $|\Phi\rangle$ and $|\Psi\rangle$ is defined as,

$$(|\Psi\rangle, |\Phi\rangle) = \langle\Psi|\Phi\rangle = \begin{bmatrix}\alpha_1^* & \alpha_2^* & \cdots & \alpha_d^*\end{bmatrix} \times \begin{bmatrix}\beta_1 \\ \beta_2 \\ \vdots \\ \beta_d\end{bmatrix}. \tag{1.15}$$

Here '$\times$' represents matrix multiplication.

Taking the inner product of the quantum state $|\Psi\rangle$ with itself we get,

$$\langle\Psi|\Psi\rangle = \begin{bmatrix}\alpha_1^* & \alpha_2^* & \cdots & \alpha_d^*\end{bmatrix} \times \begin{bmatrix}\alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_d\end{bmatrix}, \tag{1.16}$$

$$= \sum_i |\alpha_i|^2 = 1. \tag{1.17}$$

A vector space with an inner product defined on its elements is called an *inner product space*.

1.2.5.2 Linear operators and Unitary operators

A *linear operator* that takes elements from vector space A to B is defined by a function $L : A \to B$ which is linear in its inputs,

$$L\left(\sum_i \alpha_i |\psi_i\rangle\right) = \sum_i \alpha_i L(|\psi_i\rangle) \tag{1.18}$$

If a linear operator maps elements form a vector space A to itself, then it is said to be *defined on* A. If such an operator maps an element to itself, then it is called an *identity operator*.

That is, an identity operator $I_A : A \to A$ would satisfy $I_A(|\psi\rangle) = |\psi\rangle$, where $|\psi\rangle$ is an element in the vector space A.

For $|\psi\rangle$, $|\phi\rangle$ in some inner product space A, if linear operators L and $L^\dagger$ on this space satisfies

$$(|\psi\rangle, L|\phi\rangle) = \left(L^\dagger|\psi\rangle, |\phi\rangle\right), \tag{1.19}$$

then $L^\dagger$ is called the *Hermitian conjugate* or the *adjoint* of the operator L.

If $L^\dagger = L$, then L is called a *Hermitian operator*.

If such a Hermitian operator satisfies

$$\langle\phi| L |\phi\rangle \geq 0, \tag{1.20}$$

for any $|\phi\rangle$ in the inner product space A, then L is called a *positive operator*.

Any linear operator acting on vector spaces has an equivalent *matrix representation*. For a linear operator $L : A \to B$ from vector space A to B we often use the symbol L as an operator or as a matrix interchangeably.

An operator U is called *unitary* if $U^\dagger U = I$. Such operators preserve inner products between vectors. To see this, let $|\psi\rangle$ and $|\phi\rangle$ be two state vectors. The inner product of $|\psi\rangle$ and $|\phi\rangle$ is the same as the inner product between $U|\psi\rangle$ and $U|\phi\rangle$,

$$(U|\psi\rangle, U|\phi\rangle) = \langle\psi| U^\dagger U |\phi\rangle = \langle\psi| I |\phi\rangle = \langle\psi|\phi\rangle. \tag{1.21}$$

1.2.5.3 Trace

Trace of a matrix L is defined as

$$\mathrm{tr}(L) = \sum_i L_{ii}, \tag{1.22}$$

where L_{ii} is the ith diagonal element of the matrix L. The trace operator is *linear* $\mathrm{tr}(L + M) = \mathrm{tr}(L) + \mathrm{tr}(M)$ and $\mathrm{tr}(\alpha L) = \alpha(L)$, where $\alpha \in \mathbb{C}$, and *cyclic* $\mathrm{tr}(LM) = \mathrm{tr}(ML)$.

For an unitary operator U, the cyclic property gives us $\mathrm{tr}(ULU^\dagger) = \mathrm{tr}(U^\dagger UL) = \mathrm{tr}(IL) = \mathrm{tr}(L)$. That is, trace is invariant under unitary *similarity transform* $L \to ULU^\dagger$.

This allows us to define the trace of any linear operator L to be the trace of any of its matrix representations.

1.2.5.4 Postulates regarding quantum states

So far, we have seen qubits and quantum states in an abstract mathematical sense. What binds these mathematical objects to the physics are the postulates

of quantum mechanics. We present the first postulate that relates an isolated physical system with complex vector spaces. This formulation of the postulate is taken from [47].

Postulate 1. *Associated to any isolated physical system is a complex vector space with inner product known as 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.*

The next postulate justifies our use of tensor products to represent multi-qubit system.

Postulate 2. *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* $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \ldots \otimes |\psi_n\rangle$*.*

1.2.5.5 Outer product

The *outer product* of $|\Phi\rangle$ with $|\Psi\rangle$ is defined as,

$$|\Phi\rangle \langle\Psi| = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_d \end{bmatrix} \times \begin{bmatrix} \alpha_1^* & \alpha_2^* & \cdots & \alpha_d^* \end{bmatrix}, \tag{1.23}$$

where '$\times$' is the matrix multiplication.

1.2.6 Mixed states

States $|\Psi\rangle$, that are in the form of Equation (1.12) are called pure states. Using the outer product formalism such pure states are often expressed as,

$$|\Psi\rangle\langle\Psi| = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_d \end{bmatrix} \times \begin{bmatrix} \alpha_1^* & \alpha_2^* & \cdots & \alpha_d^* \end{bmatrix}. \tag{1.24}$$

We can see there is a one-to-one correspondence between states like $|\Psi\rangle$ and matrices like $|\Psi\rangle\langle\Psi|$. Since, $|\Psi\rangle$ is a quantum state the sum of the squares of its amplitudes is 1. This gives us,

$$\mathrm{tr}(|\Psi\rangle\langle\Psi|) = 1, \tag{1.25}$$

where tr is the trace operator.

This change in representation allows us to define a powerful formalism called the *density operator*. Imagine a machine that prepares a d dimensional quantum system in state $|\Psi_i\rangle$ with probability p_i. Let us assume we know the probability distribution but do not know exactly which state is prepared. Therefore, upon reception of such a system, to us, the state would look like a probabilistic mixture of states $|\Psi_i\rangle$. Such mixed states are described as,

$$\rho = \sum_i p_i |\Psi_i\rangle\langle\Psi_i| . \tag{1.26}$$

ρ is called the *density operator* or the *density matrix* of the ensemble $\{p_i, |\Psi_i\rangle\}$. Density operators are the most general description of a quantum system.

Since, p_i is a probability distribution using Equation (1.25) we have

$$\mathrm{tr}(\rho) = \mathrm{tr}\left(\sum_i p_i |\Psi_i\rangle\langle\Psi_i|\right) = \sum_i p_i \mathrm{tr}(|\Psi_i\rangle\langle\Psi_i|) = 1. \tag{1.27}$$

1.2.6.1 Maximally mixed state

If a collection $\{|\psi\rangle_i\}$ of state vectors form an orthonormal basis of the state space of a system, then we have

$$\sum_i |\psi_i\rangle\langle\psi_i| = I, \tag{1.28}$$

where I is the identity operator, or equivalently, the identity matrix. If ψ is of dimension d then the state I/d can be interpreted as a density operator, which is called the *maximally mixed state* of that system.

1.2.6.2 Partial trace

If a collection $\{|\psi\rangle_i\}$ state factors form an orthonormal basis of a state space A and the collection $\{|\phi\rangle_i\}$ for state space B, then from Postulate 2, we know that the orthonormal basis of the joint state space of A and B is $\{|\psi\rangle_i \otimes |\phi\rangle_j\}$.

If a density operator ρ_{AB} describes a system with the joint state space A and B, then

$$\rho_A = \mathrm{tr}_B(\rho_{AB}), \tag{1.29}$$

is the reduced operator for system A, where tr_B is the partial trace, defined by

$$\mathrm{tr}_B(|\psi_1\rangle \otimes |\phi_1\rangle \langle\psi_2| \otimes \langle\phi_2|) = \mathrm{tr}(|\phi_1\rangle \langle\phi_2|) |\psi_1\rangle \langle\psi_2| . \tag{1.30}$$

We say the system B is *traced out* from ρ_{AB}.

1.2.7 Measurements

A quantum system might be in a superposition of multiple orthonormal states. Given an object in an unknown quantum state one can gain information about it only via a quantum operation called *measurement*. A measurement might have several outcomes, depending on the state of the system and the measurement operation performed. These outcomes occur probabilistically. A measurement operation is characterised by some measurement operators where each operator corresponds to an outcome of the measurement. This correspondence of the measurement operator and the physically measuring the system is also a fundamental construct of quantum mechanics. It is formalised in [47] as follows,

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* ρ *immediately before the measurement then the probability that result* m *occurs is given by*

$$p(m) = \mathrm{tr}(M_m^\dagger M_m \rho), \tag{1.31}$$

and the state of the system after the measurement is

$$\frac{M_m \rho M_m^\dagger}{\mathrm{tr}(M_m^\dagger M_m \rho)}. \tag{1.32}$$

The measurement operators satisfy the completeness equation,

$$\sum_m M_m^\dagger M_m = I. \tag{1.33}$$

1.2.8 The Pauli matrices

A useful set of matrices, called the *Pauli matrices* [47] are defined as,

$$\sigma_0 \equiv I \equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{1.34}$$

$$\sigma_1 \equiv \sigma_x \equiv X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \tag{1.35}$$

$$\sigma_2 \equiv \sigma_y \equiv Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \tag{1.36}$$

$$\sigma_3 \equiv \sigma_z \equiv Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{1.37}$$

It is easily checked that the operators represented by the Pauli matrices are unitary.

1.2.9 Evolution

A closed quantum system might evolve over time due to its internal dynamics. Within the state space, this evolution is perceived as a unitary transformation of the state vector (or equivalently, the density operator) that describe the system. This correspondence of the evolution of a physical system with the unitary transformation of the density operator describing its state, is also a postulate of quantum mechanics. It is formulated in [47] as follows,

Postulate 4. *The evolution of a* closed *quantum system is described by a* unitary transformation. *That is, the state ρ of the system at time t_1 is related to the state*

ρ' of the system at time t_2 by a unitary operator U which depends only on the time t_1 and t_2,

$$\rho' = U^\dagger \rho U. \tag{1.38}$$

1.2.9.1 Evolution of open quantum systems and quantum channels

We can use Postulate 4 to describe the evolution of a open quantum system as well. To see this, let us consider an open quantum system that interacts with the environment. If we consider the whole environment and the system together, then the total system can be considered as a closed quantum system.

In an open quantum system a quantum state might interact with the environment, or some other objects. To see this let us consider a quantum system described by density operator ρ. If this system is open, then it must be interacting with an external system. Let us call this external system the environment. Initially, the environment is in state ρ_{env}. Now, their joint system $\rho \otimes \rho_{\text{env}}$ can be considered as a closed system. According to the Postulate 4, the evolution of the joint system is controlled by a unitary operator U. Therefore the evolved system is described by

$$U^\dagger(\rho \otimes \rho_{\text{env}})U. \tag{1.39}$$

After this evolution if we trace out the environment then we get the final state,

$$\mathcal{E}(\rho) = \text{tr}_{\text{env}}(U^\dagger(\rho \otimes \rho_{\text{env}})U). \tag{1.40}$$

That is, if a open quantum system in the state state ρ evolves into some ρ', then we can always find an operator of the form $\mathcal{E}$ that maps the initial state to the later.

$$\rho' = \mathcal{E}(\rho). \tag{1.41}$$

This operator $\mathcal{E}$ that acts on other operators (the density operators that describe quantum states) is called a *superoperator*.

Since ρ' is a density operator, from Equation (1.27) we have

$$\mathrm{tr}(\rho') = (\mathcal{E}(\rho)) = 1. \tag{1.42}$$

That is the superoperators $\mathcal{E}$ is trace preserving.

If the state space of the environment has an orthonormal basis $\{|\psi_i\rangle\}$ and initially $\rho_{\mathrm{env}} = |\psi_0\rangle\langle\psi_0|$, then

$$\mathcal{E}(\rho) = \sum_i \langle\psi_i| U \left[\rho \otimes \langle\psi_0| |\psi_0\rangle\right] U^\dagger |\psi_i\rangle , \tag{1.43}$$

$$= \sum_i E_i \rho E_i^\dagger, \tag{1.44}$$

where $E_i = \langle\psi_i| U |\psi_0\rangle$ is an operator on the state space of the system.

Equation (1.44) is the *operator sum representation* of the superoperator $\mathcal{E}$.

Since superoperators are trace preserving (Equation (1.42)), one can show that

$$\sum_k E_k^\dagger E_k = I. \tag{1.45}$$

This is called the *completeness relation*.

If the operator sum representation of a superoperator $\mathcal{E}$ satisfies Equation (1.45) then it is called a *quantum channel*.

Physical quantum links (for example, an optical fibre that carries photon qubits) that carry quantum information from one place to another through a physical medium are characterised by such quantum channels. To see the intuition behind this, note that while propagating through a physical medium, the qubit may interact with it. However, If we only look at the qubit system that enters the link and want to know its state when it exits, then we have to consider it as an open quantum system where the physical medium is considered as the 'environment'. Therefore, its evolution would be described by a quantum channel.

1.2.9.2 Depolarising channel

An important example of a quantum channel is the *depolarising channel* that takes in a d dimensional state ρ and either with probability p outputs a completely mixed state I/d or with probability $(1-p)$ outputs the original state ρ. Therefore,

with probability p the state information is completely lost in this channel. That is,

$$\mathcal{E}(\rho) = p\frac{I}{d} + (1-p)\rho. \tag{1.46}$$

If ρ is a qubit, that is $d = 2$, then the map $\mathcal{E}$ has an operator sum representation where operator elements $\{\sqrt{1-3p/4}I, \sqrt{p}X/2, \sqrt{p}Y/2, \sqrt{p}Z/2\}$ are expressed using the Pauli operators.

1.2.10 Bell states

The two qubit states,

$$|\Phi^+\rangle_{AB} = \frac{1}{\sqrt{2}}(|0\rangle_A \otimes |0\rangle_B + |1\rangle_A \otimes |1\rangle_B), \tag{1.47}$$

$$|\Phi^-\rangle_{AB} = \frac{1}{\sqrt{2}}(|0\rangle_A \otimes |0\rangle_B - |1\rangle_A \otimes |1\rangle_B), \tag{1.48}$$

$$|\Psi^+\rangle_{AB} = \frac{1}{\sqrt{2}}(|0\rangle_A \otimes |1\rangle_B + |1\rangle_A \otimes |0\rangle_B), \tag{1.49}$$

$$|\Psi^-\rangle_{AB} = \frac{1}{\sqrt{2}}(|0\rangle_A \otimes |1\rangle_B - |1\rangle_A \otimes |0\rangle_B), \tag{1.50}$$

are called the *Bell states*. A pair of qubits that are in a Bell state is called a *Bell Pair*.

For example, let us assume Alice and Bob share the Bell state $|\Psi^+\rangle_{AB}$ where Alice holds the qubit A and *Bob* holds the qubit B and they are spatially separated.

If Alice projectively measures her qubit in the computational basis $\{|0\rangle, |1\rangle\}$ then with probability $1/2$ she will get outcome 0 and with probability $1/2$ outcome 1. Now, if Bob performs the same measurement on his qubit B under the condition that Alice has already observed outcome 0, then he would definitely observe outcome 1. If instead Alice has received outcome 1, then Bobs outcome would be 0. That is their outcome are perfectly anti-correlated even thought they are spatially separated from each other.

1.2.11 Teleportation

Teleportation [55] is one of the unique features of quantum information. If Alice has a qubit $|\psi\rangle_C = \alpha|0\rangle_C + \beta|1\rangle_C$, she can teleport it to Bob using a pre-shared

entangled Bell state. For example if Alice and Bob share the Bell state $|\Phi^+\rangle_{AB}$, where Alice holds qubit A and Bob holds qubit B, then the state of the combined system is given by,

$$|\Phi^+\rangle_{AB} \otimes |\psi\rangle_C = \frac{1}{\sqrt{2}}(|0\rangle_A \otimes |0\rangle_B + |1\rangle_A \otimes |1\rangle_B) \otimes (\alpha|0\rangle_C + \beta|1\rangle_C). \quad (1.51)$$

Using Equation (1.47) to Equation (1.50), this can be written as,

$$\begin{aligned} |\Phi^+\rangle_{AB} \ \otimes \ |\psi\rangle_C = \\ \frac{1}{2}\Big[\ |\Phi^+\rangle_{AC} \otimes (\alpha|0\rangle_B + \beta|1\rangle_B) \ + \ |\Phi^-\rangle_{AC} \otimes (\alpha|0\rangle_B - \beta|1\rangle_B) \\ + \ |\Psi^+\rangle_{AC} \otimes (\beta|0\rangle_B + \alpha|1\rangle_B) \ + \ |\Psi^-\rangle_{AC} \otimes (\beta|0\rangle_B - \alpha|1\rangle_B)\Big]. \end{aligned} \quad (1.52)$$

Now Alice performs a measurement of Bell operator (defined in [56]) on her share of the qubits AC and depending on which outcome she gets, the post measurement state of the system becomes one of these,

$$|\Phi^+\rangle_{AC} \otimes (\alpha|0\rangle_B + \beta|1\rangle_B), \quad (1.53)$$

$$|\Phi^-\rangle_{AC} \otimes (\alpha|0\rangle_B - \beta|1\rangle_B), \quad (1.54)$$

$$|\Psi^+\rangle_{AC} \otimes (\beta|0\rangle_B + \alpha|1\rangle_B), \quad (1.55)$$

$$|\Psi^-\rangle_{AC} \otimes (\beta|0\rangle_B - \alpha|1\rangle_B). \quad (1.56)$$

At this point, Alice's two qubits AC are in a Bell state. And the entanglement between qubits A and B is broken. Moreover B is in a state whose amplitudes are now related to $|\psi\rangle_C$. Now, if Alice communicates the outcome that she has observed to Bob, then, knowing this, Bob can apply the required correction unitary and reconstruct $|\psi\rangle_C$ in his qubit B as $\alpha|0\rangle_B + \beta|1\rangle_B$.

For example, If Alice's result is $|\Phi^+\rangle_{AC}$, then Bob knows he already has $\alpha|0\rangle_B + \beta|1\rangle_B$ and does nothing.

If Alice's result is $|\Phi^-\rangle_{AC}$ then Bob would apply a unitary operator (Pauli matrix) $\sigma_3 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ on B. This converts the qubit to $\alpha|0\rangle_B + \beta|1\rangle_B$.

If Alice's result is $|\Psi^+\rangle_{AC}$ then Bob applies Pauli operator σ_1 to construct $\alpha|0\rangle_B + \beta|1\rangle_B$.

For the remaining cases Bob applies, $-\sigma_3\sigma_1 = i\sigma_2 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$.

Here we emphasise that Bob can reconstruct $|\psi\rangle_C$ only after receiving information from Alice about which one of the four outcomes she has observed. Since this message must travel within the limit of the speed of light, quantum teleportation cannot be used for instantaneous communication.

1.2.12 Entanglement swapping

Using similar techniques used in teleportation we can perform a more interesting operation, namely *entanglement swapping*. The idea was originally proposed by Zukowski *et al.* in [57] and later experimentally demonstrated in various settings [58, 59]. A formulation of this process, proposed by Biham *et al.* in [60] is most useful to us. We present it with an example below.

Let us assume that qubits A and B are in the Bell state $|\Phi^+\rangle_{AB}$ (see Equation (1.47)) and qubits C and D are in the Bell state $|\Phi^+\rangle_{CD}$. We write the joint state of the four qubits A, B, C and D as,

$$\begin{aligned} \left|\Phi^+\right\rangle_{AB} \otimes \left|\Phi^+\right\rangle_{CD} = \frac{1}{2}[&|0\rangle_A |0\rangle_B |0\rangle_C |0\rangle_D + |1\rangle_A |1\rangle_B |0\rangle_C |0\rangle_D \,, \\ &+ |0\rangle_A |0\rangle_B |1\rangle_C |1\rangle_D \,, \\ &+ |1\rangle_A |1\rangle_B |1\rangle_C |1\rangle_D]. \end{aligned} \tag{1.57}$$

Now, if we perform a Bell state measurement (defined in [56]) on the pair of qubits BC, depending on the four possible outcomes of the measurement $1, 2, 3$ and 4, the joint state of the four qubits system becomes one of these,

$$|1\rangle := \frac{1}{2}[(|0\rangle_B |0\rangle_C + |1\rangle_B |1\rangle_C) \otimes (|0\rangle_A |0\rangle_D + |1\rangle_A |1\rangle_D)], \tag{1.58}$$

$$|2\rangle := \frac{1}{2}[(|0\rangle_B |0\rangle_C - |1\rangle_B |1\rangle_C) \otimes (|0\rangle_A |0\rangle_D - |1\rangle_A |1\rangle_D)], \tag{1.59}$$

$$|3\rangle := \frac{1}{2}[(|0\rangle_B |1\rangle_C + |0\rangle_B |1\rangle_C) \otimes (|0\rangle_A |1\rangle_D + |0\rangle_A |1\rangle_D)], \tag{1.60}$$

$$|4\rangle := \frac{1}{2}[(|0\rangle_B |1\rangle_C - |0\rangle_B |1\rangle_C) \otimes (|0\rangle_A |1\rangle_D - |0\rangle_A |1\rangle_D)]. \tag{1.61}$$

respectively.

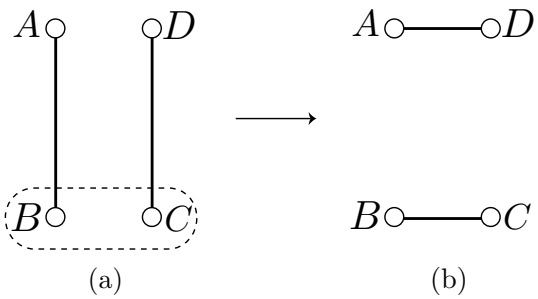


Figure 1.1: Entanglement swapping using the Bell state measurement - The qubits are represented using small circles. The solid lines connect qubits that are in Bell states. In (a) the dashed rounded rectangle encloses the two qubits that are measured using the Bell operator. (b) shows that the entanglements are swapped after the measurement.

These are essentially, (see Bell states from Equation (1.47) to (1.50))

$$|1\rangle = \left|\Phi^+\right\rangle_{BC} \otimes \left|\Phi^+\right\rangle_{AD}, \tag{1.62}$$

$$|2\rangle = \left|\Phi^-\right\rangle_{BC} \otimes \left|\Phi^-\right\rangle_{AD}, \tag{1.63}$$

$$|3\rangle = \left|\Psi^+\right\rangle_{BC} \otimes \left|\Psi^+\right\rangle_{AD}, \tag{1.64}$$

$$|4\rangle = \left|\Psi^+\right\rangle_{BC} \otimes \left|\Psi^+\right\rangle_{AD}. \tag{1.65}$$

That is, now qubit pairs BC and AD are pairwise entangled in the Bell states. Figure 1.1 justifies the name *entanglement swapping* for this procedure.

It is a remarkable quantum effect that qubits A and D, which never came in contact with each other, have became maximally entangled using entanglement swapping. If we started with any Bell state other than $|\Phi^+\rangle$s then we would have got the same entanglement swapping effect.

1.2.12.1 Entangling qubits in remote nodes using entanglement swapping

We use this entanglement swapping procedure as a primitive to create entanglement between remote nodes in a quantum network. We call the primitive an *entanglement swapping operation* and represent it as follows. Imagine three

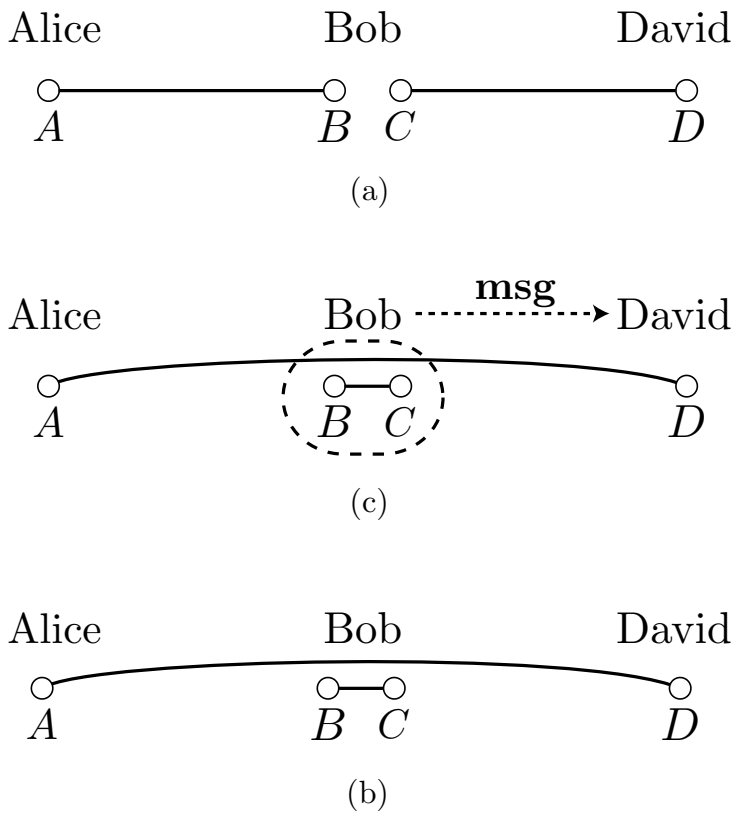


Figure 1.2: Operation *eswap*(B, C) to remotely allocate entanglement - The qubits are represented using small circles. The solid lines connect qubits that are in the Bell state $|\Phi^+\rangle$. In (b) Bob performs Bell state measurement on his qubits and communicates the outcome in a classical message **msg** to David. In (c) upon receiving **msg**, David performs the necessary local operation on D to convert the newly formed entanglement between D and Alice's qubit A to the Bell state $|\Phi^+\rangle_{AD}$.

players Alice, Bob and David. Alice holds qubit A, Bob holds qubits B and C and David holds qubit D (see Figure 1.2). Qubits A, B are entangled in the Bell state $|\Phi^+\rangle_{AB}$ and qubits C, D are in the bell state $|\Phi^+\rangle_{CD}$. Bob, who has the qubits B and C in his possession, performs the Bell state measurement on them and gets one of the 4 outcomes indicating the post measurement states as in Equation (1.62) to (1.65). For example, if the Bell measurements outcome was 3 then the post measurement state of qubits A an D is $|\Psi^+\rangle_{AD}$.

At this point though Alice's and David's qubits got remotely entangled, they do not know which of the 4 Bell states they are in. However, Bob has observed the outcome of the Bell measurement so he can communicate his outcome to David. This communication is done using classical information. Upon receiving this message from Bob, David performs local operations on its qubit D making the final state of AD to be $|\Phi^+\rangle_{AD}$. That is, even though AD was equally likely to be in any of the 4 Bell states, the message from Bob allows David to convert the entanglement to any specific shared Bell state between A and D. This whole process is packaged into the operation $eswap(B, C)$ as illustrated in Figure 1.2.

1.2.12.2 Bloch sphere

Any density matrix ρ for a single qubit can be written as,

$$\rho = \frac{I + r_x\sigma_x + r_y\sigma_y + r_z\sigma_z}{2}, \tag{1.66}$$

where, $\vec{r} = (r_x, r_y, r_z) \in \mathbb{R}^3$, such that, $\|\vec{r}\| \leq 1$ and σ_x, σ_y and σ_z are Pauli matrices.

This allows us to represent any qubit as a vector in a 3 dimensional unit sphere called *Bloch sphere* [47] (see Figure 1.3). The vector $\vec{r}$ is called the *Bloch vector* of the qubit.

1.2.13 Direction as a qubit

Incidentally, when a qubit is implemented with two level systems like electron spin [61], or photon polarisation [62], the vector $\vec{r}$ actually indicates a direction in space [47]. That is, the $\hat{\mathbf{x}}, \hat{\mathbf{y}}$ and $\hat{\mathbf{z}}$ axes of the Bloch vector are not only the

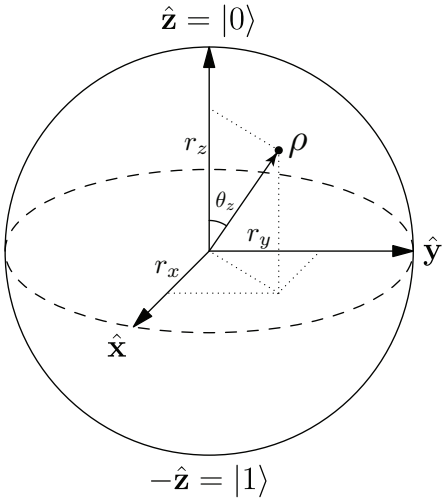


Figure 1.3: Bloch sphere - The qubit ρ is represented as a vector (r_x, r_y, r_z) within the unit sphere.

axes of the state space, but also the axes of the physical spatial reference frame. That is why, a Bloch vector of such a system can act as an arrow that points to a direction in space.

1.2.14 Pauli measurements

The direction of a Bloch vector cannot be decoded from a single qubit. However, if many identical qubits with Bloch vectors pointing to the same direction are available, then a type of measurement, called the *Pauli measurement*, allows us to approximate it [63]. For example *Pauli* σ_z *measurement* is defined by measurement operators $\{\frac{(I+\sigma_z)}{2}, \frac{(I-\sigma_z)}{2}\}$. It can be applied n times on n of the identical qubits. The statistics of the outcome is governed by the angle θ_z (see Figure 1.3) that the state makes with the $\hat{\mathbf{z}}$ axis. From this, one can approximate r_z. Similarly using $\{\frac{(I+\sigma_x)}{2}, \frac{(I-\sigma_x)}{2}\}$ and $\{\frac{(I+\sigma_y)}{2}, \frac{(I-\sigma_y)}{2}\}$, which are Pauli σ_x and Pauli σ_y measurements respectively, it is possible to approximate r_x and r_y. The approximation error depends on the number of qubits exchanged and can be estimated using Hoeffding's inequality (see, Chapter 2.6).

1.2.15 Unspeakable information

Imagine two players Alice and Bob who share a spatial reference frame. That is, they know the relative orientation of each others local Cartesian reference frame. If Alice wants Bob to point to a certain direction, then she can send him the description (coordinates) of the vector pointing to that direction and upon reception Bob will be able to orient himself as intended. This method only works because they have a common reference frame. Could they still communicate the direction if they did not have a shared reference frame? Clearly, it is not possible by exchanging only bits. However, if they could exchange an object that points to the intended direction, like a compass needle, then they would be able to communicate the direction. Here the object will carry the reference frame with it (for an example see [64]). Luckily, as we have seen, qubits can act as such needles and allow Alice and Bob to communicate a direction from scratch. The information carried by such systems is called *unspeakable information* or *non-fungible information* [45], in contrast to the *fungible* information [64] considered in classical information theory where the means of encoding does not matter.

Communication of such reference frames is very important because without them it is impossible to make sense of any quantum state encoded in some directional quantity such as spin and polarisation. For example, let us assume a spin qubit is prepared in state $|0\rangle$, as expressed in the computational basis, and given to us. The Bloch vector $\vec{r}$ for this state is $(1, 0, 0)$. If the orientations of the $\hat{\mathbf{x}}, \hat{\mathbf{y}}$ and $\hat{\mathbf{z}}$ (see Figure 1.3) are not known to us, then $(1, 0, 0)$ could be interpreted as any point on the surface of the Bloch sphere. So, to us, the quantum state will look completely arbitrary. Thus, no effective quantum communication will take place.

Formally, If Alice and Bob have two different local reference frames, then there will be a unitary operator $\mathcal{U}$ that takes states descriptions by Alice and transforms them into the equivalent state description for Bob's frame. Absence of a shared reference frame means, not knowing this unitary. If Alice prepares a state $|\psi\rangle$ and tells Bob in which state she has prepared her system, and Bob does not know which unitary operation transforms the description to a equivalent state in his frame, then any state he prepares to reconstruct $|\psi\rangle$ will look (to Alice) as if a

random unitary $\mathcal{U}$ is applied on $|\psi\rangle$. That is, $\mathcal{U}$ defines an isomorphism between Alice and Bob's local experimental operations. Therefore, if Alice prepares a qubit in the state $|\psi\rangle$ and sends it to Bob, then Bob's representation of the state would be obtained by averaging over all possible isomorphism,

$$\int d\mathcal{U}\, \mathcal{U}\, |\psi\rangle\langle\psi|\, \mathcal{U}^{\dagger} = \frac{I}{2}, \tag{1.67}$$

which is a *completely mixed state* . Equation (1.67) is true for all possible $|\psi\rangle$ thus a state description without reference frame carries no information about the system [65].

1.3 Quantum networks

A quantum network consists of multiple quantum nodes with varying degrees of quantum computing capabilities and quantum links that connect adjacent quantum nodes and allow exchange of quantum information between them. The nodes might be full-fledged quantum computers that can perform universal quantum computing or simple terminals that use a remote quantum computer over the network. These terminals might have simple capabilities of preparing and measuring certain quantum states and often do not require quantum memories [66]. Or, the nodes might be quantum routers (Chapter 5.4), which, depending on the implementation, might have a limited quantum memory and the ability to perform only a limited number of operations on their memory. Example of such limited operations are the teleportation and the entanglement swapping operations. Since a router does not perform any general purpose quantum computation, rather facilitates communication over a quantum network, it does not need to be a universal quantum computer. The quantum links might be optical fibre carrying coherent single photon pulses [67], or line of sight free space photon channels [68].

In this thesis our focus is on various problems related to quantum networks that allow $n \geq 2$ party quantum computations.

1.4 Byzantine consensus

In this section we give some preliminaries to the concept of the consensus problems in the classical distributed computing. We also introduce the Byzantine fault tolerance model and a brief description of historical development of their studies in synchronous and asynchronous networks. These works, though not directly applicable in the quantum network, can give us valuable insight and useful tools in designing reference frame agreement protocols in Chapter 3 and Chapter 4. We first define general consensus problem.

1.4.1 Consensus

In distributed computing a fundamental problem is to achieve reliability where some of the processes might be faulty. A basic step of achieving such reliability involves all the process to agree on some value. For example, all the process might want to agree on a classical bit, 0 or 1 before they could make a collective decision over the network. This problem known as the consensus problem was first introduced by Pease, Shostak and Lamport in [69]. In a network where each process can be thought of as a network node, this problem can be defined more formally as,

Definition 1. A protocol among n nodes is a Classical-Consensus protocol, if each node P_i starts with an input bit g_i and outputs a bit y_i, that satisfies the following properties:

Agreement All correct nodes should output the same bit;

Validity If all correct nodes start with the same input $g_i = b$, then they should all output this value, that is $y_i = b$.

Integrity If all the correct nodes output a bit b, then there must be at least one correct node P_i that has originally proposed the bit $g_i = b$.

Termination Each correct node must complete the protocol by successfully outputing a bit.

If a protocol among n nodes can guarantee consensus in the presence of t faulty nodes, then the protocol is called t-resilient.

The importance of the consensus problem in fault tolerant computing have inspired a plethora of works in this topic (See, for example, [70, 71] for surveys). All of these results are achieved under various assumptions on the model of computing and the network types.

1.4.2 Model of computation

There are several fault model under which the consensus problems are studied.

1.4.2.1 Fail-stop faults

The most basic type of faults under which a consensus protocol must have some resilience is the crash failure. Here some node stops completely and never resumes operation during the course of the protocol. These model of faults are called the *fail-stop model* [72]. In the fail stop model the non-faulty nodes in the network can identify which node has failed and take measures accordingly.

1.4.2.2 Byzantine faults

The most challenging model of faults in a distributed computing system is the Byzantine faults [73]. Under this model of faults a faulty node might show arbitrarily faulty behaviour. That is, a Byzantine faulty node cab eveb be indistinguishable from a non-faulty node from its communication-input-output behaviour. However it might send different inconsistent messages to different correct nodes and thus confuse them. It is also assumed that the all the faulty nodes might cooperate with each other and deploy arbitrarily sophisticated strategies to thwart the consensus effort. A protocol designed under the assumption of such powerful faulty node would be strong enough to survive any network faults. If a consensus protocol can survive in the presence of t faulty nodes then it is called a Byzantine conesn called a *Byzantine consensus* protocol.

1.4.3 Network models

The Byzantine consensus problem is studied in both synchronous [74, 75] and asynchronous [76–79] networks. In the synchronous network all the nodes share a common clock and the delays in the network communication are deterministic. In the asynchronous network there are no shared clocks and the transmission delays might be arbitrarily long for each message. In this model the only guarantee is that a message sent from a correct node to another correct node will eventually arrive at the destination. For both synchronous and asynchronous networks, it is known that Byzantine consensus is impossible in an n node network if there are more than $t < n/3$ faulty nodes [80]. For asynchronous network it is known that no deterministic protocol can guarantee the protocol termination in the presence of fail-stop of Byzantine faulty nodes [81]. However, there exists finite expected time protocols [76–79] for asynchronous Byzantine consensus.

1.5 Outline

The rest of this thesis is organised as follows.

In Chapter 2, we formalise the concept of spatial and temporal reference frames. We define the general reference frame agreement problem and discuss the known results about two party reference frame agreement protocols. Then we discuss the problems that faulty nodes and imperfect communication introduces in an $n > 2$ node reference frame agreement protocol. We discuss a two party direction estimation protocol that was originally proposed by Massar and Popescu [63] and analyse it under the effects of a depolarising channel. This protocol is later used as an example two party primitive that we use to design our fault tolerant multiparty reference frame agreement protocols. Finally, we discuss synchronous and asynchronous networks and explain why multiparty protocols for these two types of networks should be different.

In Chapter 3, we formalise the spatial reference frame agreement problem for an n node *synchronous* quantum network and give a fault tolerant protocol that achieves reference frame agreement in the presence of $t < n/3$ faulty nodes.

In Chapter 4, we formalise the reference frame agreement problem for an n node *asynchronous* quantum network and give a fault tolerant protocol that achieves reference frame agreement in the presence of $t < n/3$ faulty nodes.

In Chapter 5, we discuss the quantum network architecture where entanglement swapping is used to perform quantum routing. We discuss some existing results and give detailed description of various network primitives such as quantum repeaters, quantum routers and network graphs. We introduce the concept of routing graphs and motivate how it can help to achieve efficient quantum routing. We also briefly discuss global, local and circuit routing modes.

In Chapter 6, we give the first high-level protocol that runs on an entanglement swapping based quantum network. We consider a network graph where all the nodes are connected in a cycle and give a routing graph construction that facilitates efficient local routing and requires a quantum memory of size logarithmic in the number of nodes.

This thesis ends with Chapter 7 in a short conclusion and outlook.

2

Reference frame agreement

In this chapter we introduce the reference frame agreement problem for a quantum network. We discuss what we know about these problems in the bipartite setting and what are the new challenges faced by any protocol involving more than two parties. We discuss the differences of asynchronous and synchronous networks and why reference frame agreement protocols have to be different in these two settings. Through all these discussions we hope to gain the necessary insights and preliminary knowledge that would be essential for understanding the synchronous and asynchronous reference frame agreement protocols that we give in Chapter 3 and Chapter 4 respectively.

2.1 Reference frame

2.1.1 Spatial reference frame

A *spatial reference frame* defines a co-ordinate system in space. For example, in a Cartesian coordinate system once the Cartesian frame $(\vec{\boldsymbol{x}}, \vec{\boldsymbol{y}}, \vec{\boldsymbol{z}})$ is specified, any vector $v = \alpha\vec{\boldsymbol{x}} + \beta\vec{\boldsymbol{y}} + \gamma\vec{\boldsymbol{z}}$ can be represented as (α, β, γ) where α, β and γ are scalers. For two distant parties who only have the knowledge of their own local frame, it becomes necessary to establish a shared reference frame before they can successfully communicate spatial information (such as, location and orientation).

2.1.2 Temporal reference frame

Similar to spatial reference frames multiple parties might need to synchronise their clock rates and time differences. Once they have established it, we say that they share a *temporal reference frame* and they are synchronised in time. Any multiparty protocol or computation performed by systems that do not share a temporal reference frame are respectively called *asynchronous protocol* or *asynchronous computation*.

2.2 The reference frame agreement problem

In a quantum channel, the qubits are encoded in some physical degree of freedom. For example, the polarisation direction of a photon is often used to encode qubits [62]. This requires the sender and the receiver to agree on some set of orthonormal directions (the horizontal and the vertical) as their common spatial reference frame. Another example is the time-bin qubits [82], where both of the parties require synchronised clocks. That is, they must have a pre-agreed temporal reference frame. Efficient implementations of many quantum protocols (for example, [83]) require that the nodes share such common reference frames. This imposes some challenges because during the initialisation of a quantum network the pairwise channel delays might be unknown, clocks might be unsynchronised and spatial reference frames might be unaligned. However, there are known quantum protocols [83, 84] that allow two nodes to synchronise clocks if they begin with a shared spatial reference frame. That is, the general reference frame agreement problem can be thought of as a problem of aligning Cartesian reference frames. This can be attained if there exist protocols that allow participating nodes to agree on a *direction*.

Unlike in classical information theory, where information can be represented in bits, a spatial reference frame can only be transferred from scratch by exchanging systems that have an inherent sense of direction [85]. Examples of such systems are electron spin qubits [61] and photon polarisation qubits [62]. The receiver can extract direction information from these systems, for example, by performing tomography on them.

While preparing the direction, any sender node P_i knows the description of the direction as an unit vector $v_i = (\alpha_i, \beta_i, \gamma_i)$, which is the classical representation of the vector $\alpha_i\vec{\boldsymbol{x}}_i + \beta_i\vec{\boldsymbol{y}}_i + \gamma_i\vec{\boldsymbol{z}}_i$ represented in P_i's local Cartesian frame$(\vec{\boldsymbol{x}}_i, \vec{\boldsymbol{y}}_i, \vec{\boldsymbol{z}}_i)$. Once the quantum system, carrying that direction, arrives at a receiver P_j, the receiver constructs a representation of the direction in it's own local frame as v_j. Such an estimation procedure inevitably introduces some errors, even in correct transmissions. That is, depending on the precision of the instruments one can only expect to have $d(v_i, v_j) \leq \delta$ for some $\delta > 0$, where $d(v_i, v_j)$ is the Euclidian distance between v_i and v_j. However, this distance metric does not make sense as it is, because v_i and v_j are vector representations in two different local frames. Therefore, we must redefine our distance metric $d(.,.)$ where distance is computed by converting both vectors in the frame of the first argument. As a result $d(v_i, v_j)$ remains a valid distance measure even though P_i and P_j do not know each other's local frame. Note that, a node can only compute distance between two directions that are represented in its own local frame, a protocol for solving the reference frame agreement problem should make sure that no distance computation between vectors in two different local frame is needed to run the protocol. However, this modified Euclidian distance can be used in the analysis.

Since, there are inherent imperfections in the direction transmission and reception process, a direction agreement protocol for multiple nodes has to take this into consideration. Therefore, we define a multiparty direction agreement problem, which is equivalent to a reference frame agreement problem, as follows.

Definition 2. For $\eta > 0$, a set or nodes S in a network *η-agree* on a direction if each node $P_i \in S$ agrees on a direction v_i such that for any $P_j \in S$ we have $d(v_i, v_j) \leq \eta$.

That is, even though the nodes in S do not agree on the exact same (i.e. $\eta = 0$) direction, if η is very small then all their agreed directions are close.

2.3 The 2-party problem

When two nodes try to agree on a reference frame we call it a *bipartite* reference frame agreement problem, or a *direction estimation* problem. Barlett et al. [45]

presents a comprehensive survey on the problems of quantum computation between two parties without a shared reference frame and how to align reference frames between them using 'unspeakable' information [64]. The task of optimally aligning reference frames for two parties was first studied by Gisin and Popescu in [86] and later by Massar [87] where they use electron spin to encode direction. In [64] Peres and Scudo studies Cartesian frame alignment using quantum systems. Bagan *et al.* [88] studied aligning reference frames using quantum channels. A more recent work was done by Skotinotis and Gour [89] where they determine the quantum states and measurements that optimise accessible information in a phase reference alignment protocol. Chuang [83] gives a quantum algorithm for clock synchronisation. However, the algorithm assumes shared Cartesian reference frames among the parties.

2.4 Correct nodes and faulty nodes

In a 2-party protocol if any one of the nodes is faulty, then the protocol inevitably fails. However, in a multiparty setting with more than two nodes, if some of the nodes are faulty we still want the non-faulty nodes to closely agree on some direction. This poses various challenges that were absent in the bipartite setting. To understand and mitigate these challenges, we have to characterise the faulty and non-faulty nodes.

2.4.1 Correct nodes

A *correct node* or a *non-faulty node* in a network is a node that acts only according to the protocol specification. That is a correct node does not do anything extra to gain advantage over other nodes and does not deviate from the instructions given by the protocol. In classical cryptography such nodes are often referred as an *honest node* or *honest party*.

2.4.2 Faulty nodes

A *faulty node* in a network is a node that does not always act according to the protocol specification. To be more specific, a faulty node might,

Be non-responding. Nodes might not communicate at all due to breakdown.

Send wrong message. Nodes might send wrong messages due to some internal error (for example, failure in input-output modules).

Show correlated errors. Nodes might show correlated error because of similar manufacturing defect, or being position in a geographical location that is going through some natural disaster.

Be controlled by an adversary. Nodes might be controlled by an adversary who can read all the public transmissions in the network and adapt its strategy accordingly to thwart the network.

Such a model of faulty nodes was first introduced in the study of distributed computing by Lamport et. al. [73]. For some historic reasons a protocol that can operate in the presence of such faulty nodes is called *Byzantine fault tolerant* protocol. For our study of reference frame agreement problems we model the faulty nodes in this *Byzantine fault* model. A faulty node can successfully thwart a protocol only if its identity is unknown. Otherwise, the correct nodes can leave them out of the protocol. Therefore we assume the faulty nodes do perform any action that exposes themselves. The idea behind assuming such strong faults is that any protocol that works in the presence of such faulty nodes would be robust in most fault situations that naturally occur, or are caused by an adversary.

2.5 The multiparty problem

When $n > 2$ nodes try to agree on a reference frame, we call it a *n-party reference frame agreement problem* or *multiparty reference frame agreement problem*.

Given that some of the nodes among these n are faulty we would want only the correct nodes to be able to agree on a reference frame. A reference frame agreement protocol that achieves this, can be defined as:

Definition 3. For $\eta > 0$, a *η-reference frame agreement* protocol among n nodes is a protocol such that,

Termination. Each correct node P_i terminates the protocol, and outputs a reference frame v_i.

Consistency. For all pairs of correct nodes P_i and P_j we have $d(v_i, v_j) \leq \eta$.

That is, no matter what the faulty nodes does an *η-reference frame agreement* protocol should be able to get all the correct nodes to η-agree on a direction. Since, initially the faulty nodes are indistinguishable from the correct nodes, this is a highly non trivial task. To see what challenges the faulty nodes might cause, let us consider a naive protocol where one node fixes an arbitrary reference frame, and communicates it to all the other nodes using some two-party direction estimation protocol. However, in a networked setting there might be faulty nodes and a faulty sender might send different reference frames to different nodes. Therefore, the receivers need to verify that they are indeed receiving the same (η close to each other) reference frame. However, some of the receivers might also be faulty and end-up confusing others by sending wrong messages during any such verification. A reference frame agreement protocol must account for this. Recall that, any protocol that overcomes these problems caused by the faulty nodes is said to be a *Byzantine fault tolerant* protocol (Section 2.4.2).

2.6 2-party estimate direction protocol (2ED)

From our discussions in Section 2.2 we know that directions cannot be exchanged perfectly. A receiver has to *estimate* the direction sent by a sender. A protocol that achieves this is formally defined as,

Definition 4. For $\delta > 0$ a *δ-estimate direction* protocol is a two-party protocol where one node (the sender) sends a direction u to the other node (the receiver). Upon termination the receiver gets a δ-approximation v of u, that is, $d(u, v) \leq \delta$.

Now, we show an example of a 2-party direction estimation protocol that was first proposed by Massar and Popescu in [63]. This protocol, which we refer to as 2ED, uses quantum communication to send a direction from a sender to a receiver. It is one of the simplest possible protocols where a sender creates many identical qubits with their Bloch vector pointing to the intended direction and

the receiver measures them with Pauli measurements. From the statistics of the measurement outcomes, the receiver then estimates the Bloch vector's direction closely with high success probability. We use this protocol, since it has some experimental advantages for implementation: it does not require any quantum memory or creation of entangled states, and it succeeds even if the quantum channel has a depolarising noise. However, the downside of this choice is that this protocol is not optimal in the number of qubits sent to achieve a certain accuracy. Optimal protocols can align frames in the so-called Heisenberg limit, they have a quadratic gain over 2ED [90].

We formally write the protocol in Protocol 1:2ED.

Protocol 1: 2ED

input : Sender, direction u

output : Receiver, direction v

Sender: 2ED-Send

- Prepare $3n$ qubits with direction u
- Send them to the receiver

Receiver: 2ED-Receive

- Receive $3n$ qubits from the sender
- Measure n qubits with σ_x and compute p_x, the frequency of getting outcome $+1$
- Similarly on the remaining qubits, compute p_y and p_z with measurements σ_y and σ_z on n qubits each
- Assign $x \leftarrow 2p_x - 1$, $y \leftarrow 2p_y - 1$, $z \leftarrow 2p_z - 1$; Assign $l \leftarrow \sqrt{x^2 + y^2 + z^2}$
- Output $v \leftarrow (x/l, y/l, z/l)$

We analyse the properties of the protocol under depolarising noise in the following theorem.

Theorem 1. *For all $\delta > 0$, using a depolarizing channel $\rho \mapsto (1-\varepsilon)\rho + \varepsilon\mathbb{I}/2$ between the sender and the receiver, protocol 2ED provides to the receiver a $(1-\varepsilon)\delta + \frac{5\varepsilon}{2}$ approximation of the sender's direction. It succeeds with probability $q_{\text{succ}} \geq 1 - e^{-\Omega(\delta^2 n)}$.*

Proof. We prove this theorem in two steps. First, we consider the case when the communication channel is noise free ($\varepsilon = 0$), and then, we see how depolarizing noise affects the approximation factor.

In the noise-free case, let us fix $\delta > 0$ and denote θ_x, θ_y, and θ_z to be the angles between u and the x-, y-, and z-axis of the local frame of the receiver. Therefore, $\cos^2 \frac{\theta_x}{2}$ is the probability of getting outcome $+1$ after the Pauli measurement σ_x on a qubit. Similarly, $\cos^2 \frac{\theta_y}{2}$ and $\cos^2 \frac{\theta_z}{2}$ are the probabilities for outcome $+1$ on measurement σ_y and σ_z respectively.

Now, we show that each of the following three conditions

$$|p_x - \cos^2 \frac{\theta_x}{2}| \leq \delta/5, \tag{2.1}$$

$$|p_y - \cos^2 \frac{\theta_y}{2}| \leq \delta/5, \tag{2.2}$$

$$|p_z - \cos^2 \frac{\theta_z}{2}| \leq \delta/5, \tag{2.3}$$

holds with probability at least $(1-2e^{-\frac{2}{25}n\delta^2})$, and later show that Equations (2.1), (2.2), and (2.3) imply that $d(u, v) \leq \delta$.

We know in the ideal case, when $n \to \infty$ the relative frequency $p_x \to \cos^2 \frac{\theta_x}{2}$. However, in 2ED n is finite. Therefore, we use the Hoeffding's inequality [91] to estimate the probability with which Conditions (2.1), (2.2) and (2.1) are satisfied.

For this, note that each of the Pauli measurements σ_x performed on a single qubit is an independent event. We can denote random variables Z_j where,

$$Z_j := \begin{cases} 1 & \text{if } j\text{th measurement outcome was } +1, \\ 0 & \text{if } j\text{th measurement outcome was } -1, \end{cases} \tag{2.4}$$

Using this we define a random variable $\bar{Z} := \sum_j Z_j$. Note that, $\bar{Z}$ counts the number of times an outcome $+1$ is observed after a Pauli measurement σ_x is performed on a qubit. Therefore, we have,

$$p_x = \frac{\bar{Z}}{n}, \tag{2.5}$$

and

$$\cos^2\frac{\theta_x}{2} = \frac{\mathbf{E}(\bar{Z})}{n}, \tag{2.6}$$

where $\mathbf{E}(Z)$ is the expectation value of $\bar{Z}$. Now, the Hoeffding's inequality tells us, that with high probability, the value of $\bar{Z}$ is not far from its expectation value. That is for all $t > 0$ we have,

$$\Pr\left(\left|\bar{Z} - \mathbf{E}(\bar{Z})\right| > tn\right) \leq 2\exp\left(-2t^2 n\right), \tag{2.7}$$

Taking $t = \delta/5$ in Inequality (2.7), with Equation (2.5) and (2.6) gives us,

$$\Pr\left(\left|p_x - \cos^2\frac{\theta_x}{2}\right| > \frac{\delta}{5}\right) \leq 2\exp\left(-\frac{2n\delta^2}{25}\right). \tag{2.8}$$

The same analysis holds true for Pauli measurements σ_y and σ_z. Hence Conditions (2.1), (2.2), and (2.3) are all satisfied with probability at least $\left(1 - 2e^{\left(-2n\delta^2/25\right)}\right)^3$.

That is the success probability,

$$q_{\text{succ}} \geq \left(1 - 2e^{\left(-2n\delta^2/25\right)}\right)^3, \tag{2.9}$$

$$\geq 1 - e^{-\Omega(\delta^2 n)}. \tag{2.10}$$

Where, the Equation (2.10) follows from the Bernoulli's inequality, which is $(1+x)^r > 1 + rx$ for all real $x \geq -1$ and integer $r \geq 2$.

Denoting the vector u in the receiver's basis by (x_u, y_u, z_u), we have

$$x_u = \cos\theta_x = 2\cos^2\frac{\theta_x}{2} - 1. \tag{2.11}$$

So,

$$|x - x_u| = \left|(2p_x - 1) - \left(2\cos^2\frac{\theta_x}{2} - 1\right)\right|, \tag{2.12}$$

$$= 2\left|p_x - \cos^2\frac{\theta_x}{2}\right|, \tag{2.13}$$

$$\leq\ 2\delta/5. \tag{2.14}$$

Here, Inequality (2.14) follows from Inequality (2.1). Similarly we have,

$$|y - y_u| \leq 2\delta/5 \quad \text{and} \quad |z - z_u| \leq 2\delta/5. \tag{2.15}$$

Using (2.14) and (2.15), we get,

$$\begin{aligned} d((x,y,z),u) &= \sqrt{(x-x_u)^2+(y-y_u)^2+(z-z_u)^2}, \\ &\leq \sqrt{(2\delta/5)^2+(2\delta/5)^2+(2\delta/5)^2}, \qquad (2.16) \\ &= \frac{2\sqrt{3}\delta}{5}. \qquad (2.17) \end{aligned}$$

This means that (x,y,z) is within a sphere of radius $\frac{2\sqrt{3}\delta}{5}$ centered in u, so its angle θ with u is at most $\arcsin(2\sqrt{3}\delta/5)$. Since v is the normalization of (x,y,z), its angle with u is also θ and from a simple trigonometric observation, we have,

$$d(u,v) = 2\sin(\theta/2) \leq 2\sin\left(\frac{1}{2}\arcsin(2\sqrt{3}\delta/5)\right). \qquad (2.18)$$

Moreover, one can check that for all $\alpha \in [0,1]$, $\sin\left(\frac{1}{2}\arcsin(\alpha)\right) \leq \frac{5}{4\sqrt{3}}\alpha$, thus,

$$d(u,v) \leq \delta. \qquad (2.19)$$

Effects of noise. So far we have considered only a noiseless channel, let us now turn to the case of a depolarizing channel: if the sender sends a pure state $|\psi\rangle$, then the receiver gets the mixed state

$$\rho = (1-\varepsilon)|\psi\rangle\langle\psi| + \varepsilon\frac{\mathbb{I}}{2}, \qquad (2.20)$$

where, $0 \leq \varepsilon \leq 1$.

From Equation (2.20) one can see that the effective relative frequency p_x is given by

$$p_x = (1-\varepsilon)p'_x + \frac{\varepsilon}{2}, \qquad (2.21)$$

where p'_x is the relative frequency that the receiver would have got if the channel was noise-free, meaning that $\left|p'_x - \cos^2\frac{\theta_x}{2}\right| \leq \delta/5$. Therefore,

$$\begin{aligned} |p_x - \cos^2\frac{\theta_x}{2}| &= |(1-\varepsilon)p'_x + \frac{\varepsilon}{2} - \cos^2\frac{\theta_x}{2}|, \qquad (2.22) \\ &\leq |(1-\varepsilon)\frac{\delta}{5} + \frac{\varepsilon}{2} - \varepsilon\cos^2\frac{\theta_x}{2}|, \qquad (2.23) \\ &\leq |(1-\varepsilon)\frac{\delta}{5} + \frac{\varepsilon}{2}|, \qquad (2.24) \\ &= (1-\varepsilon)\frac{\delta}{5} + \frac{\varepsilon}{2}. \qquad (2.25) \end{aligned}$$

Here Inequality (2.24) follows because $\varepsilon \cos^2(\theta_x/2)$ is positive.

The rest of the analysis remains the same as the noise-free case by replacing $\delta/5$ by $\arcsin(2\sqrt{3}\delta/5)$ in Equation (2.1).

□

2.7 Synchronous and Asynchronous networks

In the *Synchronous network* all the nodes share a common clock. That is their timing devices have same 'tick-rate' and each pair knows the pairwise time offset. This allows them to globally schedule and synchronise protocol steps performed by each node. If any node fails to send some message at any time, then the receiver node concludes that the node is faulty.

Whereas, in an asynchronous network no such global clocks are available. The nodes might take arbitrarily different time to perform different steps of any protocol. Furthermore, the message transmissions using the links between any two nodes might take an arbitrary time at each use of the link. The only guarantee is that if a correct node sends a message to another correct node, then the message *eventually* arrives at the destination. In this setting, if a receiver node is waiting for a message to arrive from a sender, then he cannot be certain whether the message is not arriving because the sender is faulty, or the message is taking a very long time in the network. Any protocol that waits for such a transmission event before taking next action might hang indefinitely. Therefore, most of the protocols for the synchronous networks are not applicable in asynchronous networks.

3

A synchronous protocol

In this chapter we give the first multiparty reference frame agreement protocol for a synchronous network. The chapter is organised as follows. In Section 3.1 we formalise the problem. In Section 3.2 we define the communication model, give the main result and present the protocols with their proof outlines. We give the detailed proofs in Section 3.3 and finally, we discuss open problems in relation to classical agreement protocols.

3.1 The problem

We want the correct nodes in an n node synchronous quantum network to be able to establish a common reference frame from scratch, even though t of them may be arbitrarily faulty. We assume a fully connected network graph. That is, every node is connected to every other node using both classical and quantum communication channels.

Using the modified Euclidian distance metric $d(.,.)$ specified in Chapter 2 Section 2.2, we formalise the problem by instantiating the general reference frame agreement problem (Definition 3) for synchronous network as the following.

Definition 5. For $\eta > 0$, an *η-synchronous reference frame consensus protocol* among n network nodes in a synchronous quantum network is a protocol such that

Termination. Each correct node P_i terminates the protocol, and outputs a reference frame v_i.

Consistency. For all pairs of correct nodes P_i and P_j we have $d(v_i, v_j) \leq \eta$.

Note that consistency does not require that all the correct nodes share the same reference frame ($\eta = 0$), but that each node has an approximation of it (η is small). This is important because any two-node protocol using only a finite number of rounds of communication cannot allow the two nodes to share a frame exactly.

3.2 The main result and protocol outlines

We present a protocol that allows all the correctly functioning nodes in an n node fully connected synchronous network to agree on a common spatial reference frame as long as not more than $t < n/3$ nodes are arbitrarily faulty. Our Protocol 2: RF-Consensus has the appealing feature that it can use any 2-party direction estimation protocol as a black box and lift it to a fault tolerant reference frame agreement protocol for n nodes. Such 2-party protocols are characterised by the accuracy δ (i.e., the two nodes δ-agree) and the success probability q_{succ} with which such an approximation guarantee is achieved. An example bipartite direction estimation protocol 2ED was given in Chapter 2.6. The protocol RF-Consensus is characterised in the following theorem.

Theorem 2. *For $\delta > 0$, in a complete network of n nodes that are pairwise connected by public authenticated quantum and classical channels, if a bipartite δ-estimate direction protocol that uses m qubits to achieve success probability $q_{\text{succ}} \geq 1 - e^{-\Omega(m\delta^2)}$ is used, then protocol RF-Consensus is a 30δ-synchronous reference frame consensus protocol with success probability at least $1 - e^{-\Omega(m\delta^2 - \log n)}$, that can tolerate up to $t < n/3$ faulty nodes.*

Our protocol is *efficient* as we need only a linear (in the number of nodes n) number of rounds of quantum communication. We also show that this setting is *robust* to noise on the channel connecting any two nodes. To give some examples of parameters, protocol RF-Consensus achieves accuracy $30\delta = 0.02$ with success

probability 99% in a network of $n = 10$ nodes with noiseless communication, if each node transmits $m \approx 3.1 \times 10^8$ qubits at each round.

Our protocol uses ideas of [92] which solves a simpler problem from classical distributed computing called Byzantine agreement [73], in particular we use classical consensus as a subroutine. This classical problem has been extensively studied using synchronous [74, 75] and asynchronous [76–79] classical communication, as well as quantum communication [93], also in a fail-stop model in which the faulty nodes can prevent the protocol from ever terminating [94]. There, the correct nodes should perfectly agree on a single classical bit. Recall that we cannot send a direction classically without a shared reference frame, and hence we cannot use such protocols. In addition, we face two extra challenges: First, we are dealing with a continuous set of outcomes; And second, it is impossible to transmit a direction perfectly using a finite amount of communication, even on an otherwise perfect channel. In quantum networks, furthermore, we also have errors on the communication channel, which are pretty much unavoidable in a regime where we cannot easily perform quantum error correction due to the lack of a common frame. In the Byzantine problem such errors would be attributed to faulty nodes, but in our setting this would mean that *all* nodes in the network are faulty and no protocol could ever hope to succeed. Here, we thus require a careful treatment of such approximation errors.

3.2.1 Model of communication

We assume that all the communication channels are

Public. Faulty nodes can adapt their strategy depending on the network traffic.

Authenticated. Faulty nodes cannot tamper with the channel connecting correct nodes and

Synchronous. Correct nodes know when they are supposed to receive a message, and if none is received, e.g. due to communication error, then the protocol continues which ensures that our protocol cannot stall indefinitely.

We only use quantum communications to send a direction between a sender and a receiver. As an example we use protocol 2ED (Chapter 2 Section 2.6). However in the protocol each direction might carry a classical 'tag' that describes what type of message it is.

3.2.2 Protocols and the proof synopsis

In this Section, we present a summary of our protocols and subprotocols, and an outline of their proof of correctness.

Our protocol works in two phases: First, a node is elected as the *king* P_k. Second, the king choses a direction w_k and sends it to all the other nodes. We denote w_i the direction received by the node P_i in its own frame. If the king is not faulty, then 2ED ensures that $d(w_i, w_k) \leq \delta$. Then the correct nodes should decide either all to accept this direction (they output $v_i \approx w_k$ in their respective own frame), or all to reject it (output $\perp$).

This second phase is known as *king consensus*. More formally, a king consensus protocol should satisfy two properties:

δ-persistency: if the king is not faulty, all the correct nodes P_i, should output v_i such that $d(v_i, w_k) \leq \delta$; and

η-consistency: All the correct nodes reach a consensus, that is, they either all output $\perp$, or they all output directions that are η-close to each other, i.e., for all correct nodes P_i and P_j, the distance $d(v_i, v_j) \leq \eta$.

We repeat those two phases with different kings as long as a consensus is not reached. In particular, the protocol will terminate after at most $t + 1$ rounds since there are at most t faulty nodes.

So the success of our RF-Consensus protocol depends on the success of the king consensus protocol, which is achieved in three steps.

Step 1: Weak Consensus We first create a weaker protocol than king consensus by relaxing the condition that the correct nodes either *all* output a direction, or *all* output $\perp$. In a weak consensus, *some* nodes can output $\perp$ and the other

Protocol 2: RF-Consensus

Input : None

Output : $\forall i$, P_i outputs direction v_i

```
for k = 1 to t + 1 do
    v_i = King-Consensus(P_k)
    if v_i ≠⊥ then
        Output v_i
```

a direction. However we keep the condition that if two correct nodes P_i and P_j output directions u_i and u_j, then they should be close to each other. Formally, we define a *weak consensus* protocol as a protocol with the following two properties: δ-*weak persistency*: if there exists a direction w_k such that for every correct node P_i, $d(w_i, w_k) \leq \delta$, then $d(u_i, w_k) \leq \delta$; and η-*weak consistency*: For every pair of correct nodes P_i and P_j that output $u_i \neq \perp$ and $u_j \neq \perp$ respectively, we have $d(u_i, u_j) \leq \eta$.

Protocol 3: Weak-Consensus

Input : $\forall i$, P_i inputs direction w_i

Output : $\forall i$, P_i outputs direction u_i or $\perp$

```
Send w_i to all other nodes
Receive a_i[j] ← direction received from P_j
Create the set S_i ← {P_j : d(w_i, a_i[j]) ≤ 3δ}
if |S_i| ≥ n − t then
    Assign u_i ← w_i
else
    Assign u_i ←⊥
Output u_i
```

Protocol 3: Weak-Consensus achieves δ-weak persistency and (8δ)-weak consistency with probability at least $q_{\text{succ}}^{n^2-n}$ where δ is the accuracy achieved with probability q_{succ} by the two-node protocol used to send directions.

Here, with probability at least $q_{\text{succ}}^{n^2-n}$, for every correct node P_i and P_j, $d(a_i[j], w_j) \leq \delta$. It is easy to see that this protocol is δ-weak persistent. We

sketch the proof of the weak consistency. Consider the sets S_i and S_j of two correct nodes P_i and P_j. If $u_i \neq \perp$ and $u_j \neq \perp$, then S_i and S_j contains at least one correct node in common, let us call it P_α. Thus, $d(u_i, u_j) \leq d(u_i, a_i[\alpha]) + d(a_i[\alpha], w_\alpha) + d(w_\alpha, a_j[\alpha]) + d(a_j[\alpha], u_j) \leq 3\delta + \delta + \delta + 3\delta = 8\delta$.

Step 2: Graded Consensus. In a king consensus protocol, the correct nodes should have a "global" behaviour, as they should all either output a direction or $\perp$, whereas in the weak consensus each node has a "local" strategy. A *graded consensus* protocol behaves intermediately. Alongside a direction $v_i \neq \perp$ the nodes also output a grade $g_i \in \{0, 1\}$ which carries a "global" property, namely, *η-graded consistency*: If *any* correct node outputs a grade 1, then the directions between *all* the correct nodes should be η-close to each other, that is, for every pair (P_i, P_j) of correct nodes, $d(v_i, v_j) \leq \eta$.

Protocol 4: Graded-Consensus achieves (30δ)-graded consistency. It succeeds with probability at least $q_{\text{succ}}^{n^2-n}$.

The main idea of Graded-Consensus is that the nodes which output $\perp$ in the weak consensus inform the other nodes (by sending the flags f_i's). The first consequence is that for all correct nodes P_α and P_β with $f_\alpha = f_\beta = 1$, $d(u_\alpha, u_\beta) \leq 8\delta$. The second consequence is that if a correct node has grade 1, then for all correct nodes P_i and P_j, the sets T_i and T_j each contains at least one correct node, let us denote them P_α and P_β. Thus, $d(v_i, u_\alpha) \leq d(v_i, a_i[\alpha]) + d(a_i[\alpha], u_\alpha) \leq 10\delta + \delta = 11\delta$. Finally, we get, $d(v_i, v_j) \leq d(v_i, u_k) + d(u_k, u_l) + d(u_l, v_j) \leq 11\delta + 8\delta + 11\delta = 30\delta$.

Step 3: King Consensus. We are ready to present the King-Consensus protocol (Protocol 4) that achieves δ-persistency and (30δ)-consistency. Our protocol uses Classical-Consensus as a subroutine. It solves a problem which is closely related to Byzantine agreement. Here, every node P_i starts with a bit g_i and outputs a bit y_i. All the correct nodes agree on a bit b, that is if P_i is correct, then $y_i = b$ where at least one of the correct nodes, P_j has input $g_j = b$. Classical consensus can be reached if there are $t < n/3$ faulty nodes, for an example of such protocol, see e.g. [69].

If the king is not faulty, then all the correct nodes will have grade $g_i = 1$. Hence the classical consensus will also be reached with value $y_i = 1$. So, all the

Protocol 4: Graded-Consensus

Input : $\forall i$, P_i inputs direction w_i

Output : $\forall i$, P_i outputs direction v_i and grade $g_i \in \{0, 1\}$

Run Weak-Consensus(w_i)

`// This initialises the variables` u_i `and` $a_i[j]$`'s`

if $u_i = \perp$ **then**

  Send flag $f_i = 0$ to all other nodes

else

  Send flag $f_i = 1$ to all other nodes

forall the *nodes* P_j **do**

  $f_i[j] \leftarrow$ Receive f_j

forall the *nodes* P_j *with* $f_i[j] = 1$ **do**

  Create set $T_i[j] \leftarrow \{P_k : f_i[k] = 1$, and $d(a_i[j], a_i[k]) \leq 10\delta\}$

Assign $l_i \leftarrow \arg\max\{|T_i[j]|\}$

if $f_i = 1$ **then**

  Assign $v_i \leftarrow w_i$

else

  Assign $v_i \leftarrow a_i[l_i]$

if $|T_i[l_i]| > n - t$ **then**

  Assign $g_i \leftarrow 1$

else

  Assign $g_i \leftarrow 0$

Output (v_i, g_i)

Protocol 5: King-Consensus

Input : Id of the king P_k.
Output : $\forall i$, P_i outputs direction v_i or $\perp$

if *I am the king* **then**
  Fix an arbitrary direction w_k
  Send w_k to all other nodes
else
  Receive $w_i \leftarrow$ direction received from the king
Assign $(v_i, g_i) \leftarrow$ Graded-Consensus(w_i)
Assign $y_i \leftarrow$ Classical-Consensus(g_i)
if $y_i = 1$ **then**
  Output v_i
else
  Output $\perp$

correct nodes will accept the direction shared by the king. If the king is faulty and yet the correct nodes reach a consensus with $y_i = 1$, then it means that at least one correct node had grade 1. In this case the (30δ)-graded consistency implies that $d(v_i, v_j) \leq 30\delta$ for all the correct nodes P_i and P_j. As a consequence, King-Consensus is (30δ)-consistent, and so is our main protocol RF-Consensus.

3.2.3 Resource requirements and performance

The Protocol 2 runs King-Consensus $t + 1$ times, which is $O(n)$. Each of the King-Consensus has O(1) steps. Therefore, the main protocol RF-Consensus performs $O(n)$ steps.

In the Protocol 3 each of the nodes sends $(n - 1)$ quantum messages to other nodes. Thus $n(n - 1)$ messages are generated. This protocol is played at most $O(n)$ times. Therefore $O(n^3)$ quantum messages are generated to achieve reference frame consensus.

The total number of classical messages exchanged in the protocol might vary depending on which implementation of Classsical-Consensus is used as a subroutine in the Protocol 5 King-Consensus. For example, if the protocol from [69] is used

then the Classical-Consensus requires $O(n^3)$ classical message. Since this is played $O(n)$ time in our main protocol, the total number of classical messages exchanged becomes $O(n^4)$. However, if we count the classical messages exchanged by our protocol without counting the messages in the classical subroutine, then the number is $O(n^3)$.

The messages that contains only classical data contains flag values of $O(1)$ length and origin-destination IDs which are $O(\log n)$ bits long. Thus the classical messages are of $O(\log n)$ size in bits.

The number of qubits exchanged in a quantum message is dictated by the success probability $1-e^{-\Omega(m\delta^2-\log n)}$. After fixing the node count n and the 2-party approximation error δ. This success probability term determines the number of qubits m exchanged in each quantum message.

3.3 Proof of correctness

Now, we give the formal definitions of all the sub-protocols and their correctness proofs. From the discussion of Section 3.2.2, we know that this would prove the correctness of the main result presented in Theorem 2.

3.3.1 Step 1: Weak Consensus

Let us start by giving a more formal definition of a weak consensus protocol.

Definition 6. A (δ, η)-weak consensus protocol is an n-node protocol, in which each node P_i has an input direction w_i and outputs either a direction u_i or $\perp$, that satisfies the following two properties:

δ-weak persistency If there exists a direction s such that for every correct node P_i, $d(s, w_i) \leq \delta$, then every correct node P_i outputs a direction u_i with $d(s, u_i) \leq \delta$.

η-weak consistency For every pair of correct nodes P_i and P_j who output $u_i \neq \perp$ and $u_j \neq \perp$ respectively, we have $d(u_i, u_j) \leq \eta$.

Theorem 3. *For $\delta > 0$, using a two-node δ-estimate direction protocol that succeeds with probability q_{succ}, the protocol Weak Consensus is a $(\delta, 8\delta)$-weak consensus*

protocol tolerant to $t < n/3$ faulty nodes that succeeds with probability at least $q_{\text{succ}}^{n^2-n}$.

Proof. After line 2 of Protocol 3: Weak Consensus, the property

$$\forall \text{ correct nodes } P_i, P_j, \quad d(a_i[j], w_j) \leq \delta, \tag{3.1}$$

holds with probability at least $q_{\text{succ}}^{n^2-n}$ since each of the n nodes uses 2ED $n-1$ times. The rest of the proof shows that Property (3.1) implies δ-weak persistency and 8δ-weak consistency. This means that Weak-Consensus succeeds with probability at least $q_{\text{succ}}^{n^2-n}$.

Weak persistency. We assume there exists a direction s such that the input w_i of every correct node P_i satisfies $d(s, w_i) \leq \delta$. Let P_i be a correct node. We now show that $d(s, u_i) \leq \delta$. The idea is to show that $|S_i| \geq n - t$, hence $d(s, u_i) = d(s, w_i) \leq \delta$. This is done by showing that every correct node is in the set S_i. Indeed, let us consider a correct node P_j, then by triangular inequality we get,

$$d(w_i, a_i[j]) \leq d(w_i, s) + d(s, w_j) + d(w_j, a_i[j]). \tag{3.2}$$

Each of the first two terms is at most δ by assumption, and the last one is also at most δ by Property (3.1). Thus,

$$d(w_i, a_i[j]) \leq 3\delta. \tag{3.3}$$

Since there are at least $(n - t)$ non faulty nodes, $|S_i| \geq (n - t)$. This completes the proof of the δ-weak persistency.

Weak consistency. Let us consider two correct nodes P_i and P_j which output $u_i \neq \perp$ and $u_j \neq \perp$ respectively. Now we show that $d(u_i, u_j) \leq 8\delta$. The idea is to show that there exists a direction w_α such that $d(u_i, w_\alpha) \leq 4\delta$ and $d(u_j, w_\alpha) \leq 4\delta$. This is done by first showing that there exists one correct node P_α in both sets S_i and S_j.

For that, let us define the sets C_i and C_j by,

$$C_i = \{P_l : P_l \in S_i \text{ and node } P_l \text{ is correct}\}, \tag{3.4}$$

$$C_j = \{P_l : P_l \in S_j \text{ and node } P_l \text{ is correct}\}. \tag{3.5}$$

We need to prove that $C_i \cap C_j \neq \emptyset$. We do it by contradiction: let us assume that

$$C_i \cap C_j = \emptyset. \tag{3.6}$$

Note that,

$$\begin{aligned} |S_j| \geq m - t \Rightarrow |S_j - C_j| + |C_j| &\geq n - t, && (3.7)\\ \Rightarrow t + |C_j| &\geq n - t, && (3.8)\\ \Rightarrow |C_j| &\geq n - 2t, && (3.9)\\ \Rightarrow |C_j| &> \frac{n}{3}. && (3.10) \end{aligned}$$

Inequality (3.8) follows because there can be at most t faulty nodes, and Inequality (3.10) since $t < n/3$. Now,

$$\begin{aligned} |S_i \cup S_j| &= |(S_i - C_i) \cup (S_j - C_j) \cup C_i \cup C_j|, && (3.11)\\ &= |(S_i - C_i) \cup (S_j - C_j)| + |C_i| + |C_j|, && (3.12)\\ &\geq |(S_i - C_i)| + |C_i| + |C_j|, && (3.13)\\ &= |(S_i - C_i) \cup C_i| + |C_j|, && (3.14)\\ &= |S_i| + |C_j|, && (3.15)\\ &\geq (n - t) + |C_j|, && (3.16)\\ &> n - \frac{n}{3} + \frac{n}{3}. && (3.17) \end{aligned}$$

Here, Equation (3.12) follows from Equation (3.6), and Inequality (3.17) from Inequality (3.10). We just proved that $|S_i \cup S_j| > n$ which contradicts the fact that there are exactly n nodes. So, we have $C_i \cap C_j \neq \emptyset$.

Consider a correct node $P_\alpha \in (C_i \cap C_j)$. We have:

$$\begin{aligned} d(u_i, w_\alpha) &= d(w_i, w_\alpha), && (3.18)\\ &\leq d(w_i, a_i[\alpha]) + d(a_i[\alpha], w_\alpha), && (3.19)\\ &\leq 3\delta + \delta. && (3.20) \end{aligned}$$

The factor 3δ comes from the fact that P_α is in S_i and the remaining δ since P_α is correct. We can do the same reasoning with the node P_j, hence we also have:

$$d(u_j, w_\alpha) \leq 4\delta. \tag{3.21}$$

By combining Equations (3.20) and (3.21), we prove the 8δ-weak consistency:

$$d(u_i, u_j) \leq d(u_i, w_k) + d(w_k, u_j) \leq 4\delta + 4\delta = 8\delta. \tag{3.22}$$

□

3.3.2 Step 2: Graded Consensus

Again, we shall start by giving a formal definition of a graded consensus protocol.

Definition 7. A (δ, η)-graded consensus protocol is an n-party protocol, in which each node P_i has an input direction w_i and outputs a direction v_i as well as a grade $g_i \in \{0, 1\}$, that satisfies the following properties:

δ-graded persistency If there exists a direction s such that for every correct node P_i, $d(s, w_i) \leq \delta$, then every correct node P_i outputs a direction v_i such that $d(s, v_i) \leq \delta$ and $g_i = 1$;

η-graded consistency If there exists a correct node P_c who outputs grade $g_c = 1$, then for all pairs (P_i, P_j) of correct nodes, $d(v_i, v_j) \leq \eta$.

From Line 2 to Line 7 of Protocol 4: Graded-Consensus, the nodes send and receive classical bits, there is no approximation here. An important consequence is that $f_i[j] = f_j$ whenever the nodes P_i and P_j are correct.

Theorem 4. *For $\delta > 0$, Consider that Weak Consensus uses a δ-estimate direction protocol that succeeds with probability q_{succ}. Protocol Graded Consensus is a $(\delta, 30\delta)$-graded consensus protocol tolerant to $t < n/3$ faulty nodes that succeeds with probability at least $q_{\text{succ}}^{n^2-n}$.*

Proof. Similarly to the Weak Consensus protocol, with probability at least $q_{\text{succ}}^{n^2-n}$, the following property holds:

$$\forall \text{ correct nodes } P_i, P_j, \quad d(a_i[j], w_j) \leq \delta. \tag{3.23}$$

Graded persistency. We assume there exists a direction s such that, for each correct node P_i, $d(s, w_i) \leq \delta$. We first show that every correct node P_i outputs grade $g_i = 1$, and then show their output v_i satisfies $d(s, v_i) \leq \delta$.

Let us consider a correct node P_i. It outputs $g_i = 1$ if and only if $|T_i[l_i]| \geq n-t$. To show that the later condition holds, we first show that for each of the $(n-t)$ correct nodes P_j we $|T_i[j]| \geq n - t$. Therefore, by definition of l_i, we have $|T_i[l_i]| \geq |T_i[j]| \geq n - t$. This is proved by showing that for every correct nodes P_α, we have $d(a_i[j], a_i[\alpha]) \leq 4\delta$, that is, every correct node $P_\alpha \in T_i[j]$.

Since the nodes P_j and P_α are both correct, and Weak Consensus is δ-weak persistent, we know that $u_j \neq \perp$, $u_\alpha \neq \perp$ with

$$d(s, u_j) \leq \delta \quad \text{and} \quad d(s, u_\alpha) \leq \delta. \tag{3.24}$$

As a consequence $f_i[j] = f_i[\alpha] = 1$. We also know that $a_i[j]$ and $a_i[\alpha]$ are δ-approximations of u_j and u_α respectively, that is,

$$d(a_i[j], u_j) \leq \delta \quad \text{and} \quad d(a_i[\alpha], u_\alpha) \leq \delta. \tag{3.25}$$

Using the triangular inequality again with the Inequalities (3.24) and (3.25), we get,

$$\begin{aligned} d(a_i[j], a_i[\alpha]) &\leq d(a_i[j], u_j) + d(u_j, s) \\ &\quad + d(s, u_\alpha) + d(u_\alpha, a_i[\alpha]), && (3.26) \\ &\leq 4\delta. && (3.27) \end{aligned}$$

Since $f_i[j] = 1$, the set $T_i[j]$ exists, and since $f_i[\alpha] = 1$ and $d(a_i[j], a_i[\alpha]) \leq 4\delta \leq 10\delta$, $P_\alpha \in T_i[j]$. This proves that $g_i = 1$.

Now, let us show that $d(s, v_i) \leq \delta$. By δ-weak persistency, we know that $u_i \neq \perp$, therefore, $f_i = 1$. In this case, Line 12 assigns $v_i \leftarrow w_i$. As a direct consequence, we get, $d(s, v_i) = d(s, w_i) \leq \delta$. This concludes the proof of the δ-graded persistency.

Graded consistency. Let us assume that there exists a correct node P_c that outputs grade 1. In this case we show that for any two correct nodes P_i and P_j, the distance $d(v_i, v_j) \leq 30\delta$.

This proof is in three steps. First, we will show that all the correct nodes who are in the sets created at Line 9 are close to each other. More precisely, we

will show that for all the correct nodes P_α and P_β with $f_\alpha = f_\beta = 1$, we have $d(u_\alpha, u_\beta) \leq 8\delta$. The second step shows that v_i and v_j are 11δ-close to some u_α and u_β respectively where P_α and P_β are correct nodes with $f_\alpha = f_\beta = 1$. The last step combines this two facts to conclude the proof.

Step 1) This first step is a consequence of the 8δ-weak consistency of the Weak Consensus protocol used at Line 1. Indeed, consider two correct nodes P_α and P_β such that $f_\alpha = f_\beta = 1$. This means that $u_\alpha \neq \perp$ and $u_\beta \neq \perp$, hence they satisfy

$$d(u_\alpha, u_\beta) \leq 8\delta. \tag{3.28}$$

Step 2) We now prove that there exists a correct node P_α such that $d(v_i, u_\alpha) \leq 11\delta$. There are two cases to consider here. First $f_i = 1$: in this case, the correct node P_i outputs $v_i = u_i$, thus $d(v_i, u_i) = 0 \leq 11\delta$. The more interesting case is $f_i = 0$. We are going to show that in this case, there exists a correct node $P_\alpha \in T_i[l_i]$. This is done by showing that the number of nodes in the set $T_i[l_i]$ is more than the number of faulty nodes, that is, $|T_i[l_i]| > n/3$. In a similar way to the graded persistency, we will in fact prove that for every correct node P_k with $f_k = 1$, $|T_i[k]| > n/3$, hence $|T_i[l_i]| \geq |T_i[k]| \geq n/3$.

Let us then consider a correct node P_k with $f_k = 1$. By Equation (3.28), we have $d(u_k, u_{k'}) \leq 8\delta$ for every correct node $P_{k'}$ with $f_{k'} = 1$. As a consequence, we also have

$$\begin{aligned} d(a_i[k], a_i[k']) &\leq d(a_i[k], u_k) + d(u_k, u_{k'}) + d(u_{k'}, a_i[k']), \\ &\leq \delta + 8\delta + \delta. \end{aligned} \tag{3.29}$$

This with Line 9 implies that the set $T_i[k]$ contains every correct node $P_{k'}$ with $f_{k'} = 1$. Let us argue that there are more than $n/3$ such correct nodes. Recall that we have assumed that the correct node P_c has outputted grade $g_c = 1$. We thus have $|T_c[l_c]| > (n - t)$. We also know that there are at most $t < n/3$ faulty nodes. So, there must be at least $n - 2t > n/3$ correct nodes in $T_c[l_c]$, that is, there are more than $n/3$ correct nodes $P_{k'}$ with $f_{k'} = 1$.

We just proved that there exists at least one correct node P_α in $T_i[l_i]$, therefore,

$$d(v_i, u_\alpha) = d(a_i[l_i], u_\alpha), \tag{3.30}$$

$$\leq d(a_i[l_i], a_i[\alpha]) + d(a_i[\alpha], u_\alpha), \tag{3.31}$$

$$\leq 10\delta + \delta. \tag{3.32}$$

Using similar arguments, there exists at least one correct node P_β such that

$$d(v_j, u_\beta) \leq 11\delta. \tag{3.33}$$

Step 3) Now using triangular inequality with Inequalities (3.32), (3.28), and (3.33) we get,

$$\begin{aligned} d(v_i, v_j) &\leq d(v_i, u_\alpha) + d(u_\alpha, u_\beta) + d(u_\beta, v_j), & (3.34)\\ &\leq 11\delta + 8\delta + 11\delta. & (3.35) \end{aligned}$$

This proves the (30δ)-graded consistency of the protocol. □

3.3.3 Step 3: King Consensus

Definition 8. A (δ, η)-king consensus protocol is an n-node protocol in which one node P_k, called the king, choses a direction w_k and each of the other nodes P_i outputs either a direction v_i or each of them outputs $\perp$, which satisfies the following two properties:

δ-persistency If the king is correct, then all the correct nodes P_i output $v_i \neq \perp$ with $d(w_k, v_i) \leq \delta$.

η-consistency All correct nodes reach a consensus, that is, they either all output $\perp$, or they all output directions that are η-close to each other, i.e., for all correct nodes P_i and P_j, the distance $d(v_i, v_j) \leq \eta$.

Our protocol for solving the king consensus problem uses Graded-Consensus and Classical-Consensus as a subroutines. This classical protocol was introduced in the Chapter 1.4 Definition 1 (for a protocol see, e.g., [69]).

Theorem 5. *For $\delta > 0$, using a δ-estimate direction protocol that succeeds with probability q_{succ}, King-Consensus is a $(\delta, 30\delta)$-king consensus protocol that succeeds with probability at least $q_{\text{succ}}^{n^2}$.*

Proof.

Persistency. Let us assume that the king is correct. We want to show that every correct node P_i outputs $v_i \neq \perp$ with $d(w_k, v_i) \leq \delta$. Since the king is non faulty, with probability at least q_{succ}^n, we have that for all correct players P_i, the distance $d(w_k, w_i) \leq \delta$.

From the δ-graded persistency of Graded-Consensus used in Line 6, we know that for all correct nodes P_i, $d(v_i, w_k) \leq \delta$ and $g_i = 1$ with success proability at least $q_{\text{succ}}^{n^2-n}$; And from the validity of Classical-Consensus, we have that $y_i = 1$ for all correct nodes P_i. Hence all the correct nodes output a δ-approximation of w_k with probability at least $q_{\text{succ}}^{n^2}$.

Consistency. To prove consistency we will show that all the correct nodes output $\perp$, or they all output a direction. In this case we also have to show that for every pair (P_i, P_j) of correct nodes, $d(v_i, v_j) \leq 30\delta$.

Since the variables y_i are outputs of Classical-Consensus, the agreement property ensures that there exists a bit b such that for all the correct nodes P_i, $y_i = b$.

If $b = 0$, then all the correct nodes output $\perp$.

If $b = 1$, then by validity of Classical-Consensus, at least one of the correct nodes, let us denote it by P_i, has flag $g_i = 1$. Recall that the (30δ)-graded consistency of Protocol 4:Graded-Consensus says that we have in this case $d(v_i, v_j) \leq 30\delta$ for every correct nodes P_i and P_j. □

Finally we need to show that the over all success probability of RF-Consensus scales as $1 - e^{-\Omega(m\delta^2 - \log n)}$ where n is the number of nodes and δ and m are parameter specified by the 2-party direction estimation protocol used. For example, for Protocol 1: 2ED δ and m represents the approximation accuracy and number or qubits used respectively. The following lemma shows that,

Lemma 1. *For $\delta > 0$, $n, m \in \mathbb{N}$ and $n > 2$, if $q_{\text{succ}} \geq 1 - e^{-\Omega(m\delta^2)}$ then $q_{\text{succ}}^{n^2} \geq 1 - e^{-\Omega(m\delta^2 - \log n)}$.*

Proof. We show this using Bernoulli's inequality, which is, $(1 + x)^r > 1 + rx$ for all real $x \geq -1$ and integer $r \geq 2$. Using this we have,

$$q_{\text{succ}}^{n^2} \geq (1 - e^{-\Omega(m\delta^2)})^{n^2}, \tag{3.36}$$
$$\geq 1 - n^2 e^{-\Omega(m\delta^2)}, \tag{3.37}$$
$$\geq 1 - e^{-\Omega(m\delta^2 - \log n)}. \tag{3.38}$$

□

From Theorem 5 and Lemma 1 we see that the success probability of the King-Consensus protocol is at least $1 - e^{-\Omega(m\delta^2 - \log n)}$. However, the main protocol RF-Consensus in the worst case playes King-Consensus at most $t + 1 \leq n/3$ times. Using Bernoulli's inequality one can see that the overall success probability of the RF-Consensus remains at least $1 - e^{-\Omega(m\delta^2 - \log n)}$. This observation with Theorem 5 proves Theorem 2 which is the main result.

3.4 Discussion

We have presented the first protocol for spatial reference frame agreement in a synchronous quantum network. Even in the classical setting, the algorithms to solve the Byzantine agreement problem, where multiple nodes try to agree on a bit, is surprisingly complicated. It remains open if simpler and more efficient protocols could be designed for our setting, possibly by using entangled states. It is an interesting question to construct protocols that also work in an asynchronous communication model. The latter is already challenging for the classical case [76–79]. This is the topic of the next chapter. Another interesting question is whether more faulty nodes than $t < n/3$ can be tolerated. If our protocol were to succeed with probability 1 and η were sufficiently small, then we can prove that it is optimal in that sense by adapting the classical proof [80] to our setting. However, for aligning reference frames, any protocol can only succeed with probability strictly less than 1. This problem has been partially studied in the classical case [95]. Even in the constant error scenario the optimal number of faulty nodes that can be tolerated is not known for the classical Byzantine agreement problem [96]. This leaves hope to find protocols that can tolerate $t < n/2$ faulty nodes when allowing constant success probability both for Byzantine and reference frame agreement.

4

An asynchronous protocol

In this chapter we give the first multiparty reference frame agreement protocol for an asynchronous network. The chapter is organised as follows: we first formalise the problem. Then, in Section 4.2 we define the communication model, give the main result and present the protocol with proof outlines. We give the detailed proofs in Section 4.3.1. Finally, we conclude the chapter with a discussion of related classical works and open problems.

4.1 The problem

An efficient implementation of many multiparty protocols for quantum networks requires that all the nodes in the network share a common reference frame. Establishing such a reference frame from scratch is especially challenging in an asynchronous network where network links might have arbitrary delays and the nodes do not share synchronised clocks. Here, we study the problem of establishing a common reference frame in an asynchronous network of n nodes of which at most t are affected by arbitrary unknown error, and the identities of the faulty nodes are not known. We present a protocol that allows all the correctly functioning nodes to agree on a common reference frame as long as the network graph is complete and not more than $t < n/4$ nodes are faulty. As the protocol is asynchronous, it can be used with some assumptions to synchronise clocks over a network. Also, the protocol has the appealing property that it allows any existing two-node

asynchronous protocol for reference frame agreement to be lifted to a robust protocol for an asynchronous quantum network.

We formalise the problem using the modified Euclidian distance metric $d(.,.)$ specified in Chapter 2 Section 2.2.

Definition 9. For $\eta > 0$, a protocol in an asynchronous network of n nodes is an *η-asynchronous reference frame agreement protocol* if it satisfies the following conditions.

Termination. Every correct node P_i eventually terminates and outputs a direction v_i.

Correctness. If correct node P_i outputs v_i and correct node P_j outputs v_j then $d(v_i, v_j) \leq \eta$.

However, we have to achieve these termination and correctness conditions in the presence of incorrect or faulty nodes.

4.2 The main result and protocol outlines

We give a protocol that can take any 2-party asynchronous direction estimation protocol and lift it up to a fault tolerant multiparty asynchronous reference frame agreement protocol. More specifically, we present the first Protocol A-Agree which allows n nodes in a fully connected asynchronous quantum network to agree on a reference frame in the presence of $t < n/4$ faulty nodes. The result can be summarised in the following theorem.

Theorem 6. *For $\delta > 0$, in a complete network of n nodes that are pairwise connected by public authenticated quantum and classical channels, if a bipartite δ-estimate direction protocol that uses m qubits to achieve success probability $q_{\text{succ}} \geq 1 - e^{-\Omega(m\delta^2)}$ is used, then Protocol A-Agree is a 42δ-asynchronous reference frame agreement protocol with success probability at least $1 - e^{-\Omega(m\delta^2 - \log n)}$, that can tolerate up to $t < n/4$ faulty nodes.*

We have discussed in Chapter 2 classical Byzantine agreement protocols for asynchronous networks cannot solve the problem of reference frame agreement on a asynchronous quantum network. However, classical literature can still inform us

on important questions such as, how to achieve constant expected time, how to handle asynchronicity. Some of the approaches of our protocol regarding these questions are influenced by [79]. We also use the interactive consistency protocol by Ben-Or et al. [84] as a subroutine (Section 4.2.2.4).

4.2.1 Communication model

We achieve the result in Theorem 6 under the following assumptions about the communication channels and faulty nodes. For quantum networks our assumptions are,

- The pairwise channels are *public*. That is, the messages are not secret. As a result, an adversary can see the content of a message between two correct nodes and adapt its strategy accordingly.
- The pairwise channels are authenticated. That is, if a correct node sends a message to another correct node, then the message cannot be altered by any adversary. However, there might be channel noises, which can be dealt with, as in (Chapter 2 Section 2.6).
- The pairwise channel delays might be controlled by the faulty nodes. That is, the faulty nodes can control the channel delays, even the delays for message passing between any pair of correct nodes.
- If a correct node sends a message to another correct node, then the message eventually reaches the receiver. That is, even though the delay is controlled by some adversaries they cannot put infinite delay on the message between two correct nodes. However, the delay can be arbitrarily large.
- The faulty nodes might have correlated error. To create a protocol which tolerates the worst kind of faults, we also assume that the faulty nodes can cooperate with each other and have a global strategy to thwart the protocol. This is a realistic assumption because some nodes in a region might show correlated error which affects a part of the network.

4.2.2 Preliminaries

The problem of reference frame agreement over an asynchronous quantum network is necessarily multidisciplinary in nature. That is, it combines various concepts from quantum physics, information theory, cryptography and distributed computing. In this section we introduce several concepts from these fields that will be useful throughout this chapter.

4.2.2.1 Asynchronous communication

In an asynchronous network we assume that the nodes do not share any synchronised clock and the communication channel between each pair is such that a message takes an arbitrary amount of time to propagate through it. Here the only guarantee is, if a message is transmitted from a correct node then the message will eventually reach to the receiver. Also, a node might take an arbitrary amount of time to perform the next step in a protocol. Therefore, the execution time of an asynchronous protocol should be carefully defined.

4.2.2.2 Asynchronous time

We briefly present a standard definition of the running time of an asynchronous protocol. For more general definitions of asynchronous time see, for instance, [79, 97, 98].

Imagine a 'global clock' is measuring time in the network. This is a virtual clock, so the network nodes cannot read it. Time elapsed from the sending of a message to its reception is denoted the *delay* of the message. Let *period* be the longest delay in a finite execution of a protocol. The *running time* of the finite execution of the asynchronous protocol is the total time measured by the global clock, until the protocol ends, divided by the period of this execution. If the protocol never terminates, then the running time is infinite.

More formally, consider an execution of an asynchronous protocol. All the nodes executing the protocol should perform a sequence of steps. At the very first step some *init* messages were generated by a node to start the protocol. Let s_0 denote this step. We define the *round* 0 to contain only this step. Also, we call any message generated at round i an *i-message*. (For example, all the *init*

messages that are sent to different nodes are 0-messages.) Now, for each $i > 0$, let s_i be the last step where an $(i-1)$-message is received by some node. All the steps after step s_{i-1} until (including) step s_i are defined to be in round i. The *running time* of the execution is the round number of the final step.

The performance, in terms of execution time, of an asynchronous agreement protocol is determined by its expected running time. The expectation is thereby taken over all possible random inputs of the nodes, random bits used by the nodes, as well as all possible random behaviour of the faulty nodes. The exact probability distributions may not be known, but the goal is to show that the expected running time is low for all possible distributions.

4.2.2.3 Asynchronous message

In the absence of a synchronised clock, each message must have a 'begin' and 'end' tag. Also, depending on the particular application, a message might carry a [type] tag. In our problem we don't have a shared reference frame. For this reason, we cannot use the quantum channel to carry these [type] tags. This requires us to have a parallel classical channel that uses some classical degree of freedom to carry bits.

We assume that each pair of nodes are connected by an asynchronous public authenticated CQ-channel (classical quantum channel), which can send a message using both classical and quantum degrees of freedom in the absence of a shared reference frame. An example of such combined message is shown in Table 4.1 where each quantum message m_q is sandwiched between a classical 'begin' and an 'end' tag and also accompanied by a classical type tag m_c. The symbol $\perp$ denotes quantum signals that can be ignored.

Table 4.1: Channel primitive: **A message**

Step	Classical	Quantum
1	begin	$\perp$
2	m_c	m_q
3	end	$\perp$

The only assumption is the nodes can match the classical and quantum parts of the message.

4.2.2.4 Asynchronous interactive consistency

Our protocol uses the solution to the following interactive consistency problem which was first proposed by Pease, Shostak and Lamport [69].

Definition 10 (The Interactive Consistency Problem)**.** Consider a complete network of n nodes in which communication lines are private. Among the n nodes up to t might be faulty. Let $P_1, P_2, \ldots, P_n$ denote the nodes. Suppose that each node P_i has some private value of information $V_i \in |V| \geq 2$. The question is whether it is possible to devise a protocol that, given $n, t \geq 0$, will allow each correct node to compute a vector of values with an element for each of the n processors, such that:

1. All the correct nodes compute exactly the same vector;

2. The element of this vector corresponding to a given correct node is the private value of that node.

For an asynchronous network, Ben-Or and El-Yaniv [84] gives a Protocol Asynchronous-IC that solves this problem for $t < n/3$ in constant expected time. We use this protocol as a subroutine.

Note that the Asynchronous-IC requires private asynchronous classical channels. Whereas, we only require public authenticated classical and quantum channels between each pair of nodes in the network. The reason is, with authenticated public quantum channels each pair of nodes can play 2ED type protocol and establish a bipartite reference frame. Once the bipartite reference frame is established between each pair using the public authenticated classical and quantum channels, they can perform QKD that gives them a private classical channel. Therefore, they can play Asynchronous-IC at a later stage of the protocol.

4.2.3 Protocols and the proof synopsis

Now we will give the protocols and their proof synopsis. The formal proofs are given in Section 4.3.1 However, we first need to define some notations.

$w_i[j]$ represents a vector received by node P_i from node P_j using the bipartite direction estimation protocol. This vector is represented with respect to P_i's local reference frame.

In our protocol sending (type, v) to some node means the sender uses a δ-*estimate direction protocol* to send the direction v to the receiver. The sender also sends the classical tag [type] associated to this direction. The receiver will receive an approximation of the sent direction as v' where $d(v, v') \leq \delta$. Our protocol uses four different tags as types. They are, init, echo, ready_1 and ready_2.

Next we fix a notation for a cluster of vectors of certain types where the cluster has a certain cluster centre and a cluster parameter. We write it as $C_i^\delta([\text{types}], w_c)$. This means the cluster with cluster centre w_c is computed and stored by node P_i, has a cluster parameter δ and contains only the vectors with associated tags in [types]. Here [types] is a comma separated list of [type]s. The cluster parameter δ denotes that for all $u, v \in C_i^\delta([\text{types}], w_c)$ their distance $d(u, v) \leq \delta$.

For example, $C_i^\delta([\text{ready}_1,\text{ready}_2], v_c)$ denotes a cluster in which each vector has tags ready_1 or ready_2 with cluster centre v_c such that $\forall u, v \in C_i^\delta([\text{ready}_1,\text{ready}_2], v_c)$, and $d(u, v) \leq \delta$. We say that this cluster has a diameter δ.

$P(C_i^\delta([\text{type}], w_c))$ is the set of all the nodes P_j such that, $w_i[j] \in C_i^\delta([\text{type}], w_c)$. That is, it is the set of node id's from which P_i have received the vectors in the cluster $C_i^\delta([\text{type}], w_c)$.

Now we give our protocol in two steps. First, we give a protocol for asynchronous broadcast, that allows any sender to securely send a direction to all the other nodes. However, if the sender is faulty then the protocol might never terminate. Using this as a primitive we later give our asynchronous agreement protocol.

4.2.3.1 Asynchronous broadcast

As the name suggests using this protocol a sender node can send some message to all the other nodes in an asynchronous network. At first sight a naive protocol

of just sending the message to all other nodes one by one seems to be a valid protocol. However, this naive protocol does not work if the sender intentionally sends different messages to different nodes, which can easily happen in a network with faulty nodes. To guard from it, all the other nodes must communicate between each other to make sure they are receiving the same message, or a close approximation to it. However, as we have at most t faulty nodes, this verification also becomes tricky. The whole thing becomes more challenging because the network is not synchronous. As a result a receiver who is waiting for a message, cannot be certain whether to keep waiting (because the message might be taking a long time in the channel) or move on (the sending node might be faulty and not sending the message at all). Our protocol takes care of all these challenges.

Formally the protocol is defined as,

Definition 11. For $\eta > 0$, $\zeta > 0$, a protocol which is initiated by a sender node P_s, in an asynchronous network of n nodes, is an (η, ζ)-*asynchronous reference frame broadcast protocol* if it satisfies the following conditions.

Termination.

1. If the sender is correct then every correct node eventually completes the protocol.
2. If any correct node completes the protocol, then all the correct nodes eventually complete the protocol.

Consistency. If one correct node P_k outputs a direction v_k then all pairs of correct nodes P_i and P_j eventually output directions v_i, v_j where $d(v_i, v_j) \leq \eta$.

Correctness. If P_s is correct and broadcasts a direction u and if a correct node P_i outputs v_i then $d(u, v_i) \leq \zeta$.

We emphasize that the Termination condition of *asynchronous reference frame* ***broadcast*** is much weaker than the Termination condition of *asynchronous reference frame* ***agreement*** because in the broadcast protocol we do not require that the correct nodes complete the protocol if the sender is faulty. Also, in an

Protocol 6: AR-Cast

input : Sender inputs direction u

output : $\forall i$ P_i outputs direction v_i

Epoch 0: (Only Sender)

- **Send-to-all** (init, u).

Epoch 1: (Player P_i)

- Listen to init, echo, ready$_1$ and ready$_2$ type messages.
- **Wait until *Either*** *received one (init, u_i)* **Then**
  - **Send-to-all** (echo, u_i).
  - **Goto** Epoch 2.
- ***Or* until** *received a cluster of directions $C_i^{4\delta}([echo], w_c)$ of size at least $(n-2t)$* **And** *a cluster of directions $C_i^{10\delta}([ready_1,ready_2], v_c)$ of size at least $(t+1)$, so that, $d(w_c, v_c) \leq 10\delta$* **Then**
  - **Send-to-all** (ready$_2$, w_c).
  - **Goto** Epoch 3.

Epoch 2: (Player P_i)

- Listen to echo, ready$_1$ and ready$_2$ type messages.
- **Wait until *Either*** *there exists a cluster of directions $C_i^{4\delta}([echo], w_c)$ of size at least $(n-t)$* **Then**
  - **Send-to-all** (ready$_1$, w_c).
  - **Goto** Epoch 3.
- ***Or* until** *there exists a cluster of directions $C_i^{4\delta}([echo], w_c)$ of size at least $(n-2t)$* **And** *a cluster of directions $C_i^{10\delta}([ready_1,ready_2], v_c)$ of size at least $(t+1)$, so that, $d(w_c, v_c) \leq 10\delta$,* **Then**
  - **Send-to-all** (ready$_2$, w_c).
  - **Goto** Epoch 3.

Epoch 3: (Player P_i)

- **Wait until** *there exists a cluster of directions $C_i^{20\delta}([ready_1,ready_2], v_c)$ of size at least $(n-t)$* **Then**
  - Output v_c.
  - Halt

agreement protocol there is no designated sender node, whereas the broadcast protocol has a sender node.

We achieve asynchronous broadcast by our Protocol AR-Cast. The following theorem summarises its properties.

Theorem 7. *In a complete network of n nodes that are pairwise connected by public authenticated classical and quantum channels, if a bipartite δ-estimate direction protocol that uses m qubits to achieve success probability $q_{\text{succ}} \geq 1 - e^{-\Omega(m\delta^2)}$ is used, then Protocol AR-Cast is a $(42\delta, 14\delta)$-*asynchronous reference frame broadcast protocol*, with success probability at least $1 - e^{-\Omega(m\delta^2 - \log n)}$ that can tolerate up to $t < n/4$ faulty nodes.*

The Protocol 6: AR-Cast works roughly as follows. In Epoch 0 the sender sends its intended direction to all as a [init] type message. In Epoch 1 all the nodes wait until they receive an [init] from sender or sufficient number of confirmations from other nodes that they have received some directions and proceed to the next epoch. This way, even if some correct node never receives an [init] message, if the other correct nodes are advancing through the protocol, then this node in Epoch 1 will not stay behind waiting. In Epoch 2 the correct nodes, which have decided upon a direction, notify the other nodes about their decision by sending ready_1 or ready_2 type messages to all. All these previous epochs make sure that all the correct nodes eventually arrive at Epoch 3 and output a direction that satisfies Theorem 7. The formal proofs are given in Section 4.3.1.

4.2.3.2 Asynchronous agreement

Now we give our main Protocol A-Agree that uses AR-Cast as a subroutine and allows the correct nodes in an asynchronous network to agree on a reference frame.

In Epoch 0 of Protocol 7: A-Agree each of the nodes P_i proposes a direction u_i that represents their local frame. They broadcast this direction using AR-Cast. All the correct nodes wait for at least $(3t + 1)$ such broadcasts to be complete. Then they enter Epoch 1. Since, there are $(3t + 1)$ correct nodes they will eventually arrive at Epoch 1. In this step all the correct nodes create a bit string of length n where the j'th bit represents if the j'th AR-Cast has been completed successfully in Epoch 0. Then all the nodes send this bit strings to all by playing Asynchronous-IC.

Protocol 7: A-Agree

input : $\forall i$, P_i inputs direction u_i

output : $\forall i$, P_i outputs direction v_i

Epoch 0: (Player P_i)

Create a direction array w_i of size n.

$\forall j$, initialize $w_i[j] \leftarrow \perp$.

Run AR-Cast(u_i).

`// everyone broadcasts their local input`

Store received direction from P_j in $w_i[j]$.

After receiving $(3t+1)$ such directions **Goto** Epoch 1. However, still continue the incomplete AR-Casts in parallel.

Epoch 1: (Player P_i)

Create a bit string a_i of size n.

for $j \leftarrow 1$ **to** n **do**

  if $w_i[j] \neq \perp$ **then**

    Assign $a_i[j] \leftarrow 1$.

  else

    Assign $a_i[j] \leftarrow 0$.

`//` a_i `records which A-Casts are completed so far by` P_i

Run Asynchronous-IC(a_i).

`// This step reports to all which A-Casts are successfully received by` P_i

Store the output of Asynchronous-IC in vector b_i such that, element $b_i[j]$ is received from P_j.

`// After this step every correct nodes know which A-Casts are reported to be complete by which node`

Wait until *Asynchronous-IC completes* **Then**

  Goto Epoch 2

Epoch 2: (Player P_i)

Let k_i be the index of a column which has at least $(t+1)$ 1s in it. So that, for any other index l of column with (t+1) 1s $k < l$. `// After completion of Asynchronous-IC each row of` b_i `is a bit string of length` n. `That is` b_i `is essentially an` $n \times n$ `bit matrix.`

Wait until *the A-Cast initiated by* P_{k_i} *completes* **Then**

  Assign $v \leftarrow w_i[k_i]$.

  Abort all incomplete A-Casts that are running since Epoch 0.

  Output v.

After this they enters Epoch 2. In this Epoch every node has the same set of bit strings. They now look for the lowest inter k such that at least $(t+1)$ bit strings have a 1 in the k'th index of the string. If they have completed that k'th AR-Cast, then they output their direction received from that broadcast. If the k'th AR-Cast is not complete for a node, it waits until it completes and then output. The election of k ensures that at least one correct node has completed the k'th AR-Cast so by Consistency of asynchronous reference frame broadcast all the correct nodes will eventually complete the k'th AR-Cast. This ensures that the A-Agree eventually completes. There is no conditional loop in this protocol and all the subroutines run in constant expected time. So, the A-Agree is also a constant expected time protocol.

4.2.4 Resource requirements and performance

There are no conditional loops in the protocol and all the subroutines runs in $O(1)$ expected rounds. Thus our main Protocol 7: A-Agree is a constant expected round protocol.

The Protocol 6: AR-Cast requires $O(n^2)$ quantum messages, where each of the n nodes sends $O(n)$ quantum messages to others. However, in the main protocol each of the n nodes initiates its own AR-Cast in parallel. Therefore, the total number of quantum messages exchanged is $O(n^3)$

The main protocol A-Agree usages a classical interactive consistency protocol as a subroutine. Depending on the implementation chosen, this subroutine might require different number of classical messages. If we do not count the classical messages internal to this subroutine then the total number of classical message in our protocol is $O(n^3)$ because each of the quantum message have a classical part carrying type information. However, if we chose a classical asynchronous byzantine agreement subroutine that requires $O(n^4)$ messages, then the number classical message exchanged by the asynchronous interactive consistancy becomes $O(n^5)$. Therefore, the classical message count of the whole protocol becomes $O(n^5)$.

Some of these classical message contains a bit string of size $O(n)$, which determine the classical message size. This is learger than the synchronous case

where we only needed $O(\log n)$ bit length messages.

The number of qubits exchanged in each quantum message is determined by the success probability $1 - e^{-\Omega(m\delta^2 - \log n)}$. Here, once the network size n, 2-party direction estimation error δ and required success probability is fixed then we get m, the number of qubit required.

Now we are ready to give the formal proofs.

4.3 Proof of correctness

4.3.1 Asynchronous broadcast

To prove correctness of Protocol 6: AR-Cast we have to prove Theorem 7. We do this in a few steps that are formalised as lemmas. Note that at various Epochs the nodes send init, echo, ready$_1$ and ready$_2$, type messages. We first show that all the ready$_1$ and ready$_2$type message sent by correct nodes carries directions that are close to each other. This implies that when they terminate they output directions which are close to each other. We also show, that if one correct node terminates then all the other correct nodes also eventually terminate. For this, we show that when any correct node terminates it indicates that in the network, there exist sufficient number of echo, ready$_1$ and ready$_2$ type messages originating from correct nodes so that, all the correct nodes eventually complete all the Epochs of the protocol and successfully terminate.

For this, first we observe that, if two different correct nodes send [ready$_1$] type messages then the directions they send are close to each other with high probability.

Lemma 2. *For $t < n/4$, $\delta > 0, q_{\text{succ}} > 0$, if two correct nodes P_i and P_j send ([ready$_1$],u) and ([ready$_1$],v) respectively, then $d(u, v) \leq 10\delta$ with probability at least $q_{\text{succ}}^{n+n^2}$.*

Proof. In step 4 of Epoch 2 when a [ready$_1$] message is generated there are at most n init messages originated from the sender and at most n^2 echo messages generated by the other nodes. So, with probability at least $q_{\text{succ}}^{n+n^2}$ all the transmissions that are among correct nodes are successful. Conditional on this, we prove,

$$d(u, v) \leq 10\delta. \tag{4.1}$$

We show this in two steps. First, we show that there exists a common correct node P_k in $P(C_i^{4\delta}([\text{echo}], u))$ and $P(C_j^{4\delta}([\text{echo}], v))$. Then using the triangle inequality with the fact that the echo vector from P_k must be close to both of the cluster centers u and v, we derive Inequality (4.1).

Now, for the first step, let us denote A_i and A_j to be the set of nodes from which the vectors in $(C_i^{4\delta}([\text{echo}], u)$ have originated. And B_i and B_j to be the correct nodes in A_i and A_j respectively. Formally,

$$A_i = P(C_i^{4\delta}([\text{echo}], u)), \tag{4.2}$$
$$A_j = P(C_j^{4\delta}([\text{echo}], v)), \tag{4.3}$$
$$B_i = \{P_l : P_l \in A_i \text{ and } P_l \text{ is correct.}\}, \tag{4.4}$$
$$B_j = \{P_l : P_l \in A_j \text{ and } P_l \text{ is correct.}\}. \tag{4.5}$$

Note that at this step $|A_i| \geq n - t$ and $|A_j| \geq n - t$. We want to show that,

$$B_i \cap B_j \neq \emptyset. \tag{4.6}$$

We do this by contradiction: let us assume that,

$$B_i \cap B_j = \emptyset. \tag{4.7}$$

Note that,

$$|A_i| \geq n - t \tag{4.8}$$
$$\Rightarrow |A_i - B_i| + |B_i| \geq n - t, \tag{4.9}$$
$$\Rightarrow t + |B_i| \geq n - t, \tag{4.10}$$
$$\Rightarrow |B_i| \geq n - 2t, \tag{4.11}$$
$$\Rightarrow |B_i| > n - 2(n/4) = n/2. \tag{4.12}$$

Here, Inequality (4.10) holds because at most t of the nodes are faulty. And Inequality (4.12) holds because $t < n/4$.

Now,

$$\begin{aligned}
|A_i \cup A_j| &= |(A_i - B_i) \cup (A_j - B_j) \cup B_i \cup B_j|\,, \\
&\geq |(A_j - B_j)| + |B_i| + |B_j|\,, && (4.13)\\
&= |A_j| + |B_i|\,, && (4.14)\\
&> (n-t) + n/2, && (4.15)\\
&> n - n/4 + n/2 = 5n/4. && (4.16)
\end{aligned}$$

Here, Inequality (4.14) uses Inequality (4.7), Inequality (4.15) follows from the definition from the size of A_j and Inequality (4.12). And Inequality (4.16) follows because, $t < n/4$. However, this is a contradiction, because there are only n nodes in the network. Therefore, we have,

$$B_i \cap B_j \neq \emptyset. \tag{4.17}$$

So, there exists a common correct node $P_k \in B_i \cap B_j$ in $P(C_i^{4\delta}([\text{echo}], u))$ and $P(C_j^{4\delta}([\text{echo}], v))$. Since P_k is correct, it must have sent the same echo type message to both P_i and P_j. So, using the triangle inequality we have,

$$\begin{aligned}
d(w_i[k], w_j[k]) &\leq d(w_i[k], u_k) + d(u_k, w_j[k]), && (4.18)\\
&\leq \delta + \delta = 2\delta. && (4.19)
\end{aligned}$$

Now Inequality (4.1) follows because,

$$\begin{aligned}
d(u, v) &\leq d(u, w_i[k]) + d(w_i[k], w_j[k]) + d(w_j[k], v), && (4.20)\\
&\leq 4\delta + d(w_i[k], w_j[k]) + 4\delta, && (4.21)\\
&\leq 4\delta + 2\delta + 4\delta = 10\delta. && (4.22)
\end{aligned}$$

Here, Inequality (4.21) follows from the definitions of $C_i^{4\delta}([\text{echo}], u)$ and $C_j^{4\delta}([\text{echo}], v)$ and Inequality (4.22) follows from Inequality (4.19). □

In Lemma 2 we have shown the relation between two [ready$_1$] type directions from two different correct nodes. Now we show that if a correct node sends a [ready$_1$] and another correct node sends a [ready$_2$] type message then the directions they send are close with high probability. Both of these proofs use similar techniques.

Lemma 3. *For $t < n/4$, $\delta > 0, q_{\text{succ}} > 0$, if two correct nodes P_i and P_j send ([ready$_1$],u) and ([ready$_2$],v) accordingly, then $d(u, v) \leq 10\delta$ with probability at least $q_{\text{succ}}^{n+2n^2}$.*

Proof. When a [ready$_2$] message is generated there are at most n init, n^2 echo and in total n^2 [ready$_1$] or [ready$_2$] messages generated in the protocol. With probability at least $q_{\text{succ}}^{n+2n^2}$ all the transmissions that are among correct nodes are successful. Conditional on this, we show that,

$$d(u, v) \leq 10\delta. \tag{4.23}$$

We do this in two steps, first we show that there is a common correct node P_k in $P(C_i^{4\delta}([\text{echo}], u))$ and $P(C_j^{4\delta}([\text{echo}], v))$. Then using the triangle inequality with the fact that both of the cluster centers u and v must be close to the echo direction sent from P_k we prove the Inequality (4.23).

Now, for the first step, let us denote A_i and A_j to be the set of nodes from which the vectors in $(C_i^{4\delta}([\text{echo}], u)$ have originated. And B_i and B_j to be the correct nodes in A_i and A_j respectively. Formally,

$$A_i = P(C_i^{4\delta}([\text{echo}], u)), \tag{4.24}$$
$$A_j = P(C_j^{4\delta}([\text{echo}], v)), \tag{4.25}$$
$$B_i = \{P_l : P_l \in A_i \text{ and } P_l \text{ is correct.}\}, \tag{4.26}$$
$$B_j = \{P_l : P_l \in A_j \text{ and } P_l \text{ is correct.}\}. \tag{4.27}$$

Note that here $|A_i| \geq n - t$ and $|A_j| \geq n - 2t$. We want to show that,

$$B_i \cap B_j \neq \emptyset. \tag{4.28}$$

We do this by contradiction: let us assume that,

$$B_i \cap B_j = \emptyset. \tag{4.29}$$

Note that,

$$|A_i| \geq n - t \tag{4.30}$$
$$\Rightarrow |A_i - B_i| + |B_i| \geq n - t, \tag{4.31}$$
$$\Rightarrow t + |B_i| \geq n - t, \tag{4.32}$$
$$\Rightarrow |B_i| \geq n - 2t, \tag{4.33}$$
$$\Rightarrow |B_i| > n - 2(n/4) = n/2. \tag{4.34}$$

Here, Inequality (4.32) holds because at most t of the nodes are faulty. And Inequality (4.34) holds because $t < n/4$.

Now,

$$\begin{aligned} |A_i \cup A_j| &= |(A_i - B_i) \cup (A_j - B_j) \cup B_i \cup B_j|, \\ &\geq |(A_j - B_j)| + |B_i| + |B_j|, && (4.35) \\ &= |A_j| + |B_i|, && (4.36) \\ &> (n - 2t) + n/2, && (4.37) \\ &> n - n/2 + n/2 = n. && (4.38) \end{aligned}$$

Here, Inequality (4.37) follows from the definition of A_j and Inequality (4.34). And Inequality (4.38) follows because, $t < n/4$. However, this is a contradiction, because there are only n nodes in the network. Therefore, we have,

$$B_i \cap B_j \neq \emptyset. \tag{4.39}$$

So, there exists a common correct node P_k in $P(C_i^{4\delta}([\text{echo}], u))$ and $P(C_j^{4\delta}([\text{echo}], v))$. As P_k is correct, it must have sent the same echo type message to both P_i and P_j. So, using the triangle inequality we have,

$$d(w_i[k], w_j[k]) \leq d(w_i[k], u_k) + d(u_k, w_j[k]), \tag{4.40}$$
$$\leq \delta + \delta = 2\delta. \tag{4.41}$$

Now Inequality (4.1) follows because,

$$d(u, v) \leq d(u, w_i[k]) + d(w_i[k], w_j[k]) + d(w_j[k], v), \tag{4.42}$$
$$\leq 4\delta + d(w_i[k], w_j[k]) + 4\delta, \tag{4.43}$$
$$\leq 4\delta + 2\delta + 4\delta = 10\delta. \tag{4.44}$$

Here, Inequality (4.43) follows from the definitions of $C_i^{4\delta}([\text{echo}], u)$ and $C_j^{4\delta}([\text{echo}], v)$ and Inequality (4.44) follows from Inequality (4.41).

□

Now we show that all the correct nodes cannot send only [ready$_2$] type messages. That is, if there exists a [ready$_2$] message sent from a correct node, then there must pre-exist a [ready$_1$] message sent from another correct node.

Lemma 4. *For $t < n/4$, $\delta > 0, q_{\text{succ}} > 0$, if a correct node P_j sends ([ready$_2$],v), then with probability at least $q_{\text{succ}}^{n+2n^2}$, there exists a correct node P_i which has sent ([ready$_1$],u) .*

Proof. When a [ready$_2$] message is generated there are at most n [init], n^2 [echo] and in total n^2 [ready$_1$] or [ready$_2$] messages generated in the protocol. With probability at least $q_{\text{succ}}^{n+2n^2}$ all the transmissions that are among correct nodes are successful. In this case, just before making the decision to send a ([ready$_2$],v) message node P_j must have received at least (t+1) [ready$_1$] or [ready$_2$] messages from nodes in $P(C_i^{10\delta}([\text{ready}_1,\text{ready}_2]v_c))$. Of these, at least one node—let's call it P_k—is correct. If P_k has also sent a [ready$_2$] type message, then we can find another correct node in its $P(C_k^{10\delta}([\text{ready}_1,\text{ready}_2]v_c))$ and so on. This way, eventually we will find a correct node which has sent a [ready$_1$] type message.

To see this, let us define a directed graph $G(V, E)$ with vertex set $V = \{P_i : P_i \text{ is correct}\}$, and

$$E = \{(P_k, P_i) : P_k \text{ has sent ready}_2 \text{ after receiving ready}_1 \text{ or ready}_2 \text{ from } P_i\}. \tag{4.45}$$

One can convince oneself that G is a directed acyclic graph because any cycle in the graph would violate the cause and effect relation of the edge directions. Now if we look at the connected component of this graph containing P_j there must exist a node P_i in this component with no outgoing edges. Because V only contains correct nodes. This implies P_i is a correct node, which has sent a [ready$_1$] type message ([ready$_1$],u). This completes the proof. □

Now the only thing that remains is to show that two [ready$_2$] type directions sent from two correct nodes are close with high probability.

Lemma 5. *For $t < n/4$, $\delta > 0, q_{\text{succ}} > 0$, if two nodes P_i and P_j sends ([ready$_2$],u) and ([ready$_2$],v) respectively, then $d(u,v) \leq 20\delta$ with probability at least $q_{\text{succ}}^{n+2n^2}$.*

Proof. When a [ready$_2$] message is generated there are at most n [init], n^2 [echo] and in total n^2 [ready$_1$] or [ready$_2$] messages generated in the protocol. With probability at least $q_{\text{succ}}^{n+2n^2}$ all of these transmissions that are among correct nodes are successful. Conditional on this, we show that, if correct P_i sends ([ready$_2$],u) then from Lemma 4 there exists a correct node P_k which has sent ([ready$_1$],w). From Lemma 3,

$$d(u,w) \leq 10\delta, \tag{4.46}$$

and

$$d(v,w) \leq 10\delta. \tag{4.47}$$

Using the triangle inequality with these we get,

$$d(u,v) \leq d(u,w) + d(w,v) \leq 10\delta + 10\delta = 20\delta. \tag{4.48}$$

□

Now we are ready to prove that our Protocol 6 satisfies the first termination condition of definition 11.

Lemma 6 (Termination 1)**.** *For $t < n/4$, $\delta > 0, q_{\text{succ}} > 0$, if the sender P_k is correct then the Protocol 6* AR-Cast *eventually terminates with probability at least $q_{\text{succ}}^{n+n^2}$.*

Proof. There are at most n [init] messages, n^2 [echo] messages and n^2 [ready$_1$] or [ready$_2$] type messages exchanged in the protocol. With probability at least $q_{\text{succ}}^{n+2n^2}$ all of these transmissions that are among correct nodes are successful. In this case, if the sender is correct then all the correct nodes eventually receive [init] messages that are at most 2δ apart from each other and send an echo message. So, all the received [echo] messages are at most 3δ apart from the received direction in the [init] message of any correct node. Any node that has sent a [ready$_1$] type message will go to epoch 3. The faulty nodes cannot stop the [init] and [echo] messages from correct nodes but they can manipulate the delays, so that some of

the correct nodes send [ready$_2$] type messages. After sending the [ready$_2$] these correct nodes will eventually arrive at Epoch 3. From Lemma 2 and Lemma 3 we can see that for any correct P_i all the received [ready$_1$] and [ready$_2$] directions will be in $C_i^{16\delta}([\text{ready}_1,\text{ready}_2], v_c)$. And because there are $(n-t)$ of them originating from the correct nodes the Protocol 6 AR-Cast will eventually terminate. Note that, if the sender is faulty, then the definition of (η, ζ)-*reference frame broadcast protocol* (Derinition 11) do not require any termination. □

Now we show that if one correct node outputs a direction, then all the correct nodes eventually output directions that are close to each other.

Lemma 7 (Consistency)**.** *For $t < n/4$, $\delta > 0, q_{\text{succ}} > 0$, in Protocol* AR-cast*, if a correct node P_k outputs v_k then all pair of correct nodes P_i, P_j eventually output v_i, v_j respectively such that, $d(v_i, v_j) \leq 42\delta$ with probability at least $q_{\text{succ}}^{n+n^2}$.*

Proof. When a [ready$_2$] message is generated there are at most n init, n^2 echo and in total n^2 [ready$_1$] or [ready$_2$] messages generated in the protocol. With probability at least $q_{\text{succ}}^{n+2n^2}$ all of these transmissions that are between correct nodes are successful. In this case, we prove,

$$d(v_i, v_j) \leq 42\delta, \tag{4.49}$$

by showing that the successful completion of P_k implies there are enough echo, [ready$_1$] and [ready$_2$] type messages generated by correct nodes so that all the other correct nodes eventually receive them and successfully terminate and each pair of their outputs satisfies Inequality (4.49).

Now, if a correct node P_k outputs v_k then this implies it has received at least $(n-t)$ [ready$_1$] or [ready$_2$] messages from nodes in $P(C_k^{20\delta}([\text{ready}_1,\text{ready}_2], v_k))$, of which at least $(n-2t)$ are correct. Messages from these correct nodes eventually reach all the other correct nodes. Also, from Lemma 4 there exists a correct node that has sent a [ready$_1$] message, which implies all the correct nodes eventually receive at least $(n-2t)$ echo messages. That is, all the correct nodes waiting in Epoch 1 or Epoch 2 will satisfy the condition of sending a [ready$_2$] message and go to Epoch 3. Any correct node P_i, P_j waiting in Epoch 3 will eventually receive all the [ready$_1$] or [ready$_2$] messages sent from correct nodes in $P(C_i^{20\delta}([\text{ready}_1,\text{ready}_2], v_i))$ and $P(C_j^{20\delta}([\text{ready}_1,\text{ready}_2], v_j))$ accordingly, and output v_i, v_j accordingly.

Now we show that $P(C_i^{20\delta}([\text{ready}_1,\text{ready}_2], v_i))$ and $P(C_j^{20\delta}([\text{ready}_1,\text{ready}_2], v_j))$ have at least one common correct node, which implies the cluster centers are close.

To see this note that each of these clusters have at least $(n-2t) > n-2(n/4) = n/2$ correct nodes. That is more than n correct nodes in total. However there are total n nodes in the networks. This implies at least some of the correct nodes are common in both clusters. Let P_l be such a node.

Now using triangular inequality we have,

$$\begin{aligned} d(v_i, v_j) &\leq d(v_i, v_i[l]) + d(v_i[l], v_l) \\ &\quad + d(v_l, v_j[l]) + d(v_j[l], v_j), \qquad (4.50) \\ &\leq 20\delta + \delta + \delta + 20\delta = 42\delta. \qquad (4.51) \end{aligned}$$

Here Inequality (4.51) follows using Lemma 5. □

Now the second termination condition.

Lemma 8 (Termination 2)**.** *For $t < n/4$, $\delta > 0, q_{\text{succ}} > 0$, if a correct node P_i completes the protocol then all the correct nodes complete the protocol with probability at least $q_{\text{succ}}^{n+2n^2}$.*

Proof. This lemma is a corollary of Lemma 7. Because Lemma 7 ensures completion with probability at least $q_{\text{succ}}^{n+2n^2}$.

□

Now we are ready to prove that our protocol satisfies the correctness condition of definition 11.

Lemma 9 (Correctness)**.** *For $t < n/4$, $\delta > 0, q_{\text{succ}} > 0$, if a correct sender P_s sends (init,u) and a correct node P_i outputs v_i then $d(u, v_i) \leq 14\delta$ with probability $q_{\text{succ}}^{n+2n^2}$.*

Proof. There are at most n init messages, n^2 echo messages and n^2 [ready$_1$] or [ready$_2$] type messages exchanged in the protocol. With probability at least $q_{\text{succ}}^{n+2n^2}$ all of these transmissions that are between correct nodes are successful.

In this case we prove the lemma in three steps. First, we show that all the [ready$_1$] type directions sent from correct nodes are close to u. Secondly, we show that all the [ready$_2$] type directions sent from the correct nodes are close to u. And finally, from these we conclude the proof.

For the first step, let us assume that correct node P_i has sent a ([ready$_1$], v_i) message in Epoch 2. So, it has received at least $(n-t)$ echo type messages, of which at least $(n-2t)$ are from correct nodes. Let's assume for some correct node P_j $w_i[j] \in C_i^{4\delta}(v_i)$. Since P_j is correct, using the triangle inequality, we have,

$$d(u, w_i[j]) \leq d(u, u_j) + d(u_j, w_i[j]), \tag{4.52}$$
$$\leq \delta + \delta = 2\delta. \tag{4.53}$$

The diameter of the cluster $C_i^{4\delta}(v_i)$ is 4δ. So, we have, $d(v_i, w_i[j]) \leq 2\delta$. Using this and (4.53) with the triangle inequality, we have,

$$d(u, v_i) \leq d(u, w_i[j]) + d(w_i[j], v_i), \tag{4.54}$$
$$\leq 2\delta + 2\delta = 4\delta. \tag{4.55}$$

Now, for the second step, let us assume that a correct node P_l has sent a ([ready$_2$], v_l) message from Epoch 1 or Epoch 2. So, v_l is a cluster center of at least $(n-2t)$ echo type messages. Of which at least $(n-3t)$ are correct. So, a similar reasoning to the previous step shows,

$$d(u, v_l) \leq 4\delta. \tag{4.56}$$

Finally, since the sender is correct from Lemma 6 we know, all the correct nodes eventually enter Epoch 3 and successfully complete the epoch.

Let us assume a correct node P_i has received a cluster of [ready$_1$] or [ready$_2$] type directions $C_i^{20\delta}$([ready$_1$,ready$_2$], v_c) of size at least $(n-t)$. So, there is a correct node P_k for which $v_i[k] \in C_i^{20\delta}$([ready$_1$,ready$_2$], v_c). Here, $C_i^{20\delta}$([ready$_1$,ready$_2$], v_c) is a cluster of diameter 20δ. So, we have $d(v_i[k], v_c) \leq 10\delta$. Using the triangle inequality with this, and (4.55) and (4.56), we have,

$$(u, v_c) \leq d(u, w_i[k]) + d(w_i[k], v_c), \tag{4.57}$$
$$\leq 4\delta + 10\delta = 14\delta. \tag{4.58}$$

This concludes the proof.

□

Now we give an auxiliary lemma that shows how the probability of success scales with the number of nodes and the success probability of the δ-estimate direction protocol.

Lemma 10. *For $\delta > 0$ if a two-node direction estimation protocol is used that transmits m qubits to δ-estimate a direction with success probability $q_{\text{succ}} \geq (1 - e^{-\Omega(m\delta)})$ then with probability at least $q_{\text{succ}}^{n+2n^2} \geq 1 - e^{-\Omega(m\delta^2 - \log n)}$, all the direction transmissions of init, echo, [ready$_1$] and [ready$_2$] type messages are successful.*

Proof. There are at most n init messages, n^2 echo messages and n^2 [ready$_1$] or [ready$_2$] type messages exchanged in the protocol. With probability at least $q_{\text{succ}}^{n+2n^2}$ all of these transmissions that are between correct nodes are successful. Now,

$$q_{\text{succ}}^{n+2n^2} \geq (1 - e^{-\Omega(m\delta^2)})^{n+2n^2}, \tag{4.59}$$

$$\geq 1 - (n + 2n^2)e^{-\Omega(m\delta^2)}, \tag{4.60}$$

$$\geq 1 - e^{-\Omega(m\delta^2 - \log n)}. \tag{4.61}$$

Here Inequality (4.60) follows using Bernoulli's inequality, which is, $(1 + x)^r > 1 + rx$ for all real $x \geq -1$ and integer $r \geq 2$.

□

We see that, Theorem 7 follows from Lemma 6, 7, 8, 9 and 10.

4.3.2 Asynchronous agreement

We have presented an asynchronous broadcast protocol where a designated sender initiates the protocol with a direction. This protocol has a weaker success condition than an asynchronous *agreement* protocol. If the sender is faulty, then an asynchronous broadcast protocol might never terminate because in this case the correct receivers cannot decide whether the sender is faulty and not sending the [init] message, or correct but very slow. Whereas, in an asynchronous reference frame agreement protocol the main goal is to allow the correct nodes to agree on some direction despite the presence of, up to a certain number of, unidentified faulty nodes in the network. This requires extra caution to make sure that the protocol eventually terminates. We show that our Protocol 7:A-Agree successfully solves this problem by proving Theorem 6.

There are three epochs in Protocol 7. Any correct node that successfully terminates must start at Epoch 0 and terminate at Epoch 3. At each Epoch the nodes inside it, and all the messages transmitted and received by the node while in that Epoch satisfies some invariance properties. We describe and prove these properties in the following lemmas. We first show that a correct node will eventually enter Epoch 1.

Lemma 11. *For $t < n/4$, all the correct nodes eventually enter Epoch 1 of* a-agree *with probability at least $q_{\text{succ}}^{n^2+2n^3} \geq 1 - e^{-\Omega(m\delta^2 - \log n)}$.*

Proof. Each of the n nodes has initiated an AR-Cast in Epoch 0. Each of the AR-Casts has a success probability at least $q_{\text{succ}}^{n+2n^2}$. So, with probability at least $q_{\text{succ}}^{n^2+2n^3}$ all the AR-Casts from correct senders are successful. From Lemma 10 this is at least $1 - e^{-\Omega(m\delta^2 - \log n)}$.

As $t < n/4$, there are at least $(3t+1)$ correct nodes who initiates AR-Cast as sender. According to Theorem 7 these $(3t+1)$ AR-Casts will eventually terminate. So, every correct receiver will eventually receive at least $(3t+1)$ directions and go to Epoch 1 with probability at least $q_{\text{succ}}^{n^2+2n^3}$. □

Each of the correct nodes stores the output of the Asynchronous-IC protocol in an array b_i. Here b_i can be seen as a $n \times n$ matrix of bits where row j is received from node j. We can observe the following property of this matrix.

Lemma 12. *For $t < n/4$ and correct node P_i, after instruction 9 of Epoch 1 of* a-agree*, there exists a column in b_i with at least $(t+1)$ 1s in it.*

Proof. We show this by a counting argument. Note that a correct node arrives at Epoch 1 only after it have received at least $(3t+1)$ directions from other players. As a result after step 7 of Epoch 1 a_i contains at least $(3t+1)$ 1's. These a_i's become the rows of b_i after step 9. There are at most t faulty nodes. So, at least $(3t+1)$ rows of b_i are originated from correct nodes. Each of these rows must contain at least $(3t+1)$ 1's. So b_i has at least $(3t+1)^2$ 1s.

However, if no column had at least $(t+1)$ 1s, then there would be at most $(4t+1) \times t$ 1s in b_i. This contradicts the fact that b_i has at least $(3t+1)^2$ 1s. So, there must exist a column with at least $(t+1)$ 1s in it. □

We show that all the correct nodes select the same column, which has at least $t+1$ 1s in it.

Lemma 13. *After instruction 2 of Epoch 2 of a-agree, if correct node P_i has k_i and correct node P_j has k_j, then $k_i = k_j$.*

Proof. After completion of Protocol Asynchronous-IC in Epoch 1, all the correct nodes compute the same output vector. That is, $b_i = b_j$ for all correct P_i and P_j. Also, from Lemma 12 we know there exists a column in b_i with at least $(t+1)$ 1s. So, in Epoch 2 step 2 when correct node P_i and P_j selects k_i and k_j to be the chronologically smallest column index that has at least $(t+1)$ 1s. They select the same column. i.e., $k_i = k_j$. □

Now that every correct node P_i agrees on a column k_i of b_i, we observe that.

Lemma 14. *If a correct node P_i selects k_i in instruction 2 of Epoch 2, then the AR-Cast initiated by P_{k_i} in Epoch 0 eventually completes successfully.*

Proof. We show this by showing that at least one correct node has completed the AR-Cast initiated by P_{k_i}. Then the lemma follows from the termination condition of AR-Cast.

Each row $b_i[j]$ represents P_i's knowledge of which AR-Casts are successfully received by P_j. For example, if $b_i[j][l] = 1$, then it means node P_j has reported to P_i that it has completed the AR-Cast initiated by node P_l in Epoch 0. If there are at least $(t+1)$ 1s in the k_ith column of b_i, then it means that there are $(t+1)$ nodes who report that they have received the AR-Cast initiated by node P_{k_i} in Epoch 0. At least one of these reports is from a correct node. So, from the termination condition of AR-Cast (Lemma 7) all the correct nodes eventually successfully complete the AR-Cast by P_k. □

Now we are ready to prove **Theorem 6.**

Proof. There are at most n AR-Casts initiated in Epoch 0 of which $(n-t)$ are by correct nodes. From Lemma 10 each of these succeeds with probability $q_{\text{succ}}^{n+2n^2} \geq 1 - e^{-\Omega(m\delta^2 - \log n)}$. So all the correct AR-Casts succeed with,

$$q_{\text{succ}}^{n^2+2n^3} \geq (1 - e^{-\Omega(m\delta^2 - \log n)})^n, \tag{4.62}$$

$$\geq 1 - e^{-\Omega(m\delta^2 - \log n)}. \tag{4.63}$$

Here Inequality (4.63) follows from Bernoulli's inequality. Contitional on this, we show the correctness and termination of Protocol 7: A-Agree.

Correctness. To prove consistency we show that if a correct node P_i outputs v_i and a correct node P_j outputs v_j then $d(v_i, v_j) \leq 42\delta$. From step 4 of Epoch 2 of A-Agree we see that,

$$v_i = w_i[k_i], \tag{4.64}$$

$$v_j = w_j[k_j]. \tag{4.65}$$

From Lemma 7 we know that for $t < n/4$,

$$d(w_i[k_i], w_j[k_j]) \leq 42\delta. \tag{4.66}$$

This with (4.64) and (4.65) gives,

$$d(v_i, v_j) \leq 42\delta. \tag{4.67}$$

Termination To prove termination we have to show that every correct node P_i terminates with an output direction v_i.

To prove this we show that P_i eventually completes all the Epochs of A-Agree. From Lemma 11 we see that P_i must enter Epoch 1 from Epoch 0. All the steps in Epoch 1 are of constant expected time. So, a correct node will eventually complete them and go to Epoch 2. Only in step 3 of Epoch 2 P_i waits for completion of AR-Cast from P_{k_i}. However, from Lemma 14 we know that this AR-Cast eventually successfully completes. All the other incomplete AR-Casts are then aborted at Step 5 and the protocol terminates with output v_i.

□

4.4 Discussion

We have presented the first asynchronous reference frame agreement protocol. The synchronous protocol for spatial reference frame agreement that we have presented in Chapter 3 can tolerate up to $t < n/3$ faulty nodes. Whereas, our asynchronous protocol tolerates $t < n/4$ faulty nodes. Even though we pay this extra price

in fault tolerance, an asynchronous protocol is a fully general reference frame agreement protocol. For example, if we have a network where the local clocks of the nodes are not synchronised but the communication delay of each link is fixed, then our Protocol A-Agree would still work in this setting and allow the nodes to align their Cartesian reference frames. Once this is done, any bipartite quantum clock synchronisation protocol (for example, [83]) can be used as a primitive to achieve network-wide clock synchronisation.

For classical Byzantine agreement problems, where multiple nodes want to agree on a bit, it is known that any optimal protocol can only tolerate at most $t < n/3$ faulty nodes [80]. There exists asynchronous protocols that achieve this fault tolerance in the classical setting [76–79]. However, no such bounds are known for our setting where communication is imperfect even among the correct nodes. Since our synchronous protocol from Chapter 3 achieves $t < n/3$ fault tolerance, there is hope that asynchronous protocols might be found that achieve this bound. It also remains open whether entanglement can help to achieve better fault tolerance in the asynchronous multiparty setting.

5

Routing in a quantum network

In this chapter, we study a particular type of quantum network that uses entanglement swapping for routing quantum information. We discuss the operations and design principles for such a network. We give an overview of related works and define the idealised network primitives that we later use to design our routing protocols. As a contribution of this thesis we introduce the concept of routing graphs that facilitate efficient routing.

5.1 Motivation and related works

There are many uses for entangled states that are spatially separated. A few of such are super dense coding [99], Bell's theorem based cryptography [18], reducing communication complexity [100], quantum secret sharing [101] and testing local hidden variable theories [56, 102–104]. Moreover, Bell states that are shared between two spatially separated nodes can be used to teleport unknown quantum states [55]. Therefore, any quantum network that facilitates entanglement distribution between any pair of its nodes allows distributed quantum computing [8–10, 12, 13] over it.

A quantum network that allows any two spatially separated nodes to create entanglement between them by performing entanglement swapping (Chapter 1.2.12) on Bell pairs is called an *entanglement swapping based quantum network*. Such a network was first proposed by Biham *et al.* [60] in the context of cryptographic applications. This network was later studied by Bose *et al.* [105] for applications in

creating multi-particle entangled Greenberger-Horne-Zeilinger (GHZ) states [106]. Li *et al.* [107] studied how to create long range entanglement using this network where the Bell pair creation and local entanglement swapping operations are imperfect. There are works related to entanglement percolation [108] that study how entanglement swapping on weakly entangled pairs of qubits along multiple paths connecting two spatially separated nodes can increase the success probability of creating entanglement between them. This entanglement percolation model for quantum network is also studied in the context of quantum random graphs [109]. However, routing protocols that efficiently distribute entanglement between any pair of source and destination nodes in an entanglement swapping based quantum network remaines unstudied.

We briefly discuss how different types of quantum repeaters work and how these repeating techniques enable the construction of quantum routers. However, before going into that, we start with a brief description of quantum links.

5.2 Physical quantum links

A physical quantum link is a direct connection between two distant nodes that allows them to exchange quantum information between each other. Examples of such links are line of sight free space photon channels [40, 110, 111] and optical fibres [112–115]. These links enable us to exchange photon polarisation qubits [116] where the quantum information is encoded in the polarisation direction of a photon, or time-bin qubits [82, 117] where the quantum information is encoded in the discrete time-bins in which a single photon either exists or not. In some protocols the physical quantum links (for example, in DLCZ protocol [118]) are only used to establish entanglement between spatially separated nodes. We refer to all of these types of links as *physical quantum links* or *quantum links*, to contrast it from virtual links that we define later. All these links are physical realisations of different types of direct quantum channels (see Chapter 1.2.9.1).

5.3 Quantum repeater

Communication over a long distance suffers from losses. In classical signal processing, such losses are overcome using repeaters that enhance the signal's strength without losing its information content. However, in quantum communication such repeaters are not possible due to the no cloning theorem [119], which rules out replication of any unknown quantum state. Therefore, quantum repeaters are necessarily different from classical repeaters.

Consider a direct quantum link between two spatially separated nodes A and C that are a distance L apart from each other. If L is very large, then the overall transmission of the link would be very small. The idea of a quantum repeater approach [120, 121] is that a Bell state shared between this long distance L can be created by entanglement swapping (see, Chapter 1.2.12) where we start with two Bell pairs that are only $L/2$ distance apart. A node in the middle performs entanglement swapping on these two Bell pairs to attain a Bell pair over the distance L. We can cascade this method and start with Bell pairs which are $L/4$ distance apart (see Figure 5.1). And in two steps of entanglement swapping we are able to create entanglement between nodes which are L distance apart. Note that, we could, in principle, perform the entanglement swapping operation in the middle nodes in any order. For example, for the setting in Figure 5.1 (a) nodes B, C and D could perform the entanglement swapping operation one by one respectively to achieve the final state of Figure 5.1 (c). This requires $O(n)$ steps for repeating to n hops compared to $O(\log(n))$, as originally shown in Figure 5.1.

One approach to create the entanglement between the nodes which are close to each other is to locally create a Bell pair and send one of them to the other node via a direct quantum link. Implementing such an approach would require the ability to detect whether a photon has arrived without destroying entanglement, which is very difficult in practice [121]. A better way is to create the entanglement at a distance [122, 123] where an entanglement between atoms A and B can be created by detecting a photon that could have been emitted by either one. The detection is performed in such a way that it is impossible to determine this 'which-way' information which effectively entangles the atoms. If no photon is detected, then we know that the entanglement creation failed and we have to try

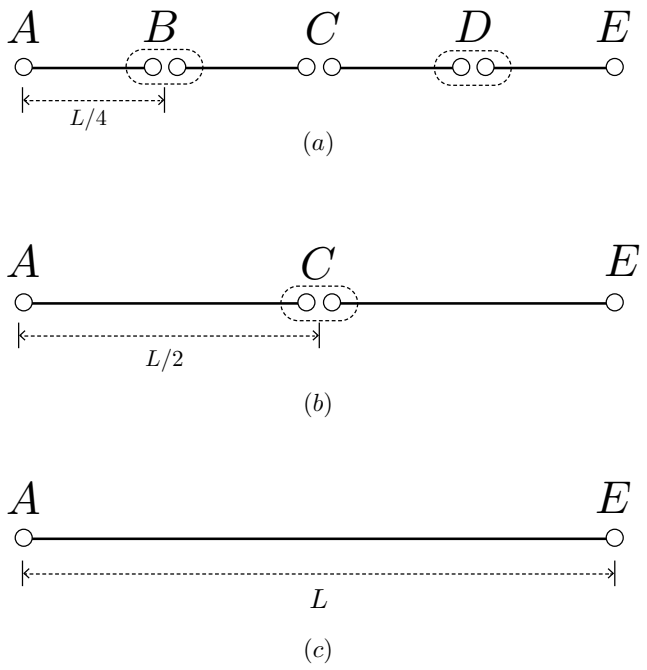


Figure 5.1: Quantum repeater using entanglement swapping. - Small circles represent the qubits held by the nodes A, B, C, D and E. The lines connecting two small circles indicate that they are in a Bell state. In (a) nodes B and D performs entanglement swapping, which creates entanglement between nodes A, C and C, E. In (b)) the node C performs entanglement swapping on its qubits that allows the nodes A and E to be entangled. This way, in two steps, initial entanglement created between nodes, which are $L/4$ distance apart, can be used to create entanglement between nodes L distance apart.

again. The most influential proposal of a quantum repeater protocol that uses this technique is the DLCZ protocol [118], which has inspired a large number of successful experiments (for example, [124–126]) that demonstrate the practicality of this approach. A study of the advantages of quantum repeating approach over directly sending an entangled qubit to a distance L can be found in [121].

So far, we have not discussed the effect of transmission loss, imperfect entanglement creation between the adjacent repeaters, and imperfect entanglement swapping operations. A recent survey by Munro *et al.* [127] has summarised various techniques that are being investigated to overcome these issues. They categorise the entanglement swapping based repeaters in two generations.

In the *first generation* approach, the process begins with establishing multiple copies of weakly entangled pair of qubits in adjacent repeaters. Then a process called entanglement purification [128–131] is used to generate a Bell pair out of them. On the next step, entanglement swapping is performed on these spatially separated Bell pairs to get long distance entanglement. Since the entanglement swapping operations might be imperfect due to the techniques used or due to the imperfection of the instruments [120], multiple copies of these newly generated long distance entanglements are again purified to get a state close to a Bell pair. That is, after every entanglement swapping step, these repeaters perform entanglement purification. The entanglement purification protocols used in these first generation repeaters require round trip classical communication between the participating nodes. Thus the frequency of performing such quantum repeating gets bottlenecked.

The *second generation* repeater approach usages entanglement swapping operations which are deterministic [59]. They also use error correction codes [132–135] to perform entanglement purification. These new techniques allow the entangled links to be used for the next step, before the classical message generated at the purification step travel from one node to other. This approach substantially decreases the performance bottleneck suffered by the first generation repeaters.

Munro *et al.* [127] also discusses a third type of repeater, that does not require quantum memory [136] in the nodes. However, to achieve quantum routing, we want to use the repeating techniques similar to the first and second generation repeaters that use quantum memory.

5.4 Quantum router

A router in a computer network forwards data packets in the network while deciding which path the packets take to reach its destination. A functioning quantum network would also require such routers where the data packet carry quantum information encoded in qubits. One approach of achieving such routing capability is via entanglement swapping and teleportation (Chapter 1.2.12 and 1.2.11). In this mode of quantum routing, the routers should cooperate with each to distribute Bell pairs between any two nodes in a network. Once a Bell pair is established, the nodes can use it to teleport a qubit in an arbitrary unknown quantum state from one to another.

In the previous section, we have seen that a node, which can perform entanglement swapping, can act as a quantum repeater. Now we show how this node can act as a quantum router. In Figure 5.2, we see that initially each of the nodes A, C, D shares a Bell pair with node B. In the node B, the qubit B_X forms a Bell pair with a qubit in the node X for $X \in \{A, C, D\}$. In the figure these Bell pairs are represented by lines connecting two small circles. Here node B can act as a quantum router in the following way. If it decides to create entanglement between node A and C then it performs entanglement swapping operation on its qubit B_A and B_C. Similarly, if it decides to create entanglement between node A and D then it performs entanglement swapping on B_A and B_D. The outcome of these two alternate routing decisions are shown in the figure. There is a third alternative, where the entanglement swapping can be performed on B_C and B_D.

In general, such quantum routers could have multiple qubits entangled to various other nodes. And by collaborating with each other they allow entanglement distribution between any two nodes in the network, thus achieving quantum routing.

5.4.1 Necessary properties of a quantum router

From this discussion we see that an entanglement swapping based quantum router should have the following properties:

- A quantum router has quantum memory.

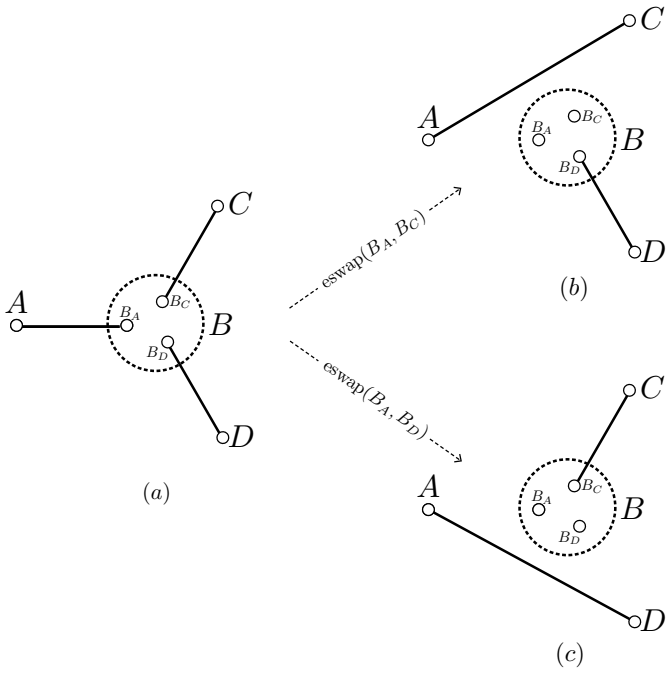


Figure 5.2: Quantum router using entanglement swapping. - Small circles represent the qubits held by the nodes A, B, C and D. The lines connecting two small circles indicate that they are Bell pairs. (a) is the initial configuration. If $eswap(B_A, B_C)$ operation is performed by B then A and C get entangled as shown in (b). Alternately, B can chose to entangle A and D as in (c). The third alternative, where C and D get entangled, is not shown here is not shown.

- A quantum router can store qubits for at least the time required to perform any communication over the network.
- A quantum router can perform the Bell state measurement on any pair of qubits in its memory.
- A quantum router can communicate with any other node using classical channels.
- A quantum router can create entanglement with another router, which is directly connected to it via physical quantum links

If any quantum node has these capabilities, it can act as a quantum router by selectively teleporting a qubit to any of its neighbours or to a distant node with which it shares a Bell state. The simplicity of these assumptions allow such a quantum node to be physically realised using current technologies [126, 137]. Moreover, to perform quantum routing in this method we do not require any full-fledged quantum computer. This opens up opportunities to implement quantum protocols over a network even before building full-fledged quantum computers.

5.5 Network graph

A network graph $G = (V, E)$ is a graph with vertex set V and edge set E where each vertex $v \in V$ represents a quantum node, (for example, quantum router, quantum computer, or other quantum devices) and each edge $e \in E$ represents a physical quantum link that connects two adjacent quantum nodes in the network. In Figure 5.3 (a) the small circles represent the nodes and the back solid lines represent the quantum links.

5.6 Routing in a quantum network

Now we illustrate quantum routing over a network of quantum routers using an example. In Figure 5.3 (a) we see 6 nodes A, B, C, D, E and F, connected using physical links represented by black lines. Let us start at a situation, where,

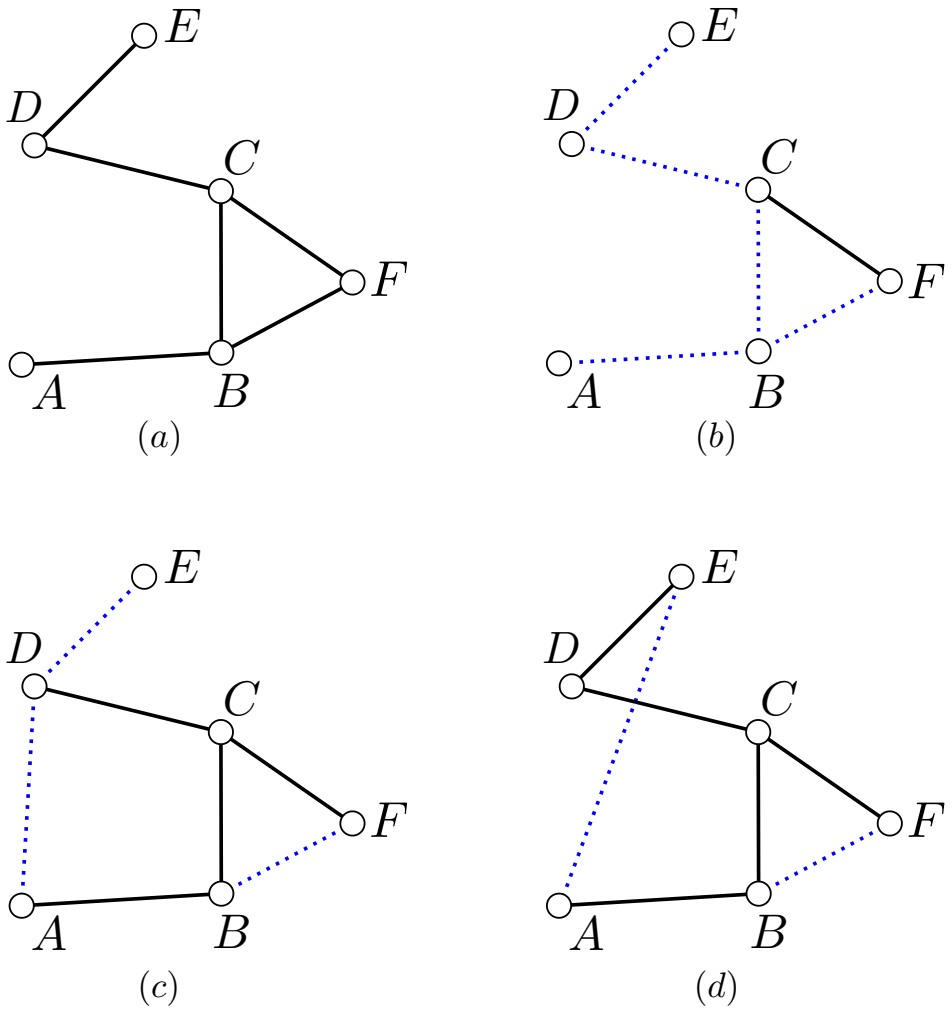


Figure 5.3: Routing example in a quantum network. - Small circles represent quantum nodes. (a) shows the network graph where solid balk lines represent quantum links. In (b) some of the nodes have shared entanglement between each other (shown in blue). After performing entanglement swapping in node C and B on the respective qubits, we get a new entanglement between A, D as in (c). This new entanglement is again swapped in D to get entanglement between A and E as shown in (d).

using these physical quantum links, some of the nodes have created entanglement between them. These pairwise shared Bell pairs are represented by blue dotted lines in Figure 5.3 (b). Note that if two nodes are directly connected by a physical link, then they can share entanglement directly (for example, using the DLCZ protocol [118]), i.e. without performing any entanglement swapping. Now, node A wants to create an entanglement with node D. One can see that, there exists a blue path from A to D via nodes B and C. These middle nodes B and C can perform an entanglement swapping operation on the entangled qubits along this path resulting in the next picture (Figure 5.3 (c)) where A and D are entangled. After this, if A wants to share entanglement with F, node D can perform the entanglement swapping operation between the newly created Bell pair and the Bell pair it previously shared with F. The outcome is shown in Figure 5.3 (d).

5.6.1 Physical link vs. virtual link

From the example in Figure 5.3 we note that, once a Bell pair is established between two nodes, it does not matter whether they are directly connected by a physical quantum link or not. They can use this newly established entanglement to perform quantum routing for any later request over the network. This brings in the concept of the virtual links. In a quantum network two nodes have a *virtual link* if they share a Bell pair between them. A virtual link can only be used once, because after the entanglement swapping operation on it, this link is destroyed and some other virtual link is create in different part of the network. Also a virtual link can be used to teleport quantum information between the connecting nodes, which also destroys the link. In contrast, a physical link always exists except for malfunctions. Two node might be connected using physical links, still they might not have a shared entanglement. However they can generate shared Bell pairs on demand by playing bipartite entanglement sharing protocol over the physical link. In Figure 5.3 (b) node C and F have a physical link that connects them (the black line), but they do not have a shared Bell state.

5.7 Routing graph

From the idea of virtual links in the previous section we see that, during the service of a single routing request over a quantum network, the most important factors are the virtual links. That is, at any time, two distant nodes can share an entanglement if there exist a set of virtual links that connects the nodes. For example in Figure 5.3 (c) when node A wants to create an entanglement with node E the virtual links between nodes A, D and D, E are used. Therefore, a good strategy is to carefully create some virtual links in the network such that any request to share Bell states between any two nodes can be served using a minimum number of entanglement swapping operations. For this, we might have to create a graph of virtual links over the network, which we call the routing graph, that has different connectivity than the underlying network graph. For example, in Figure 5.4 (a) we have a network graph that has 16 nodes connected in a cycle using physical links, and in (b) we have a routing graph, which has a different connectivity, on top of the same network, where each line denotes a Bell pair shared between the nodes.

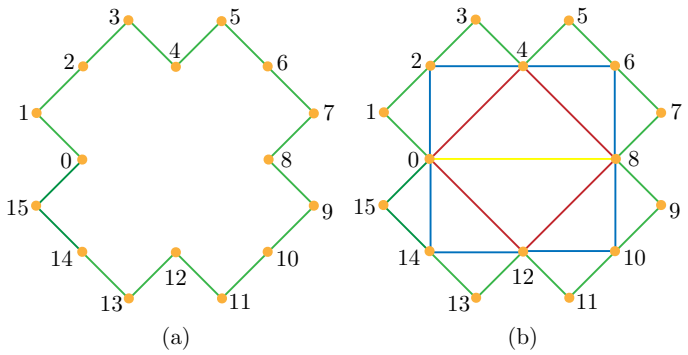


Figure 5.4: An example routing graph with 16 nodes - (a) Shows a network graph where 16 nodes are connected in a cycle with physical quantum links and (b) is one possible routing graph on it. Each link in (b) represents an Bell pair shared between two adjacent nodes.

While designing the routing graph for a network, one should try to minimise the number of entanglement swapping required to perform a routing request on it. That is, one might want to design a routing graph which has virtual links between

nodes that are a long distance apart on the network graph. However, virtual links require entangled qubits and qubits are scarce resources. Also creating the initial entanglements using the physical links takes time. Therefore, while designing a routing graph a trade-off has to be made between the routing efficiency (in terms of the number of entanglement swapping operation performed) and quantum memory requirements.

5.7.1 Replenishing the routing graph

Once a virtual link is used for routing, it gets destroyed. Therefore, if a routing protocol wants to take advantage of a pre-established routing graph, it must have an efficient way of replenishing the lost virtual links after serving each request. This efficiency might be measured in the number of time steps it takes to rebuild the routing graph. Note that, this replenishing happens after the routing request is served. Therefore, the efficiency criteria of the replenishing phase might be more relaxed than the criteria for routing efficiency.

5.8 Classical communication

A quantum network would still need classical communication among its nodes. For example a teleportation operation requires a classical message to be sent from the source to the destination. Also the quantum routing protocols would require many classical tags, such as, network address and entanglement swapping decisions, to be propagated throughout the network. Therefore, it is assumed that each pair of the nodes can exchange classical information between them without any error. This is a reasonable assumption, because we know how to build fault tolerant classical computer networks. It is safe to assume that, in a quantum network this classical communication happens through a parallel classical network.

5.9 Routing modes

Routing in an entanglement swapping based quantum network might happen in several modes.

5.9.1 Global routing

If there exists in the network an external agent, or a master node, that has the complete information of the state of the network at all times and makes all the routing decisions, then we call it a global routing scheme. Here, the state of the network at time t means the network graph and all the virtual links that are present at time t. With this information in hand, the agent can decide which path to take to serve a routing request. It can also decide which physical links should create shared Bell pairs if a pre-existing virtual path is not found.

This is a reasonable model, where the quantum network consists of a local cluster of quantum processors or memory units, for which, all the operations can be orchestrated by a *management unit*. A global routing scheme would have applications in a large quantum memory bank from where the qubits are teleported to a set of quantum processors using a quantum network.

5.9.2 Local routing

In a local routing scheme, each router has information of its own connectivity in the network graph, and remembers only the virtual links that are adjacent to it. In some schemes a router might have the virtual links information of a selected few neighbouring nodes. However, the routing decision is made by the router locally using the available partial information of the connectivity and the network address of the origin and destination nodes. This is a more reasonable mode of routing where the quantum network has many nodes spread over a large area because, for such a large network, it is impractical for each node to remember the connectivity of all the virtual links in the network, which might change over time.

5.9.3 Circuit routing

This mode of routing is useful when many shared Bell pairs are required between the source and the destination nodes. In a circuit routing, first a path (the circuit) from the source to the destination is found on the *network graph*. Then the virtual links are created along this path. After this, the swap operations are performed in the middle nodes. By repeating these steps over and over, many pairs of

entangled qubits can be created between the source and the destination. Note, that the virtual link vanishes after it is used once. So, in this mode of routing it is important that the nodes on the selected path are connected with physical links to allow repeated use.

5.10 Creating large distributed entangled states

So far, we have discussed sharing Bell pairs over the network among spatially separated nodes. We can use this network to create a large entangled state of n qubits $\rho_{1,2,3,\cdots,n}$ where each qubit is located in a separate node. To see this, let us assume node A, B and C want to create a 3 qubits state ρ_{ABC} where each of the nodes hold one qubit. We achieve this in the following steps.

1. Node A uses the quantum network to create entanglement with node B and C. Let us denote these two entangled pairs as $|\psi_{AB}\rangle$ and $|\psi_{AC}\rangle$ respectively.

2. Then Node A prepares the intended large state $\rho_{A\tilde{B}\tilde{C}}$ in it local registers. Here the $\tilde{B}$ and $\tilde{C}$ parts represent the qubits that should be held by node B and C.

3. Then Node A teleports qubit $\tilde{B}$ to node B using $|\psi_{AB}\rangle$ and qubit $\tilde{C}$ to C using $|\psi_{AC}\rangle$.

After these steps nodes A, B and C holds their respective share of the large entangled state ρ_{ABC}. This method can easily be generalised to create a large shared entangled state among any $n \geq 3$ nodes. Bose *et al.* [105] have studied some nontrivial techniques to share large entangled states among multiple nodes using entanglement swapping on Bell pairs.

5.11 Classical routing vs. quantum routing

In classical networks, most of the routing protocols take advantage of the fact that the data packets can be replicated [138]. While sending a packet from a source to a destination, the source node might save a copy of the packet so that, if some irreversible error occurs, then the node can resend the packet. If the next

node does not respond, the packet could be re-routed via other nodes. Whereas, unknown quantum states cannot be cloned [119]. Therefore, any routing protocol that requires the packets to be resent in case of a failure, cannot be adapted for a quantum network. In the routing scheme presented in this chapter, no actual data is transmitted through the middle nodes of the network. Rather, entanglement Bell pairs are distributed between distant paris of nodes. If one attempt to create an entanglement fails, then it can be attempted again. If some nodes in the middle fail, the path can be rerouted. That is, a different set of virtual links might be established, which upon entanglement swapping, allow the target and the destination to share a Bell pair. After this, the actual qubit is teleported to the destination. Therefore, no failure in initial routing attempt forces the quantum data to be resent. Thus, eliminating the necessity to copy an unknown quantum state.

5.12 Discussion

We have seen that, a router in the entanglement swapping based quantum network is functionally a quantum repeater. Therefore, along with distributing entanglement over the network it also provides all the advantages of using quantum repeaters for long distance quantum communication [121, 127]. Most of the primitive operations these routers use, such as Bell pair creation, entanglement purification and swapping cannot be performed perfectly using current technologies. However, we assume idealised versions of these primitives, such as perfect entanglement swapping operations and perfect entanglement creation between adjacent nodes. We also assume long coherence times for the quantum memories in the router. These idealisations are realistic in a sense, that elaborate subprotocols might be played to achieve shared Bell pairs which mitigates the effects of imperfect basic operations [120]. Moreover, it is reasonable to hope that future technological breakthroughs will closely approximate these idealised primitives. Thus, in our high level study we only focus on building efficient routing protocols using these idealised primitives. This is the subject of the next chapter.

6

Routing protocols for a quantum network

In this chapter, we give the first routing protocol for the entanglement swapping based quantum network networks described in Chapter 5. We discuss the protocol from a very high level, in a sense that we assume some idealised network primitives to build the protocol. Our protocol is a representative of a larger class of resource efficient quantum routing protocols that utilise a pre-established routing graph. We discuss the design principles for such routing graphs, and analyse various properties of a particular class of routing graphs that we use in a ring network. Then we give the routing protocol, prove its correctness, give bounds on its performance and resource requirements.

6.1 Designing quantum routing protocols

If node a and node b in a quantum network share a Bell pair, then they can teleport an arbitrary unknown qubit from one node to the other. Therefore, the problem of quantum routing from node a to node b is essentially the problem of sharing Bell pairs between them.

To achieve this we design a protocol where some virtual links (Bell states) are pre-established before any routing request arrives in the network. These virtual links give rise to a routing graph (Chapter 5.7) that allows efficient quantum routing between any pair of nodes. The design of the routing graph depends on

the size and the topology of the network graph. However, we want the routing graph to have the following general properties.

1. The routing graph has to be of low degree, so that a router does not need a very large quantum memory.

2. The routing graph should have lower diameter compared to the network graph, so that an optimal route needs few entanglement swapping operations.

3. The routing graph should allow local routing, so that the nodes do not need to remember the whole routing graph.

4. The routing graph should allow the protocol to minimise the associated classical communication overload.

5. The routing graph should allow the protocol to replenish it (the routing graph) efficiently after the loss of some virtual links.

With these considerations in mind, we give a routing graph construction and an associated sufficient routing protocol where the nodes are physically connected in a ring.

6.2 The ring network

We consider a quantum network where m nodes are connected by physical quantum links in a cycle (appendix A.1). We refer to this network as a *ring* of size m. This network has nice properties such as it has only m physical quantum links and it has high rotational symmetry.

The symmetry of the ring greatly simplifies the construction and analysis of the routing graphs and the associated protocol. Moreover, a ring is a practical and a fairly common network structure. This gives us a good starting point for designing quantum routing protocols.

6.2.1 The goal

On a ring network of size m, the most naive protocol would be to pre-establish Bell pairs (virtual links) along each of the edges. Once these virtual links are established any pair of nodes in the ring would be connected by two paths. We call one *clockwise* and other the *anti-clockwise* path. If node A wants to share Bell pair with node B, then it picks the shortest among these two paths and uses it for routing.

Since, each of the middle node have to perform an entanglement swapping operation, this protocol would require $O(m)$ entanglement swapping to serve a request in the worst case. However, it requires only a constant quantum memory (namely 2 qubits for the two links) at each node.

We want to minimise the number or entanglement swapping to $O(\log m)$ without blowing up the quantum memory requirement too much. Note that, even though the underlying physical links are connected in a ring, we could create any m vertex routing graph on top of this network. Therefore, a carefully designed routing graph would allow us to reduce the number of entanglement swappings required in the worst case, find the shortest path efficiently and allow efficient replenishing of the lost virtual edges of the routing graph.

More specifically, for an m node ring network, we give a routing graph that has a diameter $O(\log m)$ and where each node has a maximum degree $O(\log m)$. The associated protocol for finding the optimal path runs in $O(\log m)$ and to find the optimal route the nodes can make only local decisions depending on the source and destination addresses of the routing request. The network address for each node is $O(\log m)$ and the classical memory requirement for each node is $O(\log^3 m)$. Since the routing graph has maximum degree $O(\log m)$, the quantum memory requirement for each node is $O(\log m)$.

6.2.2 The ring routing graph

Now, we design a routing graph for a ring network of size 2^n where $n \in \mathbb{N}$ and analyse its various properties. Based on these analyses, we design our routing protocol. Later, we show how this protocol can be used to perform routing on a ring network of any size $m \in \mathbb{N}$.

We represent the physical connectivity of the ring network using a cycle $C_{2^n} = (V_{C_{2^n}}, E_{C_{2^n}})$, where the vertex set $V_{C_{2^n}} = \{0, 1, \ldots, 2^n - 1\}$ represents the nodes, and the edges set $E_{C_{2^n}} = \{a, b \in V_{C_{2^n}} \text{and } \{a, b\} : |a - b| \equiv 1 \pmod 2^n\}$ represents the physical quantum links. In graph theoretic terms, this cycle has a diameter $O(2^n)$ (see, Appendix A.5). Thus a naive protocol would require $O(2^n)$ entanglement swapping operations to serve a routing request. However, we want to create a routing graph, on top of this ring, that has a diameter and quantum memory requirement logarithmic in the number of nodes, and allows the routers to make local routing decisions, to achieve the global optimal route.

For the ring $C_{2^n} = (V_{C_{2^n}}, E_{C_{2^n}})$ of size 2^n, we design a routing graph $G_n = (V_n, E_n)$, which we call the *routing graph of order* n, where the vertex set $V_n = V_{C_{2^n}}$ and the edge set E_n contains the virtual edges. However, before giving the formal definition of the routing graph we need to introduce two useful functions p and $\gcd_2$.

We define a function $p : V_n \mapsto \mathbb{N} \cup \{0\}$ such that,

$$p(a) := \begin{cases} n & \text{if } a = 0, \\ \max\{k : 2^k \text{ divides } a\} & \text{otherwise.} \end{cases} \tag{6.1}$$

This function counts the number of 2's in the prime factorisation of a. We also define another function $\gcd_2 : V_n \times V_n \mapsto \mathbb{N}$ such that,

$$\gcd_2(a, b) = 2^{\min\{p(a), p(b)\}}. \tag{6.2}$$

As the notation suggests, integer $\gcd_2(a, b)$ is the largest power of 2 that divides both a and b. Since, $\gcd_2(0, 0) = 2^n$, strictly speaking, $\gcd_2$ is also a function of n. However, this special case is not of much importance to us. Therefore, for the sake of simplicity, we would not make this dependence explicit. And leave it to be understood from the context.

Now, we are ready to define our routing graph for the ring network.

Definition 12. A graph $G_n = (V_n, E_n)$ is called a ***routing graph of order*** n where the vertex set $V_n = \{0, 1, \ldots, 2^n - 1\}$ represents the nodes and the edge set $E_n = \{\{a, b\} : a, b \in V_n, \text{and } |a - b| \equiv \gcd_2(a, b) \pmod{2^n}\}$ represents the virtual links.

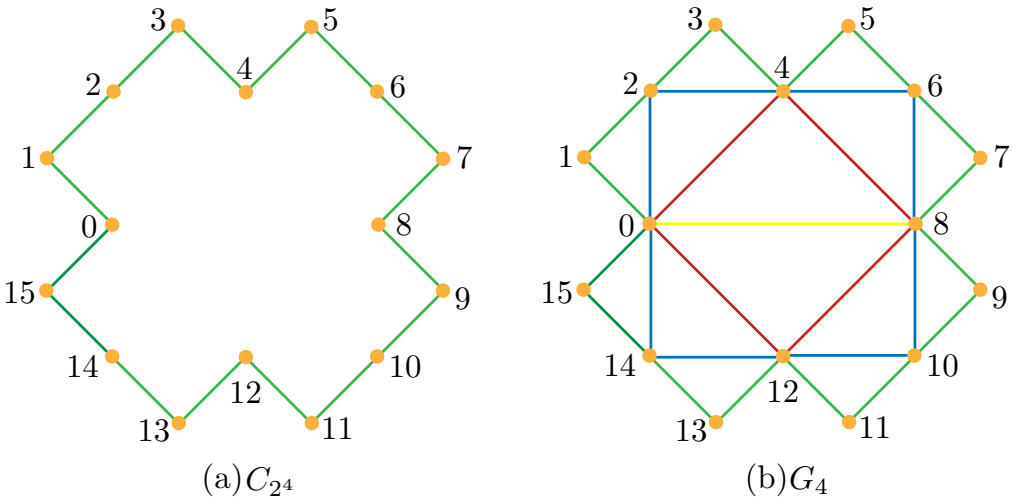


Figure 6.1: Network graph vs. routing graph - (a) is the network graph C_{2^4} and (b) is the corresponding routing graph G_4

For example, in Figure 6.1 (a) we illustrate the ring C_{2^4}, which is a network graph of size 16, and in Figure 6.1 (b) we illustrate the routing graph G_4 of order 4 that has the same nodes as C_{2^4}, but different connectivity.

A close observation of the routing graphs G_2, G_3 and G_4 side by side as in Figure 6.2 reveals that if we remove all the odd vertices of G_4 then we get a graph that is isomorphic (Appendix A.6) to G_3. Similarly, removing all the odd vertices from G_3 would give us a graph isomorphic to G_2. We generalise this observation for any G_n in the following lemma.

Lemma 15. *For all $n \geq 2$, the subgraph induced by the even vertices of G_n is isomorphic to G_{n-1} via mapping each even vertex a of G_n to the vertex $a/2$ of G_{n-1}.*

Proof. Let $H_{n-1} = (\tilde{V}_{n-1}, \tilde{E}_{n-1})$ be the subgraph induced by the even vertices of G_n. To establish the lemma, we show that H_{n-1} is isomorphic to G_{n-1} via the bijection $f : \tilde{V}_{n-1} \mapsto V_{n-1}$ defined by

$$f(a) = a/2 \tag{6.3}$$

To accomplish this, we need to establish that for all $a, b \in \tilde{V}_{n-1}$ we have $\{a, b\} \in \tilde{E}_{n-1}$ if and only if $\{f(a), f(b)\} \in E_{n-1}$.

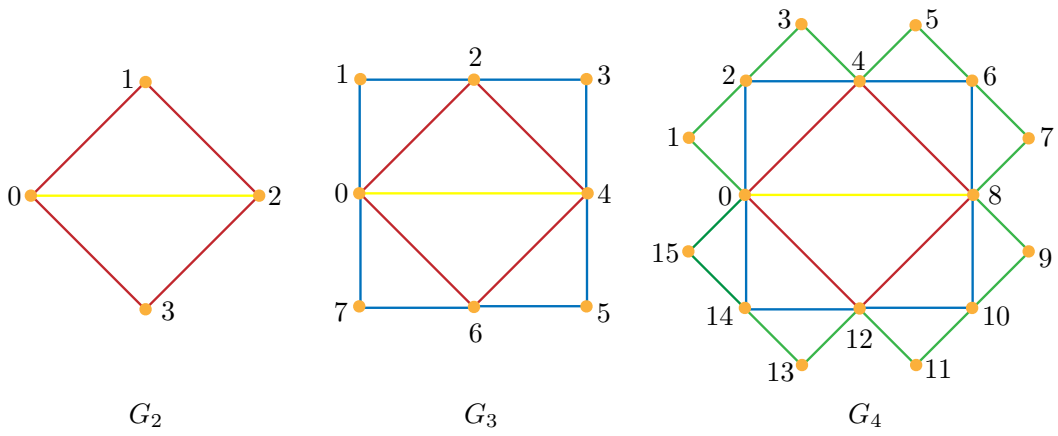


Figure 6.2: Routing graphs G_2, G_3 and G_4 - Each larger routing graph contains all the smaller graphs as a subgraphs in it

For this we first note that according to the definition of an induced subgraph (Appendix A.8), for all $a, b \in \tilde{V}_{n-1}$ we have $\{a, b\} \in \tilde{E}_{n-1}$ if and only if $\{a, b\} \in E_n$.

This means according to the Definition 12 for all $a, b \in \tilde{V}_{n-1}$ for each $\{a, b\} \in E_{n-1}$ if and only if

$$|a - b| \equiv \gcd_2(a, b) \pmod{2^n}, \tag{6.4}$$

$$\Leftrightarrow \left|\frac{a}{2} - \frac{b}{2}\right| \equiv \frac{\gcd_2(a, b)}{2} \pmod{2^{n-1}}, \tag{6.5}$$

$$\Leftrightarrow |f(a) - f(b)| \equiv q\big(f(a), f(b)\big) \pmod{2^{n-1}}, \tag{6.6}$$

where the last equivalence follows from the fact that a and b are distinct even vertices. Therefore, we have

$$\frac{\gcd_2(a, b)}{2} = q\left(\frac{a}{2}, \frac{b}{2}\right) = \gcd_2(f(a), f(b)). \tag{6.7}$$

To complete the proof, it remains to note that according to the definition of G_{n-1} (Definition 12) Equation (6.6) holds if any only if $\{f(a), f(b)\} \in E_{n-1}$. □

6.2.3 Sub-routing-graphs

Lemma 15 shows that the routing graphs have a nice hierarchical structure, where a subgraph isomorphic to the routing graph G_{n-1} can be found in the routing

graph G_n and G_{n-2} in G_{n-1} and so on. This gives rise to a useful idea that we call sub-routing-graphs.

Definition 13. For $n, m \in \mathbb{N}$ where $m \leq n$ and for G_n, the routing graph of order n, a ***sub-routing-graph*** $H_m(G_n) = (\tilde{V}_m, \tilde{E}_m)$ is the vertex induced subgraph of G_n, induced by vertex set $\tilde{V}_m = \{1 \times 2^{n-m}, 2 \times 2^{n-m}, \ldots, (2^m - 1) \times 2^{n-m}\}$ that is isomorphic to the routing graph G_m of order m.

Note that, we usually do not show the parameter G_n from $H_m(G_n)$ and simply write H_m where the parameter G_n is understood from the context.

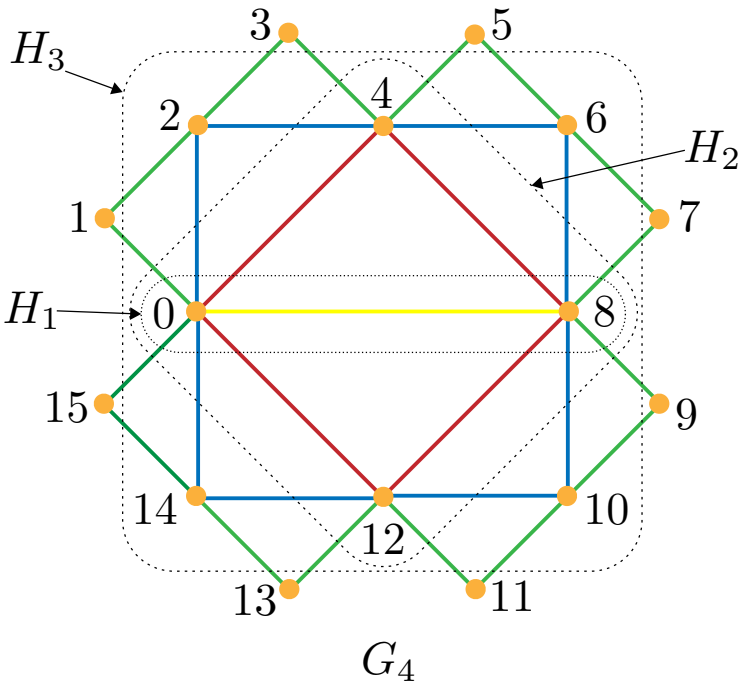


Figure 6.3: Sub-routing-graphs of G_4 - The sub-routing-graphs H_3, H_2, H_1 are shown enclosed in dashed rounded rectangles.

For example, Figure 6.3 shows routing graph G_4 and all of its sub-routing-graphs H_3, H_2 and H_1.

For two sub-routing-graphs H_m and H_k of routing graph G_n, if $m < k$, then H_m is called an ***inner sub-routing-graph*** compared to H_k and H_k an ***outer sub-routing-graph*** compared to H_m.

We also define an *outer node* (or an *outer vertex*) of a sub-routing-graph to be the vertex that has only 2 adjacent nodes within that sub-routing-graph. Formally,

Definition 14. For a routing graph G_n and any of its sub-routing-graph H_m, a vertex a is called an ***outer node*** or ***outer vertex*** of H_m if a is on H_m and it does not have more than 2 adjacent nodes that are also on H_m.

For example, in Figure 6.3 node 2 is an outer node of H_3 because it has only two neighbours within H_3, namely node 0 and 4. Node 4 is not an outer node of H_3. However, it is an outer node or H_2.

Now we characterise an outer node in terms of the function p (defined in (6.1)) that counts the number or 2's in the prime factorisation of any node a. Later, we use this characterisation to design and analyse our routing protocol.

Lemma 16. *If a is a node in a routing graph G_n such that $p(a) = k$, then a is an outer node of the sub-routing-graph H_{n-k} of G_n. If $a = 2^k t$ for some $t \in \mathcal{N}$, then $2^k(t+1 \pmod{2^n})$ and $2^k(t-1) \pmod{2^n}$ are the only two vertices on H_{n-k} that are adjacent to a.*

Proof. We prove this by showing that a is on sub-routing-graph H_{n-k} and then we give two other nodes adjacent to a that are also on H_{n-k}. We complete the proof by showing that any other edge from a connects to nodes that are not on H_{n-k}.

Now, from definition of p we have,

$$a = 2^k t, \tag{6.8}$$

where c is a positive odd integer and $t < 2^k$.

a can also be written as $a = 2^{n-(n-k)}t$, which from the definition of sub-routing-graph (Definition 13), allows us to conclude that a is on sub-routing-graph H_{n-k}.

Now consider nodes $b = 2^k(t+1) \pmod{2^n}$ and $c = 2^k(t-1) \pmod{2^n}$ from the definition of sub-routing-graph (Definition 13) we see that both of these nodes are also on sub-routing-graph H_{n-k}. And from the definition of routing graph (Definition 12) we see that $\{a, b\}$ and $\{a, c\}$ are both edges in the routing graph G_n therefore they are also in H_{n-k}.

Finally, we note that any other node that are adjacent to the node a must have the form $2^{k'}(t' \pm 1)$ where $k' < k$. However, these nodes are on sub-routing-graph $H_{n-k'}$, which is an outer sub-routing graph compared to H_{n-k}. This completes the proof. □

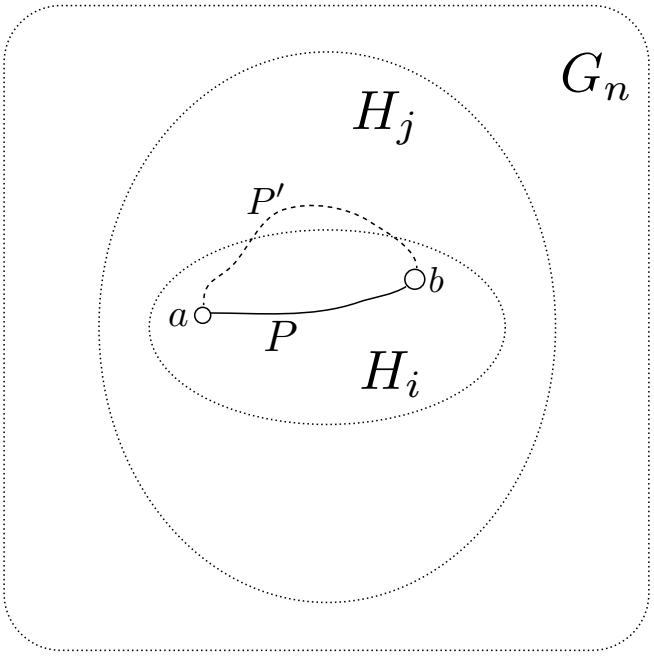


Figure 6.4: Paths going through sub-routing-graphs of G_n - Path P connecting a and b goes through H_i. Any path P' that goes through an outer sub-routing-graph H_j should be at least as long as P.

Now we are ready to show a very interesting property of distances between any pair of nodes in a routing graph. To be more precise, we show that, for a routing graph G_n, if two nodes a and b are on some sub-routing-graph H_i then, distance between a and b in H_i is the same as the distance between a and b in H_j for any $i \leq j$. To understand this let us consider Figure 6.4. Here, node a and b are on the sub-routing-graph H_i. An optimal path P connecting a and b is shown using a solid curve that goes only through H_i. The length of this path should be $d_{H_i}(a, b)$. Now we claim that even if the path is allowed to use edges that go

through the outer sub-routing-graph H_j, no shorter path can be found. That is, paths P' shown using dashed curve that goes through outer sub-routing-graph H_j, cannot be shorter than P. This is a very important observation because the construction and correctness proof of our routing protocol heavily depends on it. We prove this claim as a corollary of the following lemma.

Lemma 17. *For any $n, k \in \mathbb{N}$ and any two nodes a and b of G_n we have, $d_{G_n}(a, b) = d_{G_{n+k}}(2^k a, 2^k b)$.*

Proof. From Lemma 15 we know that a routing graph G_n recursively contains all the smaller routing graphs $G_{n-1}, G_{n-2}, \ldots, G_1$ in it. The distance between a and b in G_n will be the same as the distance between the nodes $2^k a$ and $2^k b$ in G_{n+k} because these are the same nodes in the sub-routing-graph of G_{n+k}, which is isomorphic to G_n.

Let us first establish the $k = 1$ case. It will be useful for us to consider the graph $H_n = (\tilde{V}_n, \tilde{E}_n)$ induced by the even vertices of G_{n+1}. From Lemma 15 we know that G_n is isomorphic to H_n via mapping a vertex a of G_n to the vertex $2a$ of H_n. Therefore, $d_{G_n}(a, b) = d_{H_n}(2a, 2b)$ for all $a, b \in V_n$ and to get that $d_{G_n}(a, b) = d_{G_{n+1}}(2a, 2b)$, it suffices to show that $d_{H_n}(2a, 2b) = d_{G_{n+1}}(2a, 2b)$. We accomplish this, using a proof by contradiction. For this, assume that $d_{H_n}(a, b) > d_{G_{n+1}}(a, b)$ for some $a, b \in \tilde{V}_n$. Since H_n is an induced subgraph of G_{n+1}, our assumption implies that any optimal path P_{ab} in G_{n+1} must contain vertices from $V_{n+1} \setminus \tilde{V}_n$.

Let us consider the subpaths $P_{a'b'}$ of P_{ab} which start and end with with vertices from H_n but have all other vertices belonging to $V_{n+1} \setminus \tilde{V}_n$ (Figure 6.5). For at least one of these subpaths $P_{a'b'}$ it must be that $d_{H_n}(a', b') > d_{G_{n+1}}(a', b')$ as otherwise we could replace all of them with paths of the same length contained entirely in H_n. Let us now focus on some such $P_{a'b'}$.

Since any $k \in V_{n+1} \setminus \tilde{V}_n$ is odd and no two odd vertices are adjacent, we conclude that the path $P_{a'b'}$ has length 2, i.e., $P_{a'b'} = (a', c, b')$ for some odd c. This implies that $d_{G_{n+1}}(a', b') = 2$. Furthermore, since any odd vertex c is adjacent to only $c + 1 \pmod{2^{n+1}}$ and $c - 1 \pmod{2^{n+1}}$, we obtain $|a' - b'| \equiv 2 \pmod{2^{n+1}}$. Combining this with the fact that a' and b' are even nodes, we see that a' is adjacent to b' in H_n. That is, $d_{H_n}(a', b') = 1 < 2$. This is a contradiction. Therefore, $d_{G_{n+1}}(2a, 2b) = d_{H_n}(2a, 2b) = d_{G_n}(a, b)$ for any $a, b \in V_n$.

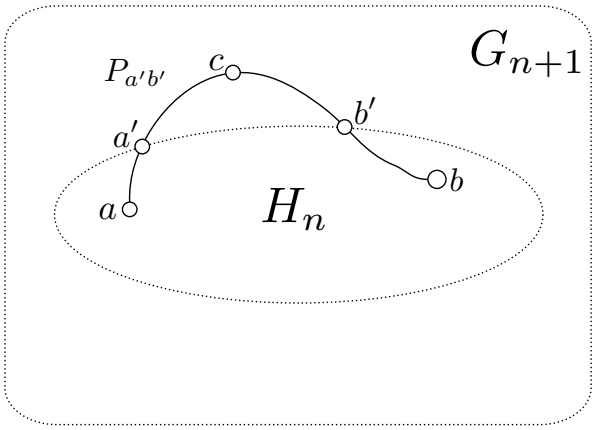


Figure 6.5: An optimal path connecting a and b through nodes not in H_n - There exist a segment $P_{a'b'} = a' \dots c \dots b'$ that exits H_n at a' and reenters at b'. Here c is an odd vertex implying sections $a' \dots c$ and $c \dots b'$ of this subpath are edges in G_{n+1}

Now that we have established the $k = 1$ case, observe that for any k we have

$$d_{G_n}(a, b) = d_{G_{n+1}}(2a, 2b) = d_{G_{n+2}}(2^2a, 2^2b) = \dots = d_{G_{n+k}}(2^k a, 2^k b), \tag{6.9}$$

which completes the proof. □

Now our intended result about the path lengths between nodes on a sub-routing-graph follows as a corollary of this lemma.

Corollary 1. *For $n, i, j \in \mathbb{N}$, routing graph G_n of order n and nodes a and b on G_n, if $i \leq j \leq n$ and nodes a and b are on sub-routing-graph H_i then $d_{H_i}(a, b) = d_{H_j}(a, b)$.*

Proof. From definition of sub-routing-graphs (Difinition 13) we know that H_i is isomorphic to G_i where a in H_i maps to $a/2^{(n-i)}$ in G_i. Similarly H_j is isomorphic to G_j where a in H_j maps to $a/2^{(n-j)}$. From Lemma 17, we get,

$$d_{H_i}(a,b) = d_{G_i}\left(\frac{a}{2^{(n-i)}}, \frac{b}{2^{(n-i)}}\right), \tag{6.10}$$

$$= d_{G_j}\left(\frac{a2^{(j-i)}}{2^{(n-i)}}, \frac{b2^{(j-i)}}{2^{(n-i)}}\right), \tag{6.11}$$

$$= d_{G_j}\left(\frac{a}{2^{(n-j)}}, \frac{b}{2^{(n-j)}}\right), \tag{6.12}$$

$$= d_{H_j}(a,b), \tag{6.13}$$

where, Equation (6.11) and Equaiton (6.13) follows from Lemma 17. □

We are now ready to bound the diameter (Appendix A.5) of the routing graph G_n. This is important because the diameter corresponds to the number of necessary entanglement swaps performed in the worst case, to allow any two nodes to communicate. We utilise the hierarchical structure of the routing graph that was shown in Lemma 15 and the fact that going through outer sub-routing-graphs does not decrease the distance between two nodes as shown in Corollary 1.

Lemma 18. *For a routing graph G_n of order n, we have that $D(G_n) \leq D(G_{n-1})+2$ and and $D(G_1) = 1$ where $D(G)$ is the diameter of a graph G In particular, we have $D(G_n) = O(\log(|V_n|)) = O(n)$.*

Proof. From Lemma 15 we know, subgraph H_{n-1} induced by the set of even vertices $\tilde{V}_{n-1}$ of G_n is isomorphic to G_{n-1}. Therefore,

$$D(H_{n-1}) = D(G_{n-1}). \tag{6.14}$$

For any vertex v of G_n, we consider an even vertex v'. We let $v' := v$ if v is even and we let $v' := v + 1 \pmod{2^n}$ if v is odd. In the latter case, we have that $\gcd_2(v, v') = 1$ and hence it follows from Definition 12 that $\{v, v'\} \in E_n$. Thus, for any vertex v of G_n, we have $d(v, v') \leq 1$ and $v' \in H_{n-1}$. Using this simple observation we now show that for any two vertices $a, b \in V_n$ the distance between

a and b in G_n is,

$$d_{G_n}(a,b) \leq (a,a') + d_{G_n}(a',b') + d(b,b'), \tag{6.15}$$
$$= d(a,a') + d_{H_{n-1}}(a',b') + d(b,b'), \tag{6.16}$$
$$\leq dH_{n-1}(a',b') + 2, \tag{6.17}$$
$$\leq D(H_{n-1}) + 2, \tag{6.18}$$
$$= D(G_{n-1}) + 2. \tag{6.19}$$

Here, (6.16) follows from Corollary 1. Since for any two vertices a and b of G_n we have $d_{G_n}(a,b) \leq D(G_{n-1}) + 2$, it follows directly from the definition of diameter that $D(G_n) \leq D(G_{n-1}) + 2$.

Now, using the fact that $D(G_1) = 1$ we get, $D(G_n) = O(n) = O(\log(|V_n|))$. □

6.2.4 A recursive construction

The existence of sub-routing-graphs $H_{n-1}, H_{n-2}, \ldots, H_1$, which are isomorphic to $G_{n-1}, G_{n-2}, \ldots, G_1$, in the routing graph G_n provides us a way of constructing the routing graph recursively. This recursive construction gives us some structural insights. Moreover, similar recursive constructions might be used for constructing various types of routing graphs for different network topologies.

For this, first we define an operation $\mathbf{ND}(.)$. For any routing graph $G_n = (V_n, E_n)$, this operation creates a new graph $\mathbf{ND}(G_n) = G'_n = (V'_n, E'_m)$ where vertex set $V'_n = \{2a : a \in V_n\}$ and edge set $E'_n = \{\{2a, 2b\} : \{a, b\} \in E_n\}$.

Now we give the recursive construction of the routing graph G_n.

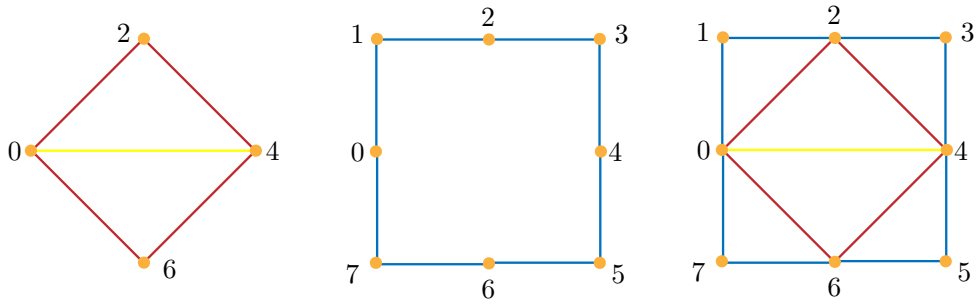
Base step:

$$G_1 = (\{0,1\}, \{\{0,1\}\}). \tag{6.20}$$

Recursive step:

$$G_n = \mathbf{ND}(G_{n-1}) \cup C_{2^n}. \tag{6.21}$$

Where $C_{2^n} = (V_{C_{2^n}}, E_{C_{2^n}})$ is the cycle graph of 2^n elements with vertex set $V_{C_{2^n}} = \{0, 1, \ldots, 2^{n-1}\}$ and edge set $E_{C_{2^n}} = \{\{a,b\} : a, b \in V_{C_{2^n}} \text{and } a = b + 1 \pmod{2^n}\}$ and $\cup$ is the graph union operation (Appendix A.9).



(a) $\mathbf{ND}(G_2)$ (b) C_{2^2} (c) $G_3 = \mathbf{ND}(G_2) \cup C_{2^2}$

Figure 6.6: Recursively constructing G_3 from G_2 and C_{2^2} - The recursive step is shown here. $\mathbf{ND}(G_2)$ is unioned with C_{2^2} to get $G_3 = \mathbf{ND}(G_2) \cup C_{2^2}$.

We show this construction with an example in Figure 6.6 where routing graph G_3 is built from G_2.

To see why this is an equivalent construction to Definition 12, one should note that $\mathbf{ND}(G_{n-1})$ is essentially the sub-routing-graph H_{n-1} of G_n.

6.2.5 Similarity with the overlay networks in classical distributed computing

From the construction of the routing graphs for the ring network in Definition 12 one can note some structural similarity with the Chord [139] or Pastry [140] overlay networks [141, 142] studied in the field of classical distributed computing [143]. However, we should note that, even thought the graphs look similar, one cannot replace the ring-routing graph with these overlay networks to achieve the same quantum routing goals. To be more precise, the virtual links in the overlay networks are determined by the routing tables stored in each node. If a node stores the address of a remote node in the overlay layer, then it is considered to have a virtual link to that remote node. However, in the quantum routing graph the virtual links represents some physical resources pre-distributed over the network, namely the entangled Bell pairs. Unlike the virtual links in the classical overlay network the quantum virtual links can only be used once. After

that, they have to play some background protocol steps to recreate a shared Bell pairs, thus recreate the lost virtual links. This constraint of replenishment is absent in the classical network where the only goal is to achieve resiliency against network failures. Therefore, the protocols, and analysis techniques used in the study of classical distributed computing are not directly applicable in the context of entanglement swapping based quantum routing studied in this thesis.

6.3 Routing protocol for the ring network

The routing graph that we have constructed for a ring network has highly regular structure. Specially, Lemma 15 shows that a 2^n node routing graph $G_n = (V_n, E_n)$ recursively contains all the smaller sub-routing-graphs $H_{n-1}, H_{n-2}, \ldots, H_1$, which are isomorphic to $G_{n-1}, G_{n-2}, \ldots, G_1$ respectively.

There can only be $O(\log(|V_n|))$ such inner sub-routing-graphs. If a protocol for routing starts traversing from both the source and the destination and at each move goes to the innermost sub-routing-graph possible then they will eventually arrive at H_1 and meet each other. This would allow for a routing protocol that gives a path of length $O(\log(|V_n|) = O(n)$. However, this is not always optimal. A slight modification, namely, checking the condition if the current traversal head and tail have a common neighbour before making any move would give rise to the optimal routing protocol. We prove this in Theorem 8.

We give the pseudocode of our routing protocol in Protocol 8: RinglogRoute, that takes as input, the source and destination node IDs a and b respectively and a number n that indicates the routing graph G_n and outputs an optimal path P connecting a and b in G_n.

Theorem 8. *For any $n \in \mathbb{N}$ and source node a and destination node b the protocol RinglogRoute(a, b, n) outputs an optimal path P connecting a and b in G_n.*

Proof. In Protocol 8: RinglogRoute the traversal starts from both the source and destination nodes. The current node of traversal starting from the source is the **head** and the current node of traversal starting from destination is the **tail**. The **head** and the **tail** traverse the network according to the protocol and when they meet each other the protocol outputs the path constructed from their movement.

Protocol 8: RinglogRoute

input : $a, b \in \{0, 1, \ldots, 2^n - 1\}$ the source and destination nodes respectively, n indicating the routing graph G_n

output: P: a list containing the shortest path from a to b in G_n

assign **head** := a, **tail** := b
for $i = 0$ *to* $n - 1$ **do**
  A.append(**head**)
  B.append(**tail**)
  if **head** *and* **tail** *are adjacent* **then**
    break
  else if **head** *and* **tail** *have a common neighbour* **then**
    A.append(common-neighbour's ID)
    break
  else
    $a_{\text{left}} := \textbf{head} + 2^i \pmod{2^n}$,
    $a_{\text{right}} := \textbf{head} - 2^i \pmod{2^n}$,
    if $p(\textbf{tail}) == i$ **and** $p(a_{left}) > p(a_{right})$ **then**
      // p is defined in Equation (6.1). If **head** is an outer node of $(n-i)$th sub-routing graph then move
      head := a_{left},
      A.append(**head**)
    else if $p(\textbf{head}) == i$ **then**
      head := a_{right},
      A.append(**head**),
    $b_{\text{left}} := \textbf{tail} + 2^i \pmod{2^n}$,
    $b_{\text{right}} := \textbf{tail} - 2^i \pmod{2^n}$,
    if $p(\textbf{tail}) == i$ **and** $p(b_{left}) > p(b_{right})$ **then**
      tail := b_{left},
      B.append(**tail**),
    else if $p(\textbf{head}) == i$ **then**
      tail := b_{right},
      B.append(**tail**) ,
  $P = A$.append(reverse(B)) // construct the complete path
  Output:P

We show the correctness of the protocol by showing that at each step any move made by **head** or **tail** going to some **newhead** or **newtail** respectively satisfies the loop invariant,

$$\begin{aligned} d(\mathbf{head}, \mathbf{tail}) =& d(\mathbf{head}, \mathbf{newhead}) + d(\mathbf{newhead}, \mathbf{newtail}) \\ &+ d(\mathbf{newtail}, \mathbf{tail}). \end{aligned} \tag{6.22}$$

That is, **newhead** and **newtail** are also on an optimal path connecting **head** and **tail**. Since we start with **head** $= a$ and **tail** $= b$, this equation (6.22) proves that the final path P constructed from the traversal of **head** and **tail** is optimal.

Let us consider some properties of the traversal from the **head**. The traversal from the **tail** would have the same properties. Note that in the ith step of the **for** loop of Protocol 8 the **head** is moved only if

$$p(\mathbf{head}) = i. \tag{6.23}$$

From Lemma 16, **head** must be an outer node of the sub-routing-graph H_{n-i}. If **tail** is also on H_{n-i}, then from Corollary 1 there exists an optimal path connecting **head** and **tail** that entirely lies in H_{n-i}. The only two nodes that are adjacent to **head** and also in H_{n-i} are,

$$a_{\text{left}} = head + 2^i \pmod{2^n}, \tag{6.24}$$

and

$$a_{\text{right}} = head - 2^i \pmod{2^n}. \tag{6.25}$$

Therefore, an optimal path from **head** to **tail** must go through one of these two nodes. If we show that the **newhead**, chosen from these two nodes by the protocol, does not increase the path length, then this will prove the loop invariant (6.22).

For the clarity of appearance, from now on we do not explicitly write (mod 2^n) in the equations. However, we should assume that all the addition and subtractions are performed modulo 2^n.

Now, from definition of the function p (Equation (6.1)) and the fact that $p(\mathbf{head}) = i$, we can write,

$$head = 2^i c, \tag{6.26}$$

where $c \in \mathbb{N}$ is an odd integer.

Using this a_{left} and a_{right} can be written as,

$$a_{\text{left}} = head + 2^i = 2^i(c+1), \tag{6.27}$$

$$a_{\text{right}} = head - 2^i = 2^i(c-1). \tag{6.28}$$

Let us also define integer l and u such that,

$$c = 2^l u + 1, \tag{6.29}$$

where $l = p(c-1)$.

Now from Equation (6.27) and Equaiton (6.29) we have,

$$a_{\text{left}} = 2^{i+l}u. \tag{6.30}$$

And from Equation (6.28) and Equation (6.29) we have,

$$a_{\text{right}} = 2^{i+1}(2^{l-1}u - 1). \tag{6.31}$$

If $p(a_{\text{left}}) > p(a_{\text{right}})$, then the **head** moves to the node **newhead** $= a_{\text{left}}$. For the loop invariant (Equation (6.22)) to hold, we have to show that moving to the other node a_{right} does not give any advantage.

To see this, if the **newhead** was a_{right} instead of a_{left}, then on the very next move it either goes to $2^{i+l}u$, which is just a_{left}, (this can be seen from in (6.30)) or to $a' = 2^{i+1}(2^{l-1u-2})$, which can be reached from the a_{left} in one step. Either way, the move to a_{right} does not give any advantage over moving to a_{left}. This is illustrated with an example in Figure 6.7.

In the other case, if $p(a_{\text{left}}) \leq p(a_{\text{right}})$, then **head** moves to **newhead** $= a_{\text{right}}$. A similar argument to the previous case shows that the alternative move in this case ,which takes the **head** to a_{left}, does not give any advantage.

Note that, for these arguments to hold we have considered two consecutive moves of **head**. The protocol makes sure that the **head** and **tail** are at least 2 distance apart before making this move. This is done by checking beforehand whether they are neighbours, or have a common neighbour.

To ensure that **head** and **tail** are at least 2 distance apart while making the choice between a_{left} and a_{right} we have the if condition in line 7 of Protocol 8.

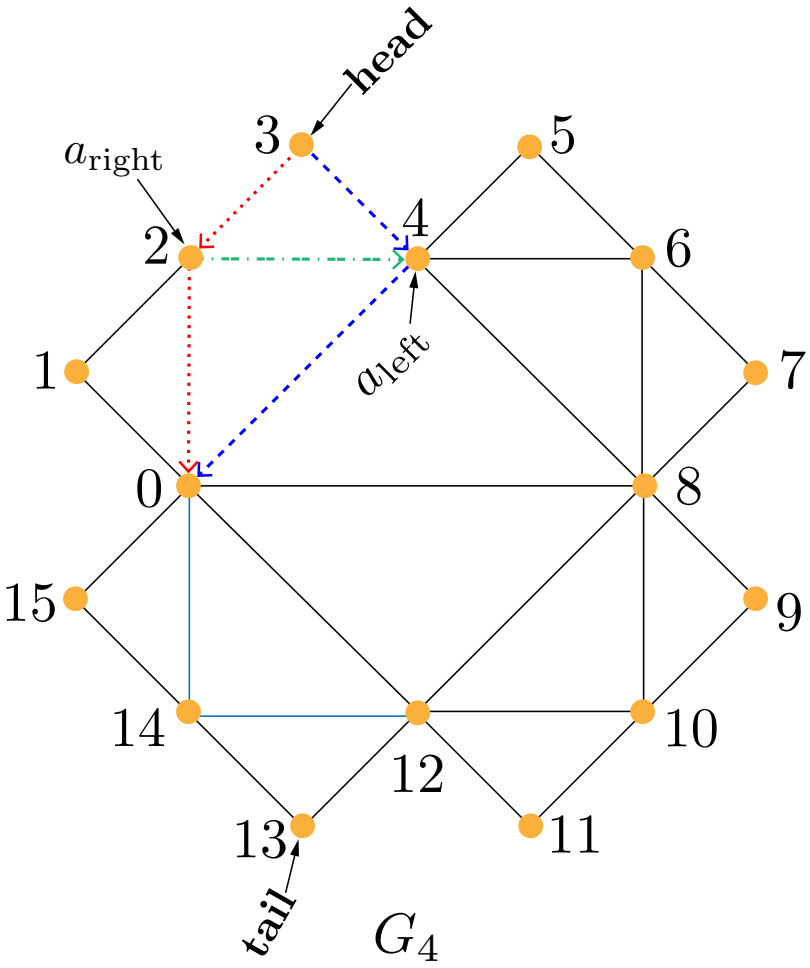


Figure 6.7: Optimal move for head in the ring routing protocol on G_4 - One of the nodes a_{left} and a_{right} is on the optimal path from **head** to **tail**. Since $p(a_{\text{left}}) > p(a_{\text{right}})$ the protocol would move **head** from node 3 to $a_{left} = 4$ (blue dashed line). If the head was moved to a_{right} instead (red dotted line), then on the next move the **head** would return to node 4 (green dash-dotted line) or go to 0, which is reachable from 4 in one step. This implies taking the red dotted path cannot decrease the distance more than taking the blue dashed path. Therefore, moving to a_{left} cannot be non-optimal.

A similar reasoning also applies on the movement of the **tail** to a new node **newtail** during the protocol.

From the **for** loop in Protocol 8 we see that the **head** and **tail** start at a and b respectively. And in ith steph when either of them is moved they are on sub-routing-graph H_{n-i} moreover the moved head or tail are outer nodes of H_{n-i}.

This shows that the loop invariant of Equation (6.22) holds throughout the protocol whenever **head** or **tail** makes a move as an outer node. This completes the proof.

□

6.3.1 Replenishing the used virtual links

After serving any routing request, the routing graph would lose some of its virtual links. One way to replenish a virtual link is to recursively replenish a longer virtual link by performing entanglement swapping on shorter links, until we reach the outer cycle that has the physical quantum links. The virtual links that are along this outer cycle can be replenished by directly creating Bell pairs using the physical quantum links.

For a routing graph G_n with 2^n nodes, virtual links between the nodes a and b can be replenished using the following two recursive steps.

1. If a and b are connected by a physical link, then replenish the virtual link by sharing a Bell pair using this link and halt.

2. Denote $c := (a + b + 2^n) \pmod{2^n}$ Perform entanglement swapping on the virtual links $\{a, c\}$ and $\{c, b\}$ to create the virtual link $\{a, b\}$.

3. Replenish the virtual links $\{a, c\}$ and $\{c, b\}$.

Note that all these replenishing steps happen after the original routing request has already been served. Therefore, they do not affect the service time of a request. However, this replenishing phase has to complete before new requests can be served. Therefore, it affects the frequency with which the new requests can be served.

6.3.2 Resource requirements for the routing protocol

Our routing protocol for a ring network of size 2^n that uses a routing graph $G_n = (V_n, E_n)$ or order n has the following requirements.

6.3.2.1 Quantum memory

Each virtual edge of the routing graph indicates a Bell pair, that requires 2 qubits to sustain. If there are N edges in a routing graph, then the network has to deploy at least $2N$ qubits to sustain it.

If we define a function $\mathbf{count}(\cdot)$ which counts the number of edges in a routing graph G_n, then from the recursive construction (Equations (6.21) and (6.21)) of the routing graph we get,

$$\mathbf{count}(G_1) = 1 \tag{6.32}$$

$$\mathbf{count}(G_n) = \mathbf{count}(G_{n-1}) + 2^n \tag{6.33}$$

Here, Equation (6.33) holds because cycle C_{2^n} has 2^n edges. This gives us $\mathbf{count}(G_n) = 2^{n+1} - 3 = O(|V_n|)$. Therefore, the total number of qubits spent on this routing graph construction is linear with the number of nodes in the network.

From the recursive construction we see that, each time each new cycle is added in (6.21) all the previous nodes' degree increases by 2. For a routing graph G_n there are n such levels and G_1 has only 1 edge. Therefore, the oldest node's (for example example node 0) degree would be, $2n - 1$. That is, each of the quantum nodes (quantum routers) requires a quantum memory of size $O(\log |V_n|)$.

6.3.2.2 Classical memory

While running the protocol, the current **head** has to check whether it is adjacent to the **tail** or has a common neighbour with it. This requires each node to remember the node ID of all the nodes that are at least 2 distance apart from it. Since each node has a degree $O(\log |V_n|)$, and the node ID's are $O(\log |V_n|)$, this requires $O(\log^3 |V_n|)$ classical memory. However, since the network structure is known to all the nodes they can compute these neighbourhood in the runtime. And efficient

methods might be designed that does not require these neighbourhood relations to be explicitly saved in the node.

6.3.2.3 Entanglement swapping operation

Since the routing graph has a diameter $O(\log |V_n|)$ (see Lemma 18) the optimal path will be of length $O(\log |V_n|)$ requiring collectively $O(\log |V_n|)$ entanglement swapping operations to be performed by the middle nodes.

6.3.2.4 Running time

The main **for** loop of the Protocol 8: RinglogRoute runs $O(n) = O(\log |V_n|)$ times. All the operations inside the loop can be implemented to run in $O(1)$. So the running time of the protocol is $O(\log |V_n|)$.

6.3.3 Running the protocol in a distributed manner on a network

We have seen how the protocol finds an optimal path from node a to b in a routing graph G_n. However, in a network all the nodes would be running as independent processes. When node a, the source, wants to create a Bell pair with a remote node b, it would run the protocol and find the next neighbouring node in the routing graph G_n that lies on the optimal path. Then it would send a classical message informing that the new node should also run RinglogRoute(a, b, n), find its position in the path, and pass on the message to the next node. While passing this message, each of the middle nodes performs the entanglement swapping operation on the two virtual links that are adjacent to it, and on the path.

This way when the message finally arrives at the destination node b, the node knows that it is entangled to a and the routing is complete. Note that the classical messages propagated in the network during the protocol contain the node IDs which are of logarithmic length in terms of the network size.

Therefore, we see that our protocol Protocol 8: RinglogRoute can perform local routing (see Chapter 5.9.2).

6.3.4 Ring networks with an arbitrary number of nodes

Until now, we have designed and analysed a routing protocol for a ring network that has 2^n nodes, where $n \in \mathbb{N}$. Now, we demonstrate how this protocol can be adapted to work on a ring network with an arbitrary number of nodes.

Let us assume that we have a ring network C_m with m routers such that, $2^{n-1} < m < 2^n$. We create a routing graph for a ring of 2^n nodes and make $2^n - m$ of the routers simulate 2 nodes each. It is done in the following steps.

1. Build routing graph G_n.
2. For $i = 0 \to 2(2^n - m) - 1$ let node $\lfloor i/2 \rfloor$ of the routing graph be simulated by the ith router in the network.
3. For $i = 2(2^n - m) \to 2^n - 1$ let node i of the routing graph be simulated by the ith router in the network.

In Figure 6.8 we see an example, where a routing graph with 16 nodes is simulated by a ring network that has 13 routers.

While running Protocol 8 RinglogRout on a network of size any m, which is not a power of 2, some of the nodes have to simulate at most 2 nodes each and have to maintain all the virtual links necessary for simulating these two nodes. After an optimal path P is computed by RinglogRout, the routers check if there exist nodes u and v on P that are simulated by the same router r. In that case, the path segment from u to v is replaced by r to compute the final path.

Since, in this mode of operation each node has to simulate at most 2 nodes, the quantum memory requirement for each node still remains O(log(m)) for an m node ring.

6.4 Discussion

We have given an efficient protocol for routing in the ring quantum network. The associated routing graph that we introduce, allows the length of the node IDs to be logarithmic in the network size. It allows the network to efficiently compute the route between any pair of nodes using only the node IDs of the pair. Usually,

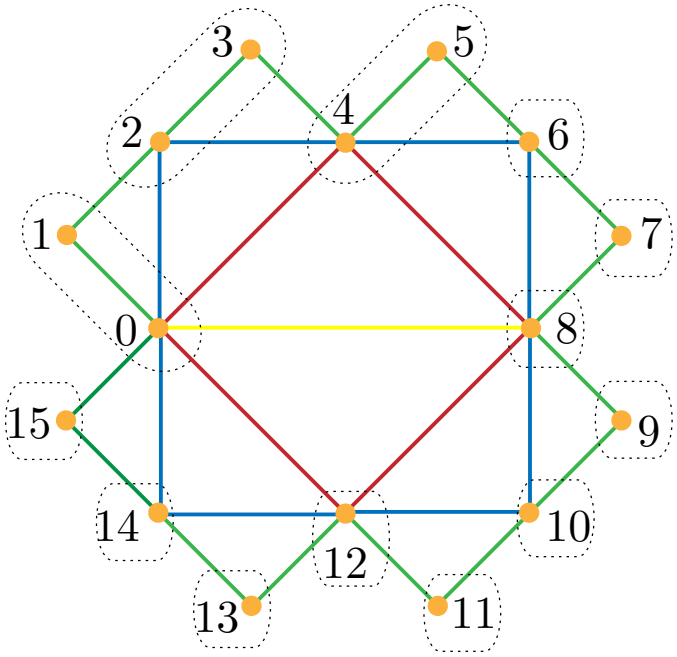


Figure 6.8: Routing protocol for a ring with an arbitrary number of nodes - Here the network has 13 nodes. Therefore, the routing graph G_4 of size 16 is simulated by this network. Each dashed rounded rectangle corresponds to a router. Each of the node pair (0,1), (2,3) and (4,5) are simulated by one router and the rest of the nodes are simulated by one router each.

computer simulations are used to quantify various properties of a routing protocol. However, the nice recursive structure of our routing graph and the associated protocol have allowed us to analytically quantify all of their important properties. This leaves hope to find routing graphs and associated protocols that have similar advantages for networks with other topologies.

7

Conclusion and outlook

The goal of this thesis has been to study various problems related to construction and operation of multiparty quantum networks. We have studied reference frame agreement problems for synchronous and asynchronous quantum networks and given protocols that solve them for $n \geq 2$ nodes. We have also studied problems related to routing quantum information in an entanglement swapping based quantum network.

Our protocols for reference frame agreement are fault tolerant because the protocol for synchronous networks can tolerate up to $t < n/3$ arbitrarily faulty nodes and the protocol for asynchronous networks can tolerate up to $t < n/4$ arbitrarily faulty nodes. In classical computing, the consensus (Byzantine agreement) and broadcast problems are studied in the cryptographic and fault tolerant settings where multiple nodes try to agree on a bit in the presence of faulty nodes. These protocols assume that the communication between any two non-faulty nodes are error-free. However, the reference frames that we agree on, cannot be transmitted using *fungible* information (i.e. only bits). The physical processes that are used to transmit the reference frames introduces inherent imperfection even in the communication between non-faulty nodes. Our reference frame agreement protocols take care of these challenges and provides a method to reach agreement on *continuous values*. Therefore, our model further generalises the Byzantine agreement problem.

We have given a very high level protocol for routing quantum information in an entanglement swapping based quantum network. Our approach has introduced

and demonstrated the advantages of a pre-established routing graph. Our protocol allows any pair of nodes on a ring network of size n to establish a Bell pair in $O(\log n)$ steps and requires only $O(\log n)$ quantum memory in each of the nodes.

7.1 Ongoing research

There are several ongoing research works, at various stages of completion, originating from our studies of quantum networks. These works are not included in this thesis. However, we briefly state them here to give a general idea of the direction towards which this thesis drives the research efforts in the field.

Our work on reference frame agreement problems has attracted considerable interest in the community and there are ongoing efforts to experimentally implement them in laboratories. The author in collaboration with the experimental physics group of Prof. Alexander Ling in Centre for Quantum Technologies, NUS is building the first prototype implementation of the synchronous reference frame agreement protocol. In this implementation, a network of 4 nodes are created which can tolerate 1 faulty node. In this prototype we use the polarisation direction of photons to transmit direction information. In the first proof-of-concept implementation each node has its own photon detectors. However, a single polarised photon source, which acts as the direction sending device of a node, is shared by all 4 nodes in turn. Currently, we are constructing a fully general prototype where each of the nodes would have their own photon sources.

The synchronous reference agreement protocol assumes that all the nodes share a common clock. That is, their local clocks are synchronised. However, synchronising clocks over a network is an important problem in itself. Our asynchronous reference frame agreement protocol, which, by definition, does not assume shared clocks, provides necessary frameworks for developing such clock synchronisation protocols. Clock synchronisation involves both frequency (tick rate) synchronisation among participating nodes and alignment of their time indices (agreeing on a 0th time index). The author is participating in an ongoing effort to develop fault tolerant clock synchronisation protocols using our asynchronous reference frame agreement protocol. Here the challenges involve identifying the communication assumptions that are subtly different from a fully

asynchronous model, and to identify and adapt a bipartite clock synchronisation protocol that can be lifted to the n party fault tolerant setting.

Various properties of the routing protocol that we introduce in this thesis can be determined analytically. However, in a realistic operating condition most routing protocols might show certain behaviours that can only be identified using extensive simulation. Since quantum primitives are often substantially different from classical primitives, the existing network simulators for classical networks are not usable in the quantum context. The author is involved in an ongoing effort to build a simulator for an entanglement swapping based quantum network that allows arbitrarily complicated protocols to be simulated for arbitrary network topologies. The simulation system is based on a discrete event simulation framework SimPy in python. The project focuses on performance of the routing protocols. Therefore, the actual quantum operations and evolution are not numerically simulated. However, the implementation assumes that these operations are performed by the routers as a black-box. Only the network layers higher than the physical layer are simulated to study network congestion and other properties. This project is currently in the testing and documentation phase.

7.2 Open problems

It is known that any optimal Byzantine agreement protocol can only tolerate up to $t < n/3$ faulty nodes. The known proofs of this bound assume error-free communication between non-faulty nodes. There are efficient classical protocols both in synchronous and asynchronous settings that achieve this bound. In our reference frame agreement protocol the agreement is achieved on a continuous value where even the communication between pairs of non-faulty nodes are inherently imperfect. Therefore, the proof techniques used to find the bounds on classical protocols do not directly translate here. This leaves hope that a different bound might be found using novel techniques, possibly using entanglement. Moreover, our asynchronous protocol achieves fault tolerance in the presence of at most $t < n/4$ faulty nodes, whereas the classical asynchronous protocol achieves $t < n/3$. Since our synchronous protocol achieves the classical bound, this gives strong hope that an asynchronous reference frame agreement protocol might also be

found that achieves this. In our work related to reference frame agreement, we did not consider relativistic effects. This is equivalent to assuming that all the nodes are on the same inertial frame, or the relativistic effects are negligible for the particular application. However, in many applications, for example in some networks involving satellites, the relativistic effects might play an important role. How to model these types of multi-party agreement problems in such relativistic scenarios remains open.

The pre-shared routing graph technique for efficient quantum routing has inspired efforts to construct such protocols for other network topologies. It remains open how such routing graph techniques might be extended in a setting where nodes constantly change their relative positions resulting in a dynamic network graph. For example, with quantum nodes spread over satellites in space, or on moving vehicles on earth, the dynamic network graph is a reasonable model to be investigated. In some distributed quantum computing schemes, a sorting network is used to redistribute qubits among the network nodes [13]. For such applications, one might investigate routing graphs that allow efficient re-permutation of n qubits over an n node network.

7.3 Concluding remark

This thesis introduces some novel ideas, to tackle several basic problems related to quantum networks, which also have many non-quantum applications. The work generalises important results in distributed computing and introduces many open questions to motivate future research efforts. My sincere hope is that this thesis will provide some useful tools and a reference point for researchers interested in this field.

Appendices

Appendix A

Graph theory

Graph theory has many uses specially in the study of communication networks. Here, we briefly introduce some of the concepts that are used in this thesis. For an in-depth introduction to graph theory we refer to, for example [144].

A.1 Graph

A *graph* $G = (V, E)$ is an order pair of sets where the set V is the set of vertices and the set E is the set of edges. If the elements $e \in E$ are of the form $e = (u, v)$ where $u, v \in V$ and e is an ordered pair then the graph is called a *directed graph*. If all $e \in E$ are two element sets of the form $e = \{u, v\}$ then the graph is called an *undirected graph*.

For example in Figure A.1 (a) graph $T = (\{1, 2, 3, 4\}, \{(1, 2), (1, 3), (1, 4)\})$ is a graph with 4 vertices and 3 edges. It is a directed graph and all the edges starts at vertex 1 and points to other vertices. Here the edges are represented as ordered pairs to preserve this *directionality*. Whereas, in Figure A.1 (b) graph $H = (\{1, 2, 3, 4\}, \{\{1, 2\}, \{1, 3\}, \{4, 1\}\})$ is an undirected graph. Since the edges or H have no directionality they are represented by 2 element sets.

Two vertices $u, v \in V$ in a graph $G = (V, E)$ are *adjacent*, if there exists an edge $\{u, v\} \in E$. If $\{u, v\} \notin E$, then they are called *non-adjacent*. For example 1 and 3 are called adjacent in the graph H but 2 and 3 are non-adjacent.

Now we define two special types of graphs, namely the cycle and the complete graph.

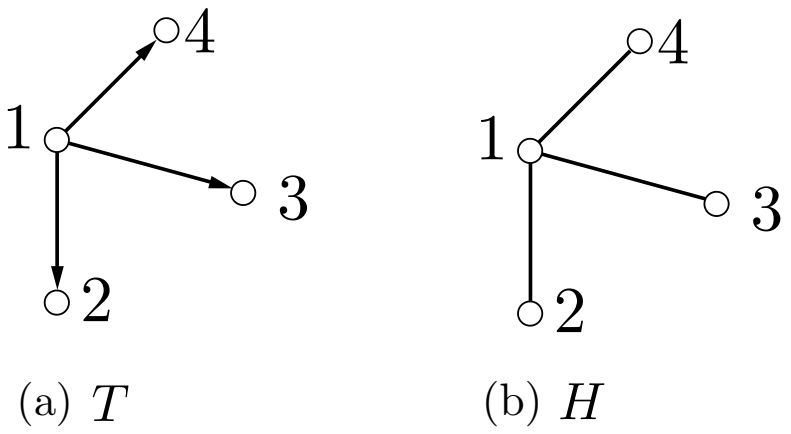


Figure A.1: Directed and undirected graphs - (a) is a representation of the directed graph $T = (\{1, 2, 3, 4\}, \{(1, 2), (1, 3), (1, 4)\})$ and (b) is the undirected graph $H = (\{1, 2, 3, 4\}, \{\{1, 2\}, \{1, 3\}, \{4, 1\}\})$.

Definition 15. A graph $C = (V, E)$ with n vertices $V = \{v_0, v_1, \ldots, v_{n-1}\}$ is called a *cycle* or a *ring* if and only if for any vertex $v_a \in V$ there exists an edge $\{v_a, v_b\} \in E$ such that, $b = a + 1 \pmod{n}$.

Definition 16. A *complete graph* is a simple graph $G = (V, E)$ where for each pair of vertices $u, v \in V$, where $u \neq v$ and there exists an edge $\{u, v\} \in E$.

A.2 Path

A *path* of length n in a graph $G = (V, E)$ is a sequence of $n + 1$ vertices $P_{ab} = (v_1, v_2, \ldots, v_{n+1})$ such that $v_1 = a$ and $v_{n+1} = b$ and for all $i < n$, $\{v_i, v_{i+1}\} \in E$. We say that P_{ab} is a path of length n, that connects nodes a and b. For example, in Figure A.1 $(2, 1, 4)$ is a path of length 2 in H.

A *simple path* of length n in a graph $G = (V, E)$ is a path $P = (v_1, v_2, \ldots, v_{n+1})$ such that no vertices are repeated in the sequence. That is in P if $i \neq j$ then $v_i \neq v_j$.

A.3 Distance

In a graph $G = (V, E)$, the distance between two vertices $u, v \in V$ is the length of the shortest simple path connecting u and v. It is represented as a function $d : V \times V \mapsto \mathbb{N}$. To avoid confusion we often use a subscript to d to indicate the graph concerned. For example in Figure A.1 (a) graph H, $d(2, 3) = 2$. This can alternatively be written as $d_H(2, 3) = 2$.

A.4 Optimal path

A path P of length n in a graph $G = (V, E)$ connecting two nodes a and b is an *optimal path* if $d(a, b) = n$. Note that for a given pair of nodes, there might be multiple optimal paths that connects them.

A.5 Diameter

The diameter of a graph $G = (V, E)$ is the length of the longest optimal path in it. Formally, diameter $D(G) = \max_{u,v \in V} d_G(u, v)$. For example, the graph H in Figure A.1 has a diameter $D(H) = 2$.

A.6 Graph isomorphism

A graph *isomorphism* f from a graph $G = (V, E)$ to a graph $G' = (V', E')$ is a bijection $f : V \mapsto V'$ from the vertex set of G to the vertex set of G' such that $\{u, v\} \in E$ if and only if $\{f(u), f(v)\} \in E'$. If such a bijection f is found then G' is *isomorphic* to G.

A.7 Subgraph

A *subgraph* $G' = (V', E')$ of a graph $G = (V, E)$ is a graph such that $V' \subseteq V$ and $E' \subseteq E$.

A.8 Induced subgraph

A induced subgraph (sometimes called an 'vertex induced subgraph') is a subset of the vertices of a graph G together with any edges whose endpoints are both in this subset. Formally, an *induced subgraph* or a *vertex induced subgraph* of a graph $G = (V, E)$ is a graph $G' = (V', E')$ where vertex set $V' \subseteq V$ and for $u, v \in V$, the edge $\{u, v\} \in E'$ if and only if $\{u, v\} \in E$.

Here, the graph G' is induced by the vertex set V' from the graph G.

A.9 Graph union

Graph $G = (V_G, E_G)$ is the *union* of two graphs $A = (V_A, E_A)$ and $A = (V_B, E_B)$ if $V_G = V_A \cup V_b$ and $E_G = E_A \cup E_B$.

Bibliography

[1] H. J Kimble. The quantum internet. *Nature*, 453:1023–1030, June 2008. doi: doi:10.1038/nature07127. 1

[2] Nicolas Gisin and Rob Thew. Quantum communication. *Nature Photonics*, 1(3):165–171, 2007.

[3] Stephan Ritter, Christian Nolleke, Carolin Hahn, Andreas Reiserer, Andreas Neuzner, Manuel Uphoff, Martin Mucke, Eden Figueroa, Joerg Bochmann, and Gerhard Rempe. An elementary quantum network of single atoms in optical cavities. *Nature*, 484:195–200, April 2012. doi: doi:10.1038/nature11023.

[4] Seth Lloyd, Jeffrey H Shapiro, Franco NC Wong, Prem Kumar, Selim M Shahriar, and Horace P Yuen. Infrastructure for the quantum internet. *ACM SIGCOMM Computer Communication Review*, 34(5):9–20, 2004.

[5] Chip Elliott. Building the quantum network. *New J. Phys.*, 4(1):46, 2002. 1

[6] Lov K Grover. Quantum telecomputation. *arXiv preprint quant-ph/9704012*, 1997. 1

[7] JI Cirac, AK Ekert, SF Huelga, and C Macchiavello. Distributed quantum computation over noisy channels. *Physical Review A*, 59(6):4249, 1999.

[8] Harry Buhrman and Hein Röhrig. Distributed quantum computing. In *Mathematical Foundations of Computer Science 2003*, pages 1–20. Springer, 2003. 89

[9] Anocha Yimsiriwattana and Samuel J Lomonaco Jr. Distributed quantum computing: A distributed shor algorithm. In *Defense and Security*, pages 360–372. International Society for Optics and Photonics, 2004.

[10] Yuan Liang Lim, Almut Beige, and Leong Chuan Kwek. Repeat-until-success linear optics distributed quantum computing. *Physical review letters*, 95(3): 030505, 2005. 89

[11] Rodney Van Meter, WJ Munro, Kae Nemoto, and Kohei M Itoh. Arithmetic on a distributed-memory quantum multicomputer. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 3(4):2, 2008.

[12] Y. Li and S. C. Benjamin. High threshold distributed quantum computing with three-qubit nodes. *New Journal of Physics*, 14(9):093008, September 2012. 89

[13] Robert Beals, Stephen Brierley, Oliver Gray, Aram W. Harrow, Samuel Kutin, Noah Linden, Dan Shepherd, and Mark Stather. Efficient distributed quantum computing. *Proc. R. Soc. A*, 469, May 2013. doi: doi:10.1098/rspa.2012.0686. 1, 89, 134

[14] Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. Universal blind quantum computation. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 517–526. IEEE, 2009. 1, 2

[15] Vedran Dunjko, Elham Kashefi, and Anthony Leverrier. Blind quantum computing with weak coherent pulses. *Phys. Rev. Lett.*, 108:200502, May 2012.

[16] Stefanie Barz, Elham Kashefi, Anne Broadbent, Joseph F. Fitzsimons, Anton Zeilinger, and Philip Walther. Demonstration of blind quantum computing. *Science*, 335(6066):303–308, 2012. doi: 10.1126/science.1214707. 1

[17] Charles H. Bennett. Quantum cryptography using any two nonorthogonal states. *Phys. Rev. Lett.*, 68:3121–3124, May 1992. 1

[18] Artur K. Ekert. Quantum cryptography based on bell's theorem. *Phys. Rev. Lett.*, 67:661–663, Aug 1991. 89

[19] Charles H Bennett, François Bessette, Gilles Brassard, Louis Salvail, and John Smolin. Experimental quantum cryptography. *Journal of cryptology*, 5(1):3–28, 1992.

[20] Nicolas Gisin, Grégoire Ribordy, Wolfgang Tittel, and Hugo Zbinden. Quantum cryptography. *Rev. Mod. Phys.*, 74:145–195, Mar 2002. 2

[21] Jonathan Barrett, Lucien Hardy, and Adrian Kent. No signaling and quantum key distribution. *Physical Review Letters*, 95(1):010503, 2005. 1

[22] Ronald L Rivest, Adi Shamir, and Leonard M Adleman. Cryptographic communications system and method, September 20 1983. US Patent 4,405,829. 1

[23] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in cryptology*, pages 10–18. Springer, 1985. 1

[24] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM journal on computing*, 26(5):1484–1509, 1997. 1

[25] Daniel J Bernstein, Johannes Buchmann, and Erik Dahmen. *Post-quantum cryptography*. Springer Science & Business Media, 2009. 2

[26] Craig Gentry et al. Fully homomorphic encryption using ideal lattices. In *STOC*, volume 9, pages 169–178, 2009. 2

[27] Peter W Shor and John Preskill. Simple proof of security of the bb84 quantum key distribution protocol. *Physical review letters*, 85(2):441, 2000. 2

[28] D Stucki, N Walenta, F Vannel, R T Thew, N Gisin, H Zbinden, S Gray, C R Towery, and S Ten. High rate, long-distance quantum key distribution

over 250km of ultra low loss fibres. *New Journal of Physics*, 11(7):075003, 2009. 2

[29] Yi Zhao, Bing Qi, Xiongfeng Ma, Hoi-Kwong Lo, and Li Qian. Experimental quantum key distribution with decoy states. *Phys. Rev. Lett.*, 96:070502, Feb 2006.

[30] Hiroki Takesue, Sae Woo Nam, Qiang Zhang, Robert H Hadfield, Toshimori Honjo, Kiyoshi Tamaki, and Yoshihisa Yamamoto. Quantum key distribution over a 40-db channel loss using superconducting single-photon detectors. *Nature photonics*, 1(6):343–348, 2007. 2

[31] Richard P Feynman. Simulating physics with computers. *International journal of theoretical physics*, 21(6/7):467–488, 1982. 2

[32] Seth Lloyd et al. Universal quantum simulators. *SCIENCE-NEW YORK THEN WASHINGTON-*, pages 1073–1077, 1996. 2

[33] Daniel S. Abrams and Seth Lloyd. Simulation of many-body fermi systems on a universal quantum computer. *Phys. Rev. Lett.*, 79:2586–2589, Sep 1997.

[34] S. Somaroo, C. H. Tseng, T. F. Havel, R. Laflamme, and D. G. Cory. Quantum simulations on a quantum computer. *Phys. Rev. Lett.*, 82:5381–5384, Jun 1999. 2

[35] A. Poppe, M. Peev, and O. Maurhart. Outline of the secoqc quantum-key-distribution network in vienna. *Int. J. Quantum Inf.*, 06(02):209–218, 2008. doi: 10.1142/S0219749908003529. 2

[36] D Stucki, M Legré, F Buntschu, B Clausen, N Felber, N Gisin, L Henzen, P Junod, G Litzistorf, P Monbaron, L Monat, J-B Page, D Perroud, G Ribordy, A Rochas, S Robyr, J Tavares, R Thew, P Trinkler, S Ventura, R Voirol, N Walenta, and H Zbinden. Long-term performance of the swissquantum quantum key distribution network in a field environment. *New J. Phys.*, 13(12):123001, 2011.

[37] M. Sasaki, M. Fujiwara, H. Ishizuka, W. Klaus, K. Wakui, M. Takeoka, S. Miki, T. Yamashita, Z. Wang, A. Tanaka, K. Yoshino, Y. Nambu, S. Takahashi, A. Tajima, A. Tomita, T. Domeki, T. Hasegawa, Y. Sakai, H. Kobayashi, T. Asai, K. Shimizu, T. Tokura, T. Tsurumaru, M. Matsui, T. Honjo, K. Tamaki, H. Takesue, Y. Tokura, J. F. Dynes, A. R. Dixon, A. W. Sharpe, Z. L. Yuan, A. J. Shields, S. Uchikoga, M. Legr, S. Robyr, P. Trinkler, L. Monat, J.-B. Page, G. Ribordy, A. Poppe, A. Allacher, O. Maurhart, T. Lnger, M. Peev, and A. Zeilinger. Field test of quantum key distribution in the tokyo qkd network. *Opt. Express*, 19(11):10387–10409, 2011. 2

[38] C Bonato, A Tomaello, V Da Deppo, G Naletto, and P Villoresi. Feasibility of satellite quantum key distribution. *New J. Phys.*, 11(4):045017, 2009. 2

[39] JP Bourgoin, E Meyer-Scott, Brendon L Higgins, B Helou, Chris Erven, Hannes Huebel, B Kumar, D Hudson, Ian D'Souza, Ralph Girard, et al. A comprehensive design and performance analysis of low earth orbit satellite quantum communication. *New Journal of Physics*, 15(2):023006, 2013.

[40] Cheng-Zhi Peng, Tao Yang, Xiao-Hui Bao, Jun Zhang, Xian-Min Jin, Fa-Yong Feng, Bin Yang, Jian Yang, Juan Yin, Qiang Zhang, Nan Li, Bao-Li Tian, and Jian-Wei Pan. Experimental free-space distribution of entangled photon pairs over 13 km: Towards satellite-based global quantum communication. *Phys. Rev. Lett.*, 94:150501, Apr 2005. 90

[41] Josep Maria Perdigues Armengol, Bernhard Furch, Clovis Jacinto de Matos, Olivier Minster, Luigi Cacciapuoti, Martin Pfennigbauer, Markus Aspelmeyer, Thomas Jennewein, Rupert Ursin, Tobias Schmitt-Manderbach, Guy Baister, John Rarity, Walter Leeb, Cesare Barbieri, Harald Weinfurter, and Anton Zeilinger. Quantum communications at esa: Towards a space experiment on the {ISS}. *Acta Astronaut.*, 63(14):165 – 178, 2008.

[42] Cristian Bonato, Markus Aspelmeyer, Thomas Jennewein, Claudio Pernechele, Paolo Villoresi, and Anton Zeilinger. Influence of satellite

motion on polarization qubits in a space–earth quantum communication link. *Opt. Express*, 14:10050–9, 2006.

[43] M. Aspelmeyer, T. Jennewein, M. Pfennigbauer, W.R. Leeb, and Anton Zeilinger. Long-distance quantum communication with entangled photons using satellites. *IEEE J. Sel. Topics Quantum Electron.*, 9(6):1541–1551, 2003. 2

[44] J. I. Cirac, P. Zoller, H. J. Kimble, and H. Mabuchi. Quantum state transfer and entanglement distribution among distant nodes in a quantum network. *Phys. Rev. Lett.*, 78:3221–3224, Apr 1997. 2

[45] Stephen D. Bartlett, Terry Rudolph, and Robert W. Spekkens. Reference frames, superselection rules, and quantum information. *Rev. Mod. Phys.*, 79:555–609, Apr 2007. doi: 10.1103/RevModPhys.79.555. 3, 24, 33

[46] N David Mermin. *Quantum computer science: an introduction*. Cambridge University Press, 2007. 3

[47] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010. 11, 13, 14, 22

[48] Mark M Wilde. *Quantum information theory*. Cambridge University Press, 2013. 3

[49] Hiroki Yamashita, Hiroyuki Itoh, Hirotoshi Tanaka, Atsumi Kawata, Kenji Nagai, Kazuhiro Yoshihara, and Ichiro Imaizumi. Flip-flop circuit, January 25 1994. US Patent 5,281,865. 3

[50] David Deutsch. Quantum theory, the church-turing principle and the universal quantum computer. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 400, pages 97–117. The Royal Society, 1985. 4

[51] Roger Colbeck and Renato Renner. Free randomness can be amplified. *Nat Phys*, 8:450–453, November 2012. 4

[52] Richard P Feynman. Quantum mechanical computers. *Foundations of physics*, 16(6):507–531, 1986. 5

[53] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493–R2496, Oct 1995. 6

[54] Wojciech H Zurek. Einselection and decoherence from an information theory perspective. In *Quantum Communication, Computing, and Measurement 3*, pages 115–125. Springer, 2002. 6

[55] 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. 17, 89

[56] Samuel L Braunstein, Ady Mann, and Michael Revzen. Maximal violation of bell inequalities for mixed states. *Physical Review Letters*, 68(22):3259, 1992. 18, 19, 89

[57] M. Żukowski, A. Zeilinger, M. A. Horne, and A. K. Ekert. "event-ready-detectors" bell experiment via entanglement swapping. *Phys. Rev. Lett.*, 71: 4287–4290, Dec 1993. 19

[58] Alexander M Goebel, Claudia Wagenknecht, Qiang Zhang, Yu-Ao Chen, Kai Chen, Jörg Schmiedmayer, and Jian-Wei Pan. Multistage entanglement swapping. *Physical Review Letters*, 101(8):080403, 2008. 19

[59] M Riebe, T Monz, K Kim, AS Villar, P Schindler, M Chwalla, M Hennrich, and R Blatt. Deterministic entanglement swapping with an ion-trap quantum computer. *Nature Physics*, 4(11):839–842, 2008. 19, 93

[60] Eli Biham, Bruno Huttner, and Tal Mor. Quantum cryptographic network based on quantum memories. *Physical Review A*, 54(4):2651, 1996. 19, 89

[61] Jarryd J Pla, Kuan Y Tan, Juan P Dehollain, Wee H Lim, John JL Morton, David N Jamieson, Andrew S Dzurak, and Andrea Morello. A single-atom electron spin qubit in silicon. *Nature*, 489(7417):541–545, 2012. 22, 32

[62] Nicholas A. Peters, Julio T. Barreiro, Michael E. Goggin, Tzu-Chieh Wei, and Paul G. Kwiat. Remote state preparation: Arbitrary remote control of photon polarization. *Phys. Rev. Lett.*, 94:150502, Apr 2005. 22, 32

[63] S. Massar and S. Popescu. Optimal extraction of information from finite quantum ensembles. *Phys. Rev. Lett.*, 74:1259–1263, Feb 1995. 23, 28, 36

[64] Asher Peres and Petra F. Scudo. Transmission of a cartesian frame by a quantum system. *Phys. Rev. Lett.*, 87:167901, Sep 2001. doi: 10.1103/PhysRevLett.87.167901. 24, 34

[65] Stephen D Bartlett, Terry Rudolph, and Robert W Spekkens. Classical and quantum communication without a shared reference frame. *Physical review letters*, 91(2):027901, 2003. 25

[66] Karl Kraus, Arno Böhm, John D Dollard, and WH Wootters. States, effects, and operations fundamental notions of quantum theory. In *States, Effects, and Operations Fundamental Notions of Quantum Theory*, volume 190, 1983. 25

[67] Paul D Townsend, JG Rarity, and PR Tapster. Single photon interference in 10 km long optical fibre interferometer. *Electronics Letters*, 29(7):634–635, 1993. 25

[68] Tobias Schmitt-Manderbach, Henning Weier, Martin Fürst, Rupert Ursin, Felix Tiefenbacher, Thomas Scheidl, Josep Perdigues, Zoran Sodnik, Christian Kurtsiefer, John G Rarity, et al. Experimental demonstration of free-space decoy-state quantum key distribution over 144 km. *Physical Review Letters*, 98(1):010504, 2007. 25

[69] M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *J. ACM*, 27(2):228–234, 1980. doi: 10.1145/322186.322188. 26, 48, 50, 57, 66

[70] Michael J Fischer. The consensus problem in unreliable distributed systems (a brief survey). In *Foundations of Computation Theory*, pages 127–140. Springer, 1983. 27

[71] Wei Ren, Randal W Beard, and Ella M Atkins. A survey of consensus problems in multi-agent coordination. In *American Control Conference, 2005. Proceedings of the 2005*, pages 1859–1864. IEEE, 2005. 27

[72] Richard D Schlichting and Fred B Schneider. Fail-stop processors: an approach to designing fault-tolerant computing systems. *ACM Transactions on Computer Systems (TOCS)*, 1(3):222–238, 1983. 27

[73] Leslie Lamport, Robert Shostak, and Marshall Pease. The Byzantine generals problem. *ACM T. Prog. Lang. Sys.*, 4(3):382–401, 1982. doi: 10.1145/357172.357176. 27, 35, 45

[74] Pesech Feldman and Silvio Micali. An optimal probabilistic protocol for synchronous byzantine agreement. *SIAM J. Comput.*, 26(4):873–933, August 1997. doi: 10.1137/S0097539790187084. 28, 45

[75] Michael Ben-Or, Elan Pavlov, and Vinod Vaikuntanathan. Byzantine agreement in the full-information model in o(log n) rounds. In *Proc. ACM STOC'06*, pages 179–186, 2006. 28, 45

[76] Ittai Abraham, Marcos K. Aguilera, and Dahlia Malkhi. Fast asynchronous consensus with optimal resilience. In *Proc. DISC'10*, pages 4–19, 2010. 28, 45, 59, 87

[77] Ittai Abraham, Danny Dolev, and Joseph Y. Halpern. An almost-surely terminating polynomial protocol for asynchronous byzantine agreement with optimal resilience. In *Proc. ACM PODC'08*, pages 405–414. ACM, 2008.

[78] Gabriel Bracha. An asynchronous [(n - 1)/3]-resilient consensus protocol. In *Proc. ACM PODC'84*, pages 154–162, 1984.

[79] Ran Canetti and Tal Rabin. Fast asynchronous byzantine agreement with optimal resilience. In *Proc. ACM STOC'93*, pages 42–51. ACM, 1993. ISBN 0-89791-591-7. doi: 10.1145/167088.167105. 28, 45, 59, 63, 64, 87

[80] Michael J. Fischer, Nancy A. Lynch, and Michael Merritt. Easy impossibility proofs for distributed consensus problems. In *Proc. ACM PODC'85*, pages 59–70, 1985. doi: 10.1145/323596.323602. 28, 59, 87

[81] Michael J Fischer, Nancy A Lynch, and Michael S Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382, 1985. 28

[82] I. Marcikic, H. de Riedmatten, W. Tittel, V. Scarani, H. Zbinden, and N. Gisin. Time-bin entangled qubits for quantum communication created by femtosecond pulses. *Phys. Rev. A*, 66:062308, Dec 2002. 32, 90

[83] Isaac L. Chuang. Quantum algorithm for distributed clock synchronization. *Phys. Rev. Lett.*, 85:2006–2009, Aug 2000. 32, 34, 87

[84] M. Ben-Or and R. El-Yaniv. Resilient-optimal interactive consistency in constant time. *DISTRIB COMPUT*, 16(4):249–262, December 2003. 32, 63, 66

[85] Asher Peres and Petra F. Scudo. Entangled quantum states as direction indicators. *Phys. Rev. Lett.*, 86:4160–4162, Apr 2001. doi: 10.1103/PhysRevLett.86.4160. 32

[86] N. Gisin and S. Popescu. Spin flips and quantum information for antiparallel spins. *Phys. Rev. Lett.*, 83:432–435, Jul 1999. doi: 10.1103/PhysRevLett.83.432. 34

[87] S. Massar. Collective versus local measurements on two parallel or antiparallel spins. *Phys. Rev. A*, 62:040101, Sep 2000. doi: 10.1103/PhysRevA.62.040101. 34

[88] Emili Bagan and Ramon Muñoz-Tapia. Aligning spatial frames through quantum channels. *Int. J. Quantum Inf.*, 4:5, 2006. 34

[89] Michael Skotiniotis and Gilad Gour. Alignment of reference frames and an operational interpretation for the g -asymmetry. *New J. Phys.*, 14(7): 073022, 2012. 34

[90] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum metrology. *Phys. Rev. Lett.*, 96:010401, 2006. doi: 10.1103/PhysRevLett.96.010401. 37

[91] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963. 38

[92] Mattias Fitzi and Ueli Maurer. From partial consistency to global broadcast. In *Proc. ACM STOC'00*, pages 494–503, 2000. doi: 10.1145/335305.335363. 45

[93] Michael Ben-Or and Avinatan Hassidim. Fast quantum byzantine agreement. In *Proc. ACM STOC'05*, pages 481–485. ACM, 2005. doi: 10.1145/1060590.1060662. 45

[94] Matthias Fitzi, Nicolas Gisin, and Ueli Maurer. Quantum solution to the Byzantine agreement problem. *Phys. Rev. Lett.*, 87:217901, 2001. doi: 10.1103/PhysRevLett.87.217901. 45

[95] R. L. Graham and A. C. Yao. On the improbability of reaching byzantine agreements. In *Proc. ACM STOC'89*, pages 467–478, 1989. doi: 10.1145/73007.73052. 59

[96] M. Fitzi, S. Wolf, and J. Wullschleger. On the power of imperfect broadcast. In *Proc. IEEE ISIT'06*, pages 504–505, 2006. doi: 10.1109/ISIT.2006.261766. 59

[97] Eshrat Arjomandi, Michael J Fischer, and Nancy A Lynch. Efficiency of synchronous versus asynchronous distributed systems. *Journal of the ACM (JACM)*, 30(3):449–456, 1983. 64

[98] Hagit Attiya and Marios Mavronicolas. Efficiency of semisynchronous versus asynchronous networks. *Mathematical Systems Theory*, 27(6):547–571, 1994. 64

[99] Charles H Bennett and Stephen J Wiesner. Communication via one-and two-particle operators on einstein-podolsky-rosen states. *Physical review letters*, 69(20):2881, 1992. 89

[100] Harry Buhrman, Richard Cleve, and Wim Van Dam. Quantum entanglement and communication complexity. *SIAM Journal on Computing*, 30(6):1829–1841, 2001. 89

[101] Richard Cleve, Daniel Gottesman, and Hoi-Kwong Lo. How to share a quantum secret. *Physical Review Letters*, 83(3):648, 1999. 89

[102] John F Clauser, Michael A Horne, Abner Shimony, and Richard A Holt. Proposed experiment to test local hidden-variable theories. *Physical review letters*, 23(15):880, 1969. 89

[103] Nicolas Gisin and Asher Peres. Maximal violation of bell's inequality for arbitrarily large spin. *Physics Letters A*, 162(1):15–17, 1992.

[104] B Hensen, H Bernien, AE Dréau, A Reiserer, N Kalb, MS Blok, J Ruitenberg, RFL Vermeulen, RN Schouten, C Abellán, et al. Loophole-free bell inequality violation using electron spins separated by 1.3 kilometres. *Nature*, 526(7575): 682–686, 2015. 89

[105] Sougato Bose, Vlatko Vedral, and Peter L Knight. Multiparticle generalization of entanglement swapping. *Physical Review A*, 57(2):822, 1998. 89, 102

[106] Daniel M Greenberger, Michael A Horne, and Anton Zeilinger. Going beyond bells theorem. In *Bells theorem, quantum theory and conceptions of the universe*, pages 69–72. Springer, 1989. 90

[107] Ying Li, Sean D Barrett, Thomas M Stace, and Simon C Benjamin. Long range failure-tolerant entanglement distribution. *New Journal of Physics*, 15(2):023012, 2013. 90

[108] Antonio Acín, J Ignacio Cirac, and Maciej Lewenstein. Entanglement percolation in quantum networks. *Nature Physics*, 3(4):256–259, 2007. 90

[109] Sébastien Perseguers, M Lewenstein, A Acín, and J Ignacio Cirac. Quantum random networks. *Nature Physics*, 6(7):539–543, 2010. 90

[110] Xian-Min Jin, Ji-Gang Ren, Bin Yang, Zhen-Huan Yi, Fei Zhou, Xiao-Fan Xu, Shao-Kai Wang, Dong Yang, Yuan-Feng Hu, Shuo Jiang, et al. Experimental free-space quantum teleportation. *Nature Photonics*, 4(6): 376–381, 2010. 90

[111] WT Buttler, RJ Hughes, PG Kwiat, GG Luther, GL Morgan, JE Nordholt, CG Peterson, and CM Simmons. Free-space quantum-key distribution. *Physical Review A*, 57(4):2379, 1998. 90

[112] Danna Rosenberg, Jim W Harrington, Patrick R Rice, Philip A Hiskett, Charles G Peterson, Richard J Hughes, Adriana E Lita, Sae Woo Nam, and Jane E Nordholt. Long-distance decoy-state quantum key distribution in optical fiber. *Physical review letters*, 98(1):010503, 2007. 90

[113] Richard J Hughes, George L Morgan, and C Glen Peterson. Quantum key distribution over a 48 km optical fibre network. *Journal of Modern Optics*, 47(2-3):533–547, 2000.

[114] Philip A Hiskett, D Rosenberg, CG Peterson, RJ Hughes, S Nam, AE Lita, AJ Miller, and JE Nordholt. Long-distance quantum key distribution in optical fibre. *New Journal of Physics*, 8(9):193, 2006.

[115] Damien Stucki, Nicolas Gisin, Olivier Guinnard, Grégoire Ribordy, and Hugo Zbinden. Quantum key distribution over 67 km with a plug&play system. *New Journal of Physics*, 4(1):41, 2002. 90

[116] Sacha Kocsis, Guo-Yong Xiang, Tim C Ralph, and Geoff J Pryde. Heralded noiseless amplification of a photon polarization qubit. *Nature Physics*, 9(1): 23–28, 2013. 90

[117] Ivan Marcikic, Hugues De Riedmatten, Wolfgang Tittel, Hugo Zbinden, Matthieu Legré, and Nicolas Gisin. Distribution of time-bin entangled qubits over 50 km of optical fiber. *Physical Review Letters*, 93(18):180502, 2004. 90

[118] L-M Duan, MD Lukin, J Ignacio Cirac, and Peter Zoller. Long-distance quantum communication with atomic ensembles and linear optics. *Nature*, 414(6862):413–418, 2001. 90, 93, 98

[119] William K Wootters and Wojciech H Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, 1982. 91, 103

[120] 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. 91, 93, 103

[121] Nicolas Sangouard, Christoph Simon, Hugues De Riedmatten, and Nicolas Gisin. Quantum repeaters based on atomic ensembles and linear optics. *Reviews of Modern Physics*, 83(1):33, 2011. 91, 93, 103

[122] S. Bose, P. L. Knight, M. B. Plenio, and V. Vedral. Proposal for teleportation of an atomic state via cavity decay. *Phys. Rev. Lett.*, 83:5158–5161, Dec 1999. 91

[123] C. Cabrillo, J. I. Cirac, P. García-Fernández, and P. Zoller. Creation of entangled states of distant atoms by interference. *Phys. Rev. A*, 59: 1025–1033, Feb 1999. 91

[124] A Kuzmich, WP Bowen, AD Boozer, A Boca, CW Chou, L-M Duan, and HJ Kimble. Generation of nonclassical photon pairs for scalable quantum communication with atomic ensembles. *Nature*, 423(6941):731–734, 2003. 93

[125] Caspar H van der Wal, Matthew D Eisaman, Axel André, Ronald L Walsworth, David F Phillips, Alexander S Zibrov, and Mikhail D Lukin. Atomic memory for correlated photon states. *Science*, 301(5630):196–200, 2003.

[126] Chin-Wen Chou, Julien Laurat, Hui Deng, Kyung Soo Choi, Hugues De Riedmatten, Daniel Felinto, and H Jeff Kimble. Functional quantum nodes for entanglement distribution over scalable quantum networks. *Science*, 316 (5829):1316–1320, 2007. 93, 96

[127] William J Munro, Koji Azuma, Kiyoshi Tamaki, and Kae Nemoto. Inside quantum repeaters. *Selected Topics in Quantum Electronics, IEEE Journal of*, 21(3):1–13, 2015. 93, 103

[128] Charles H Bennett, Gilles Brassard, Sandu Popescu, Benjamin Schumacher, John A Smolin, and William K Wootters. Purification of noisy entanglement and faithful teleportation via noisy channels. *Physical review letters*, 76(5): 722, 1996. 93

[129] David Deutsch, Artur Ekert, Richard Jozsa, Chiara Macchiavello, Sandu Popescu, and Anna Sanpera. Quantum privacy amplification and the security of quantum cryptography over noisy channels. *Physical review letters*, 77 (13):2818, 1996.

[130] Jian-Wei Pan, Christoph Simon, Časlav Brukner, and Anton Zeilinger. Entanglement purification for quantum communication. *Nature*, 410(6832): 1067–1070, 2001.

[131] W Dür and HJ Briegel. Entanglement purification and quantum error correction. *Reports on Progress in Physics*, 70(8):1381, 2007. 93

[132] Emanuel Knill and Raymond Laflamme. Theory of quantum error-correcting codes. *Physical Review A*, 55(2):900, 1997. 93

[133] Liang Jiang, Jacob M Taylor, Kae Nemoto, William J Munro, Rodney Van Meter, and Mikhail D Lukin. Quantum repeater with encoding. *Physical Review A*, 79(3):032325, 2009.

[134] WJ Munro, KA Harrison, AM Stephens, SJ Devitt, and Kae Nemoto. From quantum multiplexing to high-performance quantum networking. *Nature Photonics*, 4(11):792–796, 2010.

[135] Ashley M Stephens, Jingjing Huang, Kae Nemoto, and William J Munro. Hybrid-system approach to fault-tolerant quantum communication. *Physical Review A*, 87(5):052333, 2013. 93

[136] Sreraman Muralidharan, Jungsang Kim, Norbert Lütkenhaus, Mikhail D Lukin, and Liang Jiang. Ultrafast and fault-tolerant quantum communication across long distances. *Physical review letters*, 112(25):250501, 2014. 93

[137] Wolfgang Pfaff, BJ Hensen, Hannes Bernien, Suzanne B van Dam, Machiel S Blok, Tim H Taminiau, Marijn J Tiggelman, Raymond N Schouten, Matthew Markham, Daniel J Twitchen, et al. Unconditional quantum teleportation between distant solid-state quantum bits. *Science*, 345(6196):532–535, 2014. 96

[138] R Comroe and Daniel J Costello Jr. Arq schemes for data transmission in mobile radio systems. *Selected Areas in Communications, IEEE Journal on*, 2(4):472–481, 1984. 102

[139] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. *SIGCOMM Comput. Commun. Rev.*, 31(4):149–160, August 2001. ISSN 0146-4833. doi: 10.1145/964723.383071. URL `http://doi.acm.org/10.1145/964723.383071`. 118

[140] Antony Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Middleware 2001*, pages 329–350. Springer, 2001. 118

[141] David Andersen, Hari Balakrishnan, Frans Kaashoek, and Robert Morris. *Resilient overlay networks*, volume 35-5. ACM, 2001. 118

[142] Arturo Crespo and Hector Garcia-Molina. Semantic overlay networks for p2p systems. In *Agents and Peer-to-Peer Computing*, pages 1–13. Springer, 2004. 118

[143] Hao Zhang, Yonggang Wen, Haiyong Xie, and Nenghai Yu. Dht theory. In *Distributed Hash Table*, pages 5–22. Springer, 2013. 118

[144] Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001. 137

Index