Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Engineering

Simulating Scalable Quantum Networks

Lochlann Hackett

Supervisor: Prof. Daniel Kilper

April 19, 2023

A dissertation submitted in partial fulfilment
of the requirements for the degree of
MAI (Electrical Engineering)

Declaration

I hereby declare that this dissertation is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at `http://www.tcd.ie/calendar`.

I have completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at `http://tcd-ie.libguides.com/plagiarism/ready-steady-write`.

I consent / do not consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish (EU GDPR May 2018).

I agree that this thesis will not be publicly available, but will be available to TCD staff and students in the University's open access institutional repository on the Trinity domain only, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement. **Please consult with your supervisor on this last item before agreeing, and delete if you do not consent**

Signed: Lochlann H Ackett

Date: 17/04/2023 Type text here

Abstract

The combination of multi-qubit nodes into large-scale quantum networks will unlock a new era of quantum computing with applications ranging from and enhanced time-keeping [1], quantum cryptography [2, 3] and distributed quantum computing [4]. Apart from select applications such as Quantum key distribution (QKD), little is known about the requirements of quantum applications and protocols [5]. For this reason, accurate tools for the simulation and emulation of quantum networks will be vital for planning for the quantum internet. Two critical metrics that can be used to evaluate quantum networks are the fidelity of entangled links, F, and the rate at which qubits can be entangled or ebit rate, r_e. Many approaches exist to improve the fidelity of links such as the use of quantum repeaters that divide quantum links into shorter ones [3, 6, 7], long lived quantum memories [8–10], or purification methods such as filtration [11] and distillation [12–14]. Methods to increase ebit rate have also been implemented through the use of spontaneous parametric down conversion (SPDC) sources [15], wavelength multiplexed channels [3, 16], and even coexisting quantum and classical channels [17, 18].

In today's world, most quantum communications are still ad-hoc experimental channels that have been set up with a clear purpose in mind. However, they lack reconfigurability or scalability to be incorporated into a large-scale quantum internet. This thesis simulates small to large scale quantum networks using current, highly scalable but considerably lossy network components and compares the fidelity and ebit rate of such quantum networks with with the state-of-the-art experimental networks. By simulating such networks, the limitations and areas of weakness can be identified for large scale quantum networks.

This thesis has a focus of simulating solid-state spin qubits based on semiconductor defects that have been at the forefront of research for quantum memories. They are excellent candidates for large scale quantum networks due to their ability to efficiently interface with optical photons [19, 20], compatibility with integrated technologies [21, 22], wide selection of host materials [23], and scalable production [22]. The most well-known example of such a quantum memory is the nitrogen-vacancy (NV) center in diamond [8–10].

Lay Abstract

Classical computers used today are made up of millions of classical bits, charges across capacitors and transistors that are represented by a 1 or a 0. Quantum computing works in a fundamentally different way by leveraging the uncertainty of the quantum particles to create "qubits", which exist as a combination of both 1 and 0 until they are measured. These qubits allow for specialised quantum algorithms which solve certain problems in timescales that are orders of magnitude faster than classical computers [24].

Just like in classical computer networks, photons in fibre-optic cables can be used to connect quantum computers. The primary obstacle to creating a quantum internet is the exponential loss of photons in optical fibres with distance [25]. In classical networks this loss can be counteracted trough the use of repeaters, but for quantum states, the no cloning theorem dictates that a qubit cannot be copied, or repeated. This limitation poses a significant challenge since the quantum capacity of an optical fibre becomes zero if the fidelity of an entangled quantum link drops below the critical threshold of 0.5 [26].

This thesis simulates quantum networks of various sizes with various characterstics using modern scalable components, and identifies challenges and limitations to progress towards the realisation of a quantum internet.

Acknowledgements

First and foremost, I would like to Thank Daniel Kilper for the consistent support and knowledge provided by him throughout the academic year, I truly could not have done this without his help. I would also like to thank Jerry Horgan and Anuj Agrawal for helping to get me started with NetSquid.

I would also like to give a big thanks to Boulat Bash, Marco Ruffini and the rest of the MiniNet quantum team for their insight on quantum simulation. I am also grateful to Axel Dahlberg and the rest of the creators of NetSquid for making this thesis possible.

Nobody has been more important to me in the pursuit of this project than the members of my family. I would like to thank my parents for their continued support.

Contents

List of Figures

List of Tables

Nomenclature

m	Number of channels combined in distillation	
C	Number of wavelengths available in a channel	
F	Fidelity of entanglement	
F_i	Initialisation fidelity of an NV centre	
K	Number of shortest-paths considered	
L_f	Photon loss in fibre-optic cable	dB/km
L_n	Photon loss in when entering a node	dB
N	Total number of nodes in a network	
N_l	Total number of nodes on one link	
T_1	Relaxation time	s
T_2	Dephasing time	s
D	Depolarisation rate	$\%/km$
ψ^+	Bell state	
d	Total distance from source to destination	km
d_l	Distance between adjacent nodes	km
p_{ws}	Probability of rewiring in a Watts-Strogatz graph	
k_{ws}	Order of a Watts-Strogatz graph	
r_e	Entangled bit rate	$ebits/s$
t_{ROT}	Time of rotation instruction	ns
t_{CNOT}	Time of controlled NOT instruction	ns
t_{MEAS}	Time of measurement	ns
ρ	Density matrix of a mixed quantum state	
σ	Ideal density matrix of a quantum state	
ψ	Qubit to be teleported	
$\epsilon(\rho)$	Noise function on a qubit	
ADC	Asymmetric Depolarising channel	
SPDC	Spontaneous parametric down conversion	
DWM	Dense wave modulation	

1 Introduction

1.1 Background

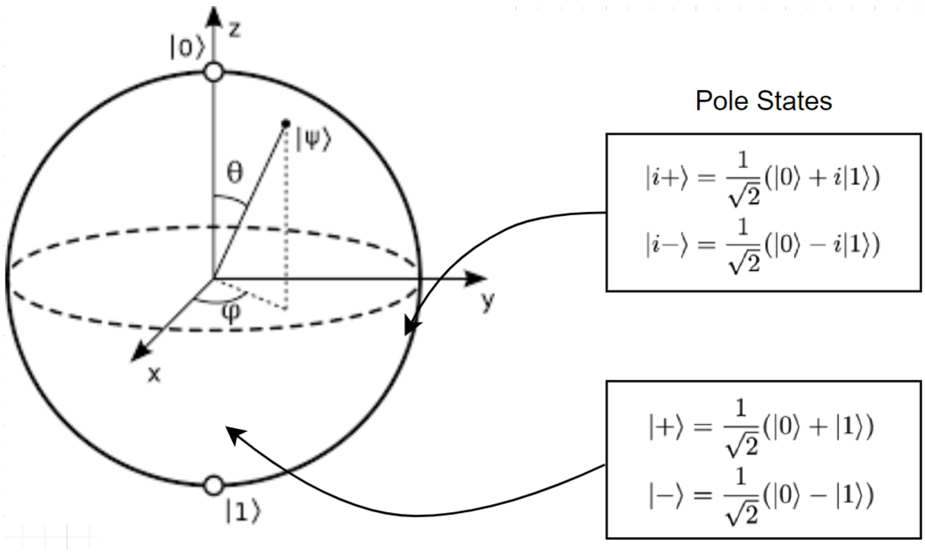
Classical computers and the internet used today are fundamentally composed of classical bits, charges across capacitors and transistors that are represented by boolean values. Since the 1980s, a fundamentally different system of computing has been developed.

Quantum computing leverages the uncertainty of the quantum properties of physical systems to create "qubits", which exist in a superposition of their boolean states until they are measured. Such physical systems include optical degrees of freedom of photons, spin states of atomic nuclei or electrons, and energy states of atoms. These qubits allow for the specialised quantum algorithms which solve some problems orders of magnitude faster than classical computers [24].

Quantum Entanglement is the physical phenomenon where two qubits are generated, interact, or share spatial proximity in such a way that their states become correlated. Entanglement can hold over arbitrary distances and taking advantage of this property of quantum systems allows us to create quantum networks. Through coordinated entanglement swaps it has been demonstrated that qubits can be "teleported" from Alice to Bob, who represent two separated quantum computers. This idea will be explained in depth in section 2.5.

Why create Quantum networks? Although qubits can coordinate their values over arbitrary distances, they cannot communicate information faster than the speed of light [30]. The result of a Bell measurement by Alice is required for Bob to realise the qubit Alice wishes to teleport. Since there is no speedup in communication the obvious question follows: Why not use quantum computers locally and send their results over a classical network? The two key use cases for quantum networks are currently Quantum Key Distribution (QKD) and Distributed Quantum Computing.

Widespread interest in the networking of quantum devices peaked when it was realised that a quantum system by nature provides knowledge of eavesdroppers in the network. If a state is measured it will change that state. This property forms the basis for QKD which is



$|0\rangle$
z
θ
$|\psi\rangle$
y
φ
x
$|1\rangle$
Pole States
$|i+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$
$|i-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$
$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
$|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

Figure 1.1: The Bloch sphere is a geometrical representation of the state of a qubit. The sphere has its centre at the origin of a three-dimensional Cartesian coordinate system. The north pole of the sphere represents the state "0", and the south pole represents the state "1". The equator of the sphere represents a qubit that is in a superposition of "0" and "1" states. Any pure state can be represented by a vector that points to a point on the surface of the Bloch sphere, where The angle between the vector and the positive z-axis represents the probability that the qubit will be measured in the state "0".

already in use commercially and institutionally [31, 32].

Distributed Quantum computing could evolve computing as we know it. Say there are two quantum computers with N number of qubits each. Through coordinated quantum teleportation, a computing system with something close to 2N qubits can be achieved. The pull towards this is that quantum computers scale poorly in terms of circuit depth (which relates to added noise due to gate operations) and qubit connectivity (ie qubits generally need to be physically close to each other to perform operations) [32]. Distributed quantum computing could allow for a complete breakthrough of the current limits of quantum computation.

Despite the continued research on quantum hardware and quantum algorithms, little is known about the requirements of quantum applications and protocols [5]. For this reason, accurate tools for the simulation and emulation of quantum networks will be vital for planning for the quantum internet. Simulators can also allow us to model theorised components, such quantum repeaters which will be discussed in section 3. Many simulators exist already for modelling quantum networks such as SQUANCH, QuNetSim, NetSquid, and Strawberry Fields, which will be discussed further in section 3.

1.2 Overall goal and Objectives

1.2.1 What is the same as the interim report

The overall goal or vision of this project is the realisation of a long-distance quantum network, capable of generating entanglement faster than it is lost, with minimal loss in fidelity. Such a network could have a large societal impact, as it is critical for the application of distributed quantum computing.

The creation of such a network will require: The use of theorised hardware that is yet to be implemented physically, and detailed planning through in-depth simulation of the physical layers, protocols, and control planes of these networks. Fundamental differences between quantum and classical communications will also necessitate the development of a new network stack, which is something that can be designed and tested using simulators and emulators [5].

In hopes of progressing towards this overall goal, the objectives specific to this project are as follows:

- **To robustly simulate complex quantum networks under conditions that approximate a physical quantum network closely.**

  A key step towards the overall goal is the development of a network simulator with the capability to simulate quantum and classical systems simultaneously. Quantum networks rely on classical networks for their control plane. The timings of quantum operations and the sharing of results of measurements must be handled through carefully timed classical messages. As these networks grow in size, adverse effects due to loss, noise and congestion are certain to appear. These phenomena can be modelled in simulators to provide insight into challenges and solutions for the physical realisation of such networks.

  Many simulators exist, but the most promising one that will be used in this thesis is NetSquid due to its discrete event engine that allows for precise tracking of time-dependent effects for accurate noise modeling.

- **To capture the general effect of decoherence on a quantum network through approximation.**

  The dimension of the Hilbert space of a quantum system grows with a factor of 2^n with the n qubits. To put this in perspective, for a system of 25 qubits, the dimension of the Hilbert space is more than 33 million. A multitude of environmental factors cause the decoherence of quantum states. This decoherence occurs in multiple forms such as dephasing, amplitude damping and depolarising noise. [33, 34] Calculating each of these gate operations over a large network is not efficient for a classical computer. Thus, a goal of this thesis is to identify ways to approximate this decoherence in such a way that the general effect of the noise is captured, while maintaining enough resources to simulate larger network topologies and more complex protocols.

- **To identify challenges associated with large scale of quantum networks.**

  Quantum networking is a relatively new field and there is a general focus in research on hardware that improves fidelity and throughput. There is a similar focus on error correction and purification techniques to improve the output fidelity of noisy quantum links. This thesis intends to take a different approach and look to the future at potential large scale quantum networks, their routing, and control.

  There is also a focus in this paper on scalable components to further this objective. The memories considered are highly manufacturable NV-centres [8–10], the channels are multi-mode fibre optic lines, and the switching is done using the already widely used Reconfigurable Optical Add Drop Multiplexers (ROADMs). Although ROADMs inject considerably more loss into an fibre channel than a fixed connection on a single-mode fibre, they have significant advantages including dynamic switching, and remote reconfiguration and installation of new wavelength channels [35]. They are also already a widespread network element in global networks. The coexistence of classical communications and quantum communications within a fibre channel has already been demonstrated for QKD [2, 18, 36], and proposals have been made for their coexistence for entanglement distribution channels [17]. Towards the objective of a scalable highly inter-connected quantum internet, the investigation of the use of ROADMs for the routing of large scale quantum networks is a step in the right direction.

1.2.2 What has changed from interim the interim report

- **Emulation aspect.**

  The true difference between emulators and simulators is highly debated amongst academics. I. McGregor defines an emulation model as "one where some functional part of the model is carried out by a part of the real system" [37]. It is impossible to truly mimic the so-called "spooky action at a distance" of sub-atomic particles with classical bits, so emulation as McGregor defines it is not truly possible for qubits. However, there are a multitude of factors that define whether a model is a simulator or an emulator, and these differences were the reason why the goal of emulating a quantum network was dropped in favour of using a simulator on its own.

  Emulators are generally designed to be operated in real time, whereas simulators are not. Considering the fact that the operations of the proposed system involve complex matrices and occur on the time-frame of nanoseconds, this would not be feasible for this project. The repeatability property of simulators also make them ideal for experimentation. Two or more model runs will always execute in exactly the same way provided that the simulation parameters are unchanged. In emulators this is generally not the case.

The use of the discrete event simulator for this thesis allowed for complex scheduling of events and in depth analysis of multiple network topologies and protocols. While emulation of quantum networks will certainly have its merits, the implementation of emulation was outside the scope of this project.

- **ZALM.**

  The modelling of the ZALM [15] architecture was an intentionally optimistic goal for this thesis. With complex hardware such as SPDC sources, and spectral mode converters, this architecture was outside the scope of a MAI thesis to model alone. However, researching and considering implementing this architecture provided great insight into what the physical aspects of a quantum network are of most importance for simulation, and what the the characteristics of a long-range quantum network should be. The concept of wavelength multiplexing was carried over into the codebase of this thesis from the ZALM paper to increase overall throughput of a quantum link.

2 State-of-the-Art/Literature review

2.1 Existing Quantum Networks

There exist many quantum networks today that utilise a diverse range of technologies and protocols depending on their use cases. QKD networks lie on the more simplistic side of these networks due to the fact that entanglement between end nodes is not required to create a secure channel.

Since the proposal of the BB84 Protocol in 1984 [38], Many QKD networks have been built using this protocol and its variants. Examples include the DARPA quantum network [39], the Tokyo quantum network, and the World's longest quantum channel from Beijing to Shanghai [32]. More recently, JPMorgan Chase's Optical Transport Lab in Columbus, Ohio built a prototype 800 Gbps QKD-secured optical channel using Dense Wavelength Division Multiplexed (DWDM) channels on the C-band, multiplexed with the QKD channel on the Oband [40]. The potential for satellites to be used for QKD networks has been proven where the channel propagates through free-space. The Micius satellite implemented such a link to the ground over a distance of 1200 km in 2016 [41].

Networks for communication between quantum computers are more complex as they require the generation of entanglement between end nodes for quantum teleportation. Significant challenges exist for this type of network. In classical networks, there is no limit to the distance of a fibre-optic link as optical amplifiers can compensate for the transmission loss. In quantum networks, amplifiers cannot be used due to the no-cloning theorem. Entanglement can hold over arbitrary distance, so there is no fundamental limit in distance for quantum teleportation. However, there is a limit to the maximum rate at which two nodes can distribute entanglement in an imperfect channel [42].

One of the key challenges is that for real-world applications such as long-distance quantum communication, quantum networks, and cryptography, heralded entanglement must be generated [43]. This means that a heralding signal must be sent to each node confirming successful entanglement generation. Another critical challenge is the interfacing between photons and quantum memories. A diverse array of hardware has been investigated for quantum memories such as neutral atoms, ion traps, rare-earth ion-doped crystals, quantum dots, and atomic gases [32]. Although quantum memories can operate at room temperature, their rate of decoherence increases with temperature and therefore most experiments with quantum memories are conducted in the range of milli-Kelvin to several Kelvin.

2.2 Diamond vacancy centres

In recent years much research has gone into solid-state quantum memories such as Nitrogen or Silicon Vacancy centres in diamond, where angled reactive-ion etching is used to scalably fabricate quantum memories starting from bulk diamond [44]. Cutting-edge technologies have produced heralded entanglement over distances up to 1.3km using solid-state Nitrogen-Vacancy (NV) centres in diamond [45]. NV centre technologies have demonstrated very long qubit lifetimes in the order of seconds [46] and the ability to generate entanglement faster than it is lost. Qubit states in NV centres can be prepared with very high initialisation Fidelity, even at room temperature [8]. Figure 2.1 depicts an NV centre's structure and ground state as a function of axial magnetic field.

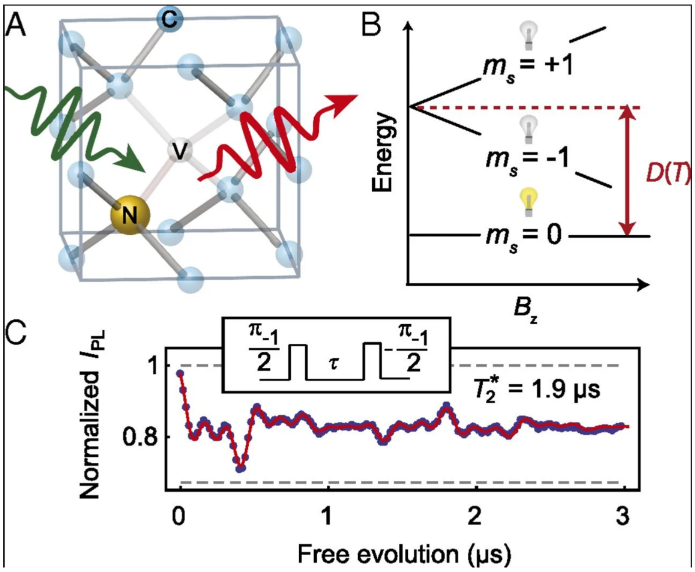


Figure 2.1: Depiction of a Nitrogen vacancy center in diamond from [27]. A) Depicts an NV center in a diamond lattice, where the green arrow represents the laser used for optical excitation and the red arrow represents the phonon-broadened fluorescence used to measure the spin state. B) shows the structure of NV center ground state as a function of axial magnetic field. The light bulbs represent the relative fluorescence difference for the $m_S = 0$ and $m_S = \pm 1$ states. Temperature differential shifts the crystal field splitting (D), whereas magnetic fields (Bz) split the $m_S = \pm 1$ sublevels. C) depicts Ramsey measurement performed on the $m_S = 0$ to $m_S = -1$ transition. The inset illustrates the pulse sequence that can be used for dynamical decoupling.

In a recently developed technique called dynamical decoupling [47], one of five controlled nuclear spins is initialized by a projective measurement. Ideally, this prepares the nuclear spin superposition state $\frac{1}{\sqrt{2}}(|\uparrow\rangle + |\downarrow\rangle)$. This process involves the application of a sequence of carefully chosen quantum operations called "pulses" to the system at regular intervals. These pulses "flip" the system back and forth between two states, so that any noise-induced changes are undone by subsequent pulses. [9, 10, 28] Reduced time between pulses desensitizes the spin to more rapidly fluctuating fields. Phase cycling of successive π pulses, has the effect of coherently rotating the Bloch vector, which helps to mitigate pulse errors [48] By adjusting the timing and strength of the pulses, it is possible to cancel out the effects of the noise and preserve the coherence of the system for a longer period of time. Dynamical decoupling allows for exceptionally long coherence decay times of up to 3600s [10].

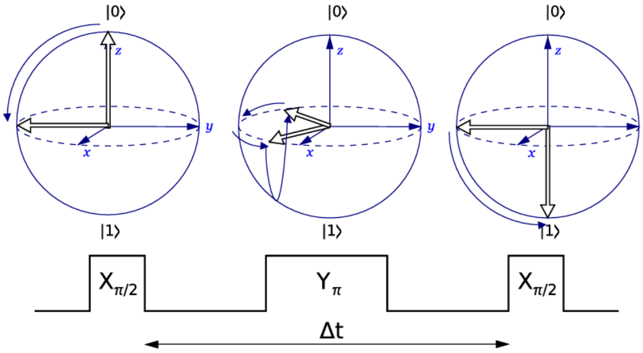


Figure 2.2: Here the free precession time the time during which the spin of the electron in the centre undergoes rotation in the absence of any external electromagnetic field. By penetrating this time with successive π pulses of a finely tuned strength, the state can be repeatedly flipped between two desirable states.

Despite NV centres exceptionally long coherence time they suffer from weak optical transitions, and spectral diffusion [49, 50]. Silicone Vacancy (siV) centres have been researched to tackle this problem. By placing SiV centers inside diamond photonic crystal cavities, a quantum-optical switch controlled by a single colour centre is formed. Raman transitions are used to realize a single-photon source with a tunable frequency and bandwidth in a diamond waveguide [22]. SiV centres demonstrate optical transitions with an inhomogeneous distribution on the order of the lifetime-broadened linewidth. This property arises from the inversion symmetry of the SiV center, which protects the optical transitions from electric field noise in the environment [22], even in nano structures [49]. SiV centres can be used to create many coupled quantum emitters per cavity as well as arrays of multiple atom-cavity nodes. They can also be used as quantum memories or quantum gates for either photonic or spin qubits. Sukachev demonstrated a coherence time in SiV centres of 13 ms, which is sufficient to maintain quantum states between quantum repeater nodes separated by 103 km. This quantum memory lifetime can be further extended by implementing dynamic decoupling techniques [9] or using nuclear spins for longer-lived memories [51].

Diamond Vacancy quantum memories have the ability to function at room temperature [51], and resemble current communications technology much closer than many other implementations of quantum memories due to their solid state. Furthermore, they could benefit from scalable, cost-effective fabrication [22, 52], allowing for easier incorporation into widespread use for quantum repeaters and global quantum networks.

2.3 Decoherence overview for NV centres

The decoherence of quantum memories that utilise spin states can be described primarily by two different processes, dephasing and relaxation, which cause random rotations on the Bloch sphere about the z axis (dephasing) and x,y axes (relaxation). [53] Together, dephasing and relaxation decohere a quantum memory at a rate of $\frac{1}{T_2} = \frac{1}{T_d} + \frac{1}{2T_1}$ [28].

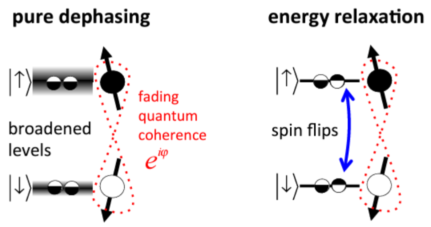


Figure 2.3: Diagram from [28] illustrating dephasing and relaxation's effect on a spin qubit. Dephasing is caused by energy state shifts in spin sublevels, whereas relaxation is the gradual decay of the excited state to the ground state due to electromagnetic noise or thermalisation with the diamond lattice. Both processes cause the phase $e^{i\phi}$ to become random after sufficient time.

2.3.1 T_2^*

T_2^*, is the inhomogeneous decoherence time and is defined as the free precession time τ after which the qubit's coherent state decays from 1 to 1/e on the interval $C(\tau) \in [0, 1]$. $C = 1$ indicates a definite relationship between the spin phases and $C = 0$ indicates a random relationship between the spin phases. The free procession time is the period between the initialization of the two-state superposition and phase readout. The decoherence rate $\frac{1}{T_2^*}$ has contributions from both dephasing and spin relaxation, so T_d is defined as the pure dephasing contribution to T_2^*. In NV centres, maintaining $T_2^* > 100\mu s$ requires a long-term ambient magnetic field stability $< 0.5\mu T$, which is around two orders of magnitude smaller than the Earth's magnetic field. [28]

2.3.2 T_2

T_2 is the homogeneous decoherence time and is defined as the time for loss of coherence from $C = 1$ to $C = 1/e$ with the usage of control pulses during the period of decoherence.

A Hahn echo serves to demonstrate this, where a free precession period is punctuated by a π pulse. This pulse mitigates the dephasing caused by slowly changing environmental fields. Through carefully timed control pulses, this process allows T2 to be extended through reduction of the T_d contribution, so that $T_2 \rightarrow T_1$. Ultimately, it has been shown that T_2 is limited to $2T_1$ [53]. Dynamical decoupling prolongs the phase coherence of a spin ensemble using the periodic application of microwave control π pulses [9, 10, 28].

2.3.3 T_1

Spin-flips between the different energy levels of a spin lead to decoherence. For the NV ground state, these flips are the transitions between the m_s = 0,-1, +1 states. Unlike dephasing, relaxation does not conserve the energy of the spin subsystem. Relaxation is largely caused by spin transitions associated with interactions of the NV's electronic orbitals with phonons in the crystal which ultimately limit the spin relaxation times between individual m_s sub-levels to about 7-10 milliseconds at $T \approx 300K$ [?]. The dominant mechanisms at this temperature are two-phonon Raman processes where a phonon is scattered, which leaves the spin in a different energy state and emits a phonon of a different energy level [28].

2.4 Quantum network simulators

As it stands most quantum communications are still ad-hoc channels, but there is constant progress toward real quantum networks. The next stage of development for quantum networks will require a coordinated effort across physicists and engineers to connect these isolated networks. To prepare for this next stage, accurate simulation and emulation of quantum networks and proposed technologies will be vital. This section, existing quantum simulators will be discussed, as well as their merits and flaws.

A large list of quantum software libraries hosted at quantiki.org[1] details approximately 100 different quantum simulation software packages. Of these, there are multiple quantum networking simulators including SQUANCH, NetSquid, QuISP, SimulaQron, Strawberry Fields, and QuNetSim. There are many more QKD simulators, however, these simulators are designed for a single use case which is not useful in the scope of this paper.

SimulaQron simulates quantum processors held by the end-nodes of the network, connected by simulated quantum communication channels, allowing the simulation of single quantum networks, as well as inter-connected quantum networks. It is designed with the goal of providing a platform for application layer development to software engineers [54]. However, its key drawbacks come with its real-time nature. This package allows the user to define the

[1] https://quantiki.org/wiki/list-qc-simulators

noise in the network themselves. However, in order to model noise effectively it is essential to be able to model time precisely, which can only be done using discrete event simulators.

QuNetSim is an open-source, Python-based, real-time quantum network simulation framework, that is straightforward to use making it accessible to people who may not fully understand the physics behind it. However, due to it being a real-time simulator it uses a large number of resources for larger simulations [55]. Its authors have suggested it works best for up to 15 hosts with a small number of network hops.

Strawberry fields is a quantum programming architecture for light-based quantum computing. It is a python based, full-stack library for the design, simulation, optimization, and quantum machine learning of continuous-variable circuits [56]. It makes use of the Continuous Variable (CV) model in which the most basic information processing unit is the infinite-dimensional bosonic mode, making it well-suited for simulating bosonic systems or settings where continuous quantum operators, such as position and momentum are present.

NetSquid is a powerful python based simulator for large-scale quantum networking and modular computing systems that are subject to physical non-idealities. It has the ability to simulate the physical properties of quantum devices such as quantum gate and memory, noise and loss of a quantum channel, and time-dependent quantum state decoherence [57]. Real-time simualors would not be suitable for this purpose as complex matrix calculations are required to be performed often in nanosecond timeframes. Upon scaling the amount of qubits to be modelled up for larger networks, this quickly becomes unfeasible for real-time simulators. NetSquid on the other hand can take its time to process events and accurately execute events as they would occur in a physical system down to the correct nanosecond. It achieves this by integrating the PyDynAA discrete-event simulation engine (see 2.7, a modular framework for modelling quantum hardware, and an asynchronous programming framework for defining quantum protocols. It has been used to develop a link layer protocol and a simplistic network stack [5], which shows promise for the purposes of this project.

2.5 Proposed solutions to limitations

To progress towards the vision of realising a quantum internet, methods for accurate long-distance quantum communications must be developed. Fibre-optic channels for classical networks typically have an attenuation length of up to 100km [58]. Unlike in classical channels, amplifiers cannot be used quantum states cannot be copied due to the no-cloning theorem. Quantum repeaters have been proposed as a solution to this problem.

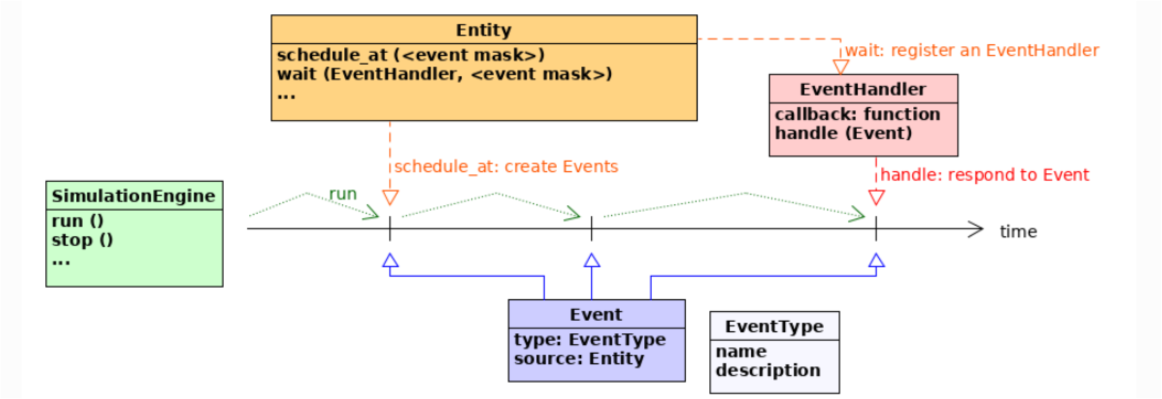


Figure 2.4: Overview of the pyDynAA discrete event simulator used by NetSquid All simulation objects are sub-classes of Entities. Entities can respond to events by the declaration of an event handler with callback to wait for events of a specific or general type. By yielding on an expression, it becomes scheduled in the simulation engine, and once that value is available after a delay, or a resource becomes available, the event will execute. From [29].

The idea is to split the channel into multiple shorter channels, zeroing the exponential loss function at each repeater. The function of a quantum repeater is dependent on three primary operations to create long-distance Bell states:

2.5.1 Entanglement Distribution

The aim is to generate entanglement between a source node and a distant target node through the distribution of entangled Bell Pairs through a chain of intermediate quantum repeater nodes. The transmission of qubits from node-node is known as quantum teleportation and the protocol is described as follows. Let Alice and Bob here represent two quantum computers. Alice has a qubit ψ, to be teleported, in the state:

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

Through coordinated quantum and classical communication, Alice can transfer this qubit to Bob. The idea is to set up entanglement between the two using a middleman, called Charlie, who represents a Bell pair source. Charlie prepares the Bell pair ϕ. The entangled pair can be written in Dirac notation as:

$$|\phi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{2.2}$$

Charlie now sends one qubit to both Alice and Bob. Alice and Bob's qubit from the Bell pair are denoted A and B respectively:

$$|\phi\rangle = \frac{1}{\sqrt{2}}(|0\rangle_A|0\rangle_B + |1\rangle|1\rangle_B) \tag{2.3}$$

The combined system can be written as:

$$|\psi\rangle \otimes |\phi\rangle = \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|100\rangle + \beta|111\rangle) \tag{2.4}$$

Alice entangles her qubit with the system by applying a CNOT gate followed by a Hadamard gate, resulting in the state:

$$\begin{aligned}(|H\rangle \otimes |I\rangle \otimes |I\rangle)(CNOT\rangle \otimes |I\rangle)(|\psi\rangle \otimes |\phi\rangle) = \frac{1}{\sqrt{2}}(&|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle) \\ &+ |10\rangle(\alpha|0\rangle - \beta|1\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle))\end{aligned}$$

When Alice measures her qubits it will always yield one of four results. On the basis of her measurement, Bob's state will be projected to:

$$|00\rangle \rightarrow (\alpha|0\rangle + \beta|1\rangle) \tag{2.5}$$

$$|01\rangle \rightarrow (\alpha|1\rangle + \beta|0\rangle) \tag{2.6}$$

$$|10\rangle \rightarrow (\alpha|0\rangle - \beta|1\rangle) \tag{2.7}$$

$$|11\rangle \rightarrow (\alpha|1\rangle - \beta|0\rangle) \tag{2.8}$$

Using the classical channel, the two bits are then sent from Alice to Bob, and then depending on these bits one of 4 operations is applied to Bob's Bell state qubit. Thus, Bob now possesses some qubit in the state psi and Alice no longer possesses it.

2.5.2 Entanglement purification

If the entangled states are stored in matter qubits of a Quantum Memory, they become prone to dephasing, meaning the relative phase between the qubits can change spontaneously. There may also be loss during state-preparation and measurement, decreasing the fidelity of the entanglement. The fidelity F, of a system with density matrix ρ, with an ideal density matrix σ, is defined as [59]:

$$F(\rho, \sigma) = (Tr(\sqrt{\sqrt{\sigma}\rho\sqrt{\sigma}}))^2 \tag{2.9}$$

The first purification scheme was proposed by Bennett et al. [60]. In this scheme, Alice and Bob already have access to 2 entangled Bell pairs. Both Alice and Bob apply CNOT gates to the corresponding qubits of each pair as control and target, respectively. The target qubits are then measured out in the computational basis $|0\rangle$, $|1\rangle$, and the results are shared over a classical channel. The unmeasured state of the control-qubits is kept only if the measurement results of the target-qubits match. The resulting state of the unmeasured qubit will be of a higher fidelity so long as the initial fidelity of both pairs was greater than 50%. If the measurement results do not match, the purification has failed and has to be repeated. This makes the purification protocol inherently probabilistic, but it is heralded.

The state of a two-level quantum system may be represented as a unit vector from the centre to the surface of a sphere known as the Bloch sphere. The unitary operation $Rx(\theta)$ rotates the Bloch Vector with respect to the x-axis by an angle θ. In matrix notation, it is expressed as:

$$R_x(\theta) = \begin{bmatrix} cos(\theta/2) & -isin(\theta/2) \\ -isin(\theta/2) & cos(\theta/2) \end{bmatrix} \tag{2.10}$$

Deutsch et. al proposed that before applying the CNOT operation, Alice should perform the rotation $R_x(\pi/2)$, and Bob should perform the inverse operation [61]. This results in a theoretical increase in Fidelity of approximately 100 times when compared to Bennett's protocol [6].

2.5.3 Entanglement swapping

Now that high-fidelity entangled links can be between nodes, a protocol for combining these links can be created. Many different protocols exist [62], but the basic idea is to repeatedly perform the quantum teleportation protocol with additional links until a unified long-distance combined state is created. As a starting point for this thesis, some of this code from NetSquid's tutorials was adapted to create a quantum repeater network, using entanglement swapping and the same protocol described in section 2.5.1. A fibre depolarisation model is used here to show that distance travelled has an effect on the fidelity of a link. However, this model is not accurate and its values used for the depolarisation rate are designed to exaggerate the effect of depolarisation in fibre. Figure 3 shows the design of a repeater in this network, its inputs, and outputs. The results shown in figure 4 demonstrate that repeaters can improve the fidelity of long-distance quantum links, but too many repeaters can also be added as noise is added at each repeater during gate operations.

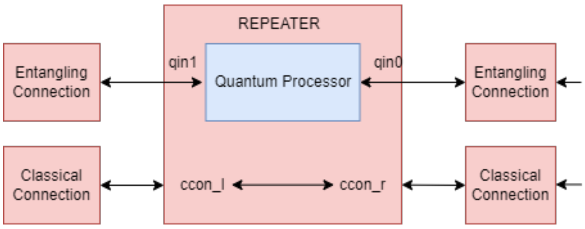


Figure 2.5: Repeater node block diagram

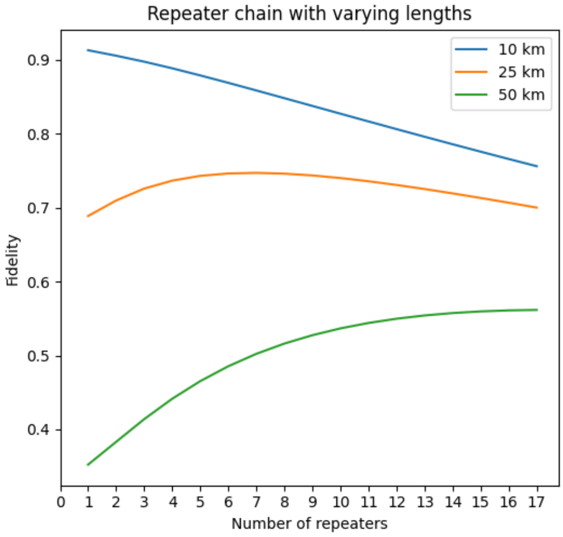


Figure 2.6: Results of repeater chain

2.6 ZALM

An important challenge to be overcome for quantum communications is the low throughput of quantum channels. A key component of that throughput comes from the entanglement generation rate from source to target. A technique to increase the entanglement generation rate is the use of Dense Wave Modulation channels. This modulation technique is used to multiplex optical signals into distinct wavelength bands, multiplying the throughput by the number of bands. This technique can also be used for light-based quantum networks using single photons. In the ZALM paper proposed by K. Chen et. al. [15], a Spontaneous Parametric Down Converter (SPDC) source under CW pumping at 800 nm is used to create a large 1THz bandwidth, a beam-splitter Bell state measurement is performed and the output beams are de-multiplexed into 80 DWDM channels, spaced 12.5 GHz apart. Although this architecture proposes propagating photons through free space, the high throughput makes up for the channel losses. This ZALM Bell pair source enables the production of greater than 100 entangled bits per second with F exceeding 99%, greatly outperforming free-running narrow band SPDC sources.

2.7 route finding

Routing algorithms are used to find the most efficient path or route between two points in a network. They help to optimize the use of resources and minimize costs, while also ensuring that data or traffic is delivered reliably and quickly.

Dijkstra's algorithm is a popular routing algorithm that is used to find the shortest path between two nodes in a graph. It works by maintaining a set of unvisited nodes and

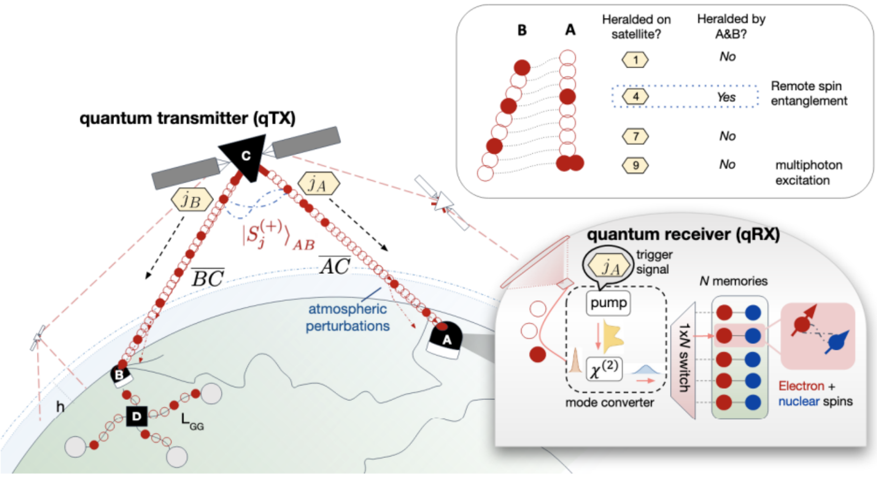


Figure 2.7: Large scale ZALM network composed of satellites as the quantum transmitters (qTX) and ground stations as the quantum receivers (qRX). From [15].

calculating the distance from the source node to each adjacent node, updating the distance as it goes. It continues this process until the destination node is reached, and then returns the shortest path.

Dijkstra's algorithm is a commonly used route finding algorithm in networks engineering that works by maintaining a set of unvisited nodes and calculating the distance from the source node to each adjacent node, updating the distance as it goes. It continues this process until the destination node is reached, and then returns the shortest path. [63] This algorithm will form the basis for route finding in this project. The pseudo code for it's implementation is as follows:

Algorithm 1 Dijkstra's algorithm

```
for each vertex v in Graph G.vertices do:
    $dist[v] \leftarrow INF$
    $prev[v] \leftarrow UNDEFINED$
    Add v to Q
end for
$dist[source] \leftarrow 0$
while $N \neq Empty$ do
    $u \leftarrow$ vertex in Q with min dist[u] Remove U from Q
    for each neighbour v of u still in Q do:
        $alt \leftarrow dist[u] + G.edges(u, v)$
        if $alt < dist[v]$ then
            $dist[v] \leftarrow alt$
            $prev[v] \leftarrow u$
        end if
    end for
end while
```

This is a powerful algorithm that can find the shortest path from a source to any node on a network, however the shortest path will not always be available for use due to wavelength blocking. For cases when alternative paths need to be considered, a different algorithm must be used. Yen's algorithm is an extension of Dijkstra's algorithm that is used to find K-shortest paths between two nodes. It works by using Dijkstra's algorithm to find the shortest path, removing that path from the graph, and repeating this process until K paths have been found. [27]

3 Technical work

The technical work done in this thesis simulates a distillable quantum network with 3 different network topologies: A two-node network, a six-node network and a Watts-Strogatz small-world network.

3.1 Architecture

3.1.1 Nodes

A node in this network is assumed to be an operational quantum computer that is interconnected to N other nodes by quantum and classical channels. A nodes is primarily composed of NV centre quantum memories. A "Quantum Source" uses a state sampler function to create quantum states with a given source fidelity and forwards them into the quantum memories when the node is the source node. For M memory positions there will be M/2 input ports and M/2 output ports. The quantum memories indexed with odd numbers are directly mapped to incoming qubits from other nodes and the quantum memories indexed with even numbers are directly mapped to the quantum sources.

These nodes make use of Reconfigurable Optical Add/Drop Multiplexers (ROADMS) for input/output. If a node recieves a photon that is bound for another node, it uses a wavelength selective switch (WSS) to redirect the wavelength carrying the photon in question to the correct optical fibre. The default case makes use of 10 channels that would be wavelength bins if this was a physical model.

3.1.2 Channels

The channels used for quantum communications are multi-mode fibre-optic cables. Attenuation loss, α in fibre-optic cables can vary widely depending on the type of fibre. Typical fibres employed nowadays attenuate the signal by 0.2dB/km, the with the lower limit limit being around 0.14dB/km [64]. Ultralow-loss reduced clad fibre was modelled in these simulations with α =0.16 dB/km [65].

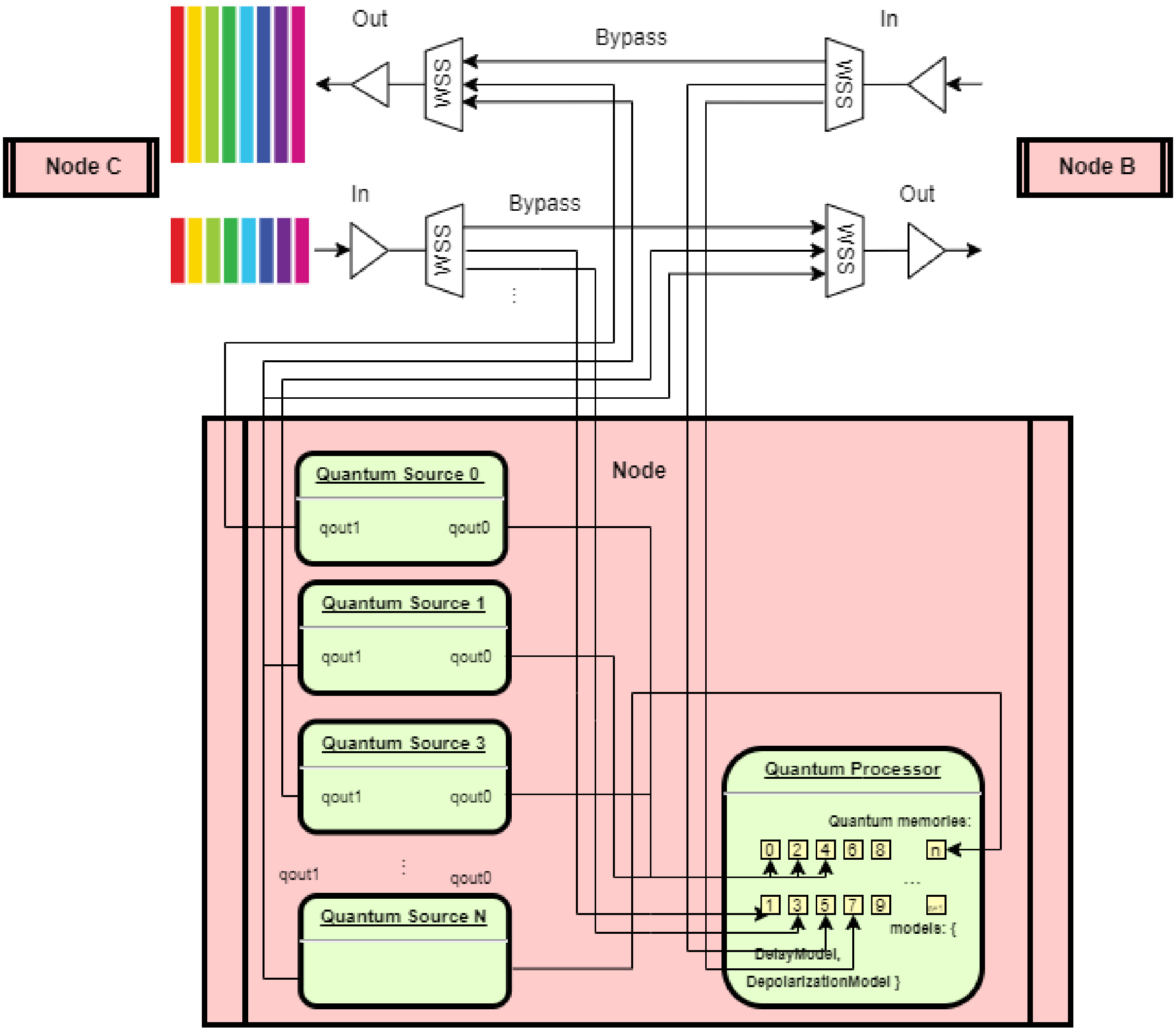


Figure 3.1: Diagram indicating the connections and sub-components in a Node.

The classical channels used in these simulations are assumed to be perfect optical-fibres that transmit messages at a speed of $200,000ms^{-1}$.

3.1.3 Quantum Memories

NV centers were modelled in this system due to their high coherence time. Specifically, they will be NV spin qubits. Researchers have demonstrated that through spin bath decoupling methods or bath engineering techniques, the Hahn-echo spin-coherence time (T2) of an NV center can reach approximately one second [66, 67]. Despite this, the spin coherence time of an NV center may experience considerable degradation in practical applications due to increased decoherence from paramagnetic spin defects.

3.1.4 Noise Models

Noise in quantum systems can be characterized by the Kraus representation, given by

$$\epsilon(\rho) = \sum_k E_k \rho E_k^\dagger \tag{3.1}$$

where E_k and $E_k^\dagger$ are the Kraus operators that satisfy $\sum_k E_k \rho E_k^\dagger = I$ [68] For this simulation we will consider the following noise models:

- **Amplitude damping noise:**

  The amplitude damping channel is an approximation that mimics the relaxation noise found in various physical systems, such as optical systems, spin-1/2 particle chains, and photon emission from an atoms. In this channel the $|0\rangle$ state as represents the ground state of a two-level atom, and the $|1\rangle$ state is the excited state of the atom. In the process of spontaneous emission, the atom tends to decay from its excited state to its ground state, even when it is in a superposition of these two states. The probability of this decay process is denoted by the parameter γ, which ranges from 0 to 1. [69]

  For a qubit, with decay probability γ, the channel's action on a the conidered state, $\varepsilon(\rho)$, can be defined using the Kraus operators:

$$E_0 = \sqrt{p}\begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{bmatrix} \qquad E_1 = \sqrt{p}\begin{bmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{bmatrix}$$

$$E_2 = \sqrt{1-p}\begin{bmatrix} \sqrt{1-\gamma} & 0 \\ 0 & 1 \end{bmatrix} \quad E_3 = \sqrt{1-p}\begin{bmatrix} 0 & 0 \\ \sqrt{\gamma} & 0 \end{bmatrix}$$

  The stationary state that is unaffected by generalized amplitude damping is

$$\rho_\infty = \begin{bmatrix} p & 0 \\ 0 & 1-p \end{bmatrix},$$

  such that $\varepsilon\left(\rho_\infty\right) = \rho_\infty$.

- Dephasing channel:

  The dephasing noise acting on a single qubit can be described by the following equation.

$$\rho \rightarrow \epsilon_{DF}(\rho) = (1-\epsilon)\rho + \epsilon Z \rho Z^\dagger \tag{3.2}$$

  Where $\epsilon \in [0,1]$ and Z is the Pauli operator.

When the action of the amplitude damping and dephasing noise is combined the result is given by: [33]

$$\rho \rightarrow \mathcal{E}_{\mathrm{D}}(\rho) = \begin{pmatrix} 1 - \rho_{11} e^{-t/T_1} & \rho_{01} e^{-t/T_2} \\ \rho_{01}^{*} e^{-t/T_2} & \rho_{11} e^{-t/T_1} \end{pmatrix} \tag{3.3}$$

- Depolarizing channel. The above noise model is incredibly memory intensive for a classical computer to simulate, so a model that can that can approximate this noise discretely is necessary for simulating a large network. The depolarization channel is such a model, where a qubit is assumed to experience from discrete Pauli X (bit-flip) errors, Z (phase flip) errors, or Y (both) [70]. When the probabilities of each type of error are different it is said to be an asymmetric depolarising channel (ADC) [33], which is defined as:

  $$\rho \rightarrow \epsilon_{ADc}(\rho) = (1 - P_{\Sigma})\rho = p_X X \rho X + p_Y Y \rho Y + p_Z Z \rho Z \tag{3.4}$$

  where $P_{\Sigma} \equiv P_X + P_Y + P_Z$

  $$\rho \rightarrow (1 - a)|\upsilon\rangle\langle\upsilon| + a/M \tag{3.5}$$

  Where $|\upsilon\rangle$ is the noiseless output state and $M = 2^N$, and N is the number of Qubits.

- **T1T2NoiseModel**

  A model exists in NetSquid that applies noise to the qubits density matrices based off the relaxation time (T_1) and dephasing time (T_2) of a quantum memory. This model approximates the combined damping and dephasing via Twirling, which maps arbitrary noise models to more symmetric ones.

  This method uses the Kraus matrices to express the noise acting on the channel in terms of Pauli matrices. Twirling over the Pauli group removes the off-diagonal terms, leading to an ADC with error probabilities: [33]

  $$p_X = p_Y = \frac{1 - e^{-t/T_1}}{4} \text{ and } p_Z = \frac{1 - e^{-t/T_2}}{2} - \frac{1 - e^{-t/T_1}}{4}.$$

  This approximation of decoherence for the NV centers is used in this thesis as it's efficiency allows for the simulation of larger networks.

A recent study using dynamical decoupling for the nuclear spins of NV centres provides exceptionally long T_1 and T_2, of 3600 s and 1.46 s, respectively [10] Nuclear spins are known to benefit from long coherence times, but are slow to manipulate and suffer from weak thermal polarisation when compared to electron spins. Therefore, electron spins are more applicable for communication qubits. Both types of spins could be used in an analogous manner to cache and main memory in classical computing. A study detailing electron spins in NV centres gives values of ($T_1 = 2.68ms$), ($T_2 = 3.3\mu s$) [71].

3.1.5 Loss Model

Signal attenuation (also known as fibre loss) is one of the most important properties of an optical fibre because it largely determines the maximum unamplified or repeaterless separation between a transmitter and a receiver, and can be attributed to a multitude of factors, such as absorption, scattering, and bending losses. [25] As Classical repeaters can not be used for quantum connections this loss will be a large limiting factor these simulations. Attenuation loss increases exponentially with distance. IF P(0) is the optical power in a fibre at the origin then the the power P(d) at distance d into the fibre is given by:

$$P(d) = P(0)10^{-\alpha_p d} \tag{3.6}$$

where α_p is the attenuation constant. α_p can vary depending on the type of fibre optic cable used. Loss in power is the ratio of lost photons to the initial number sent. When a single photon is considered, this loss can be thought of as the probability that the photon is lost in the fibre over a given distance.

The other major cause of loss in this system is the loss associated with the ROADMs. ROADMs insert different losses into a channel when adding, dropping, and switching. [35, 36] These simulations approximate an insertion of 2dB loss each time a ROADM adds, drops or switches an incoming photon. Converting from dB to a probability gives the total probability that a photon reaches its destination as:

$$P_{succ} = 1 - (10^{\frac{2N}{10}} \times 10^{-\frac{\alpha_p d}{10}}) \tag{3.7}$$

where N is the total number of nodes on a connection including the source and destination nodes. This is modelled in "CustomLossModel", which generates a random decimal d, where $0 \geq d > 1$. If $P_succ < d$ then the state of the qubit is set to none. This error model is is applied whenever a qubit is scheduled to exit a quantum channel.

3.1.6 Entangle Protocol

The entangle protocol is of the class LocalProtocol and runs independently on a source and destination node. It is triggered by a WAITING signal, or the failure of the distillation protocol.

The entanglement process begins in the source node with the generation of the bell state $|\psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$ in the QSource sub-component. The QSource component can be thought of as an abstraction of the laser pulse used to excite an NV center and generate the $|\psi^+$ state. For the Nth connection on a node, this qubit is forwarded to the quantum memory at position $2(N-1)$ and to the input port of the quantum channel. Once the quantum memory has this qubit stored in its input position, the quantum processor performs a SWAP operation taking 20ns [] on positions $2(N-1)$ and $2N$. After this a SUCCESS signal is sent to the parent protocol on the node and the QSource is triggered for a second time. The quantum processor, upon receiving the second qubit, stores it in the input memory position and sends its second SUCCESS signal.

The Qubit propagates through the channel and the photon loss model is applied. After the propagation time the destination node receives the state ψ^+ or a "None" state and places it into the quantum memory at position $2N$. The processor in the destination node now follows the same protocol as the source node, filling positions $[2N-1, 2N]$ with the inbound states, sending a SUCCESS signal after storing each state. While the entanglement may not necessarily have succeeded at this point, the quantum memory will have a "None" state in at least one position if the photon was lost in the fibre at this point. The handling of classical messages between the nodes is performed in the distillation protocol, and these "SUCCESS" messages are only used to trigger the distillation protocol.

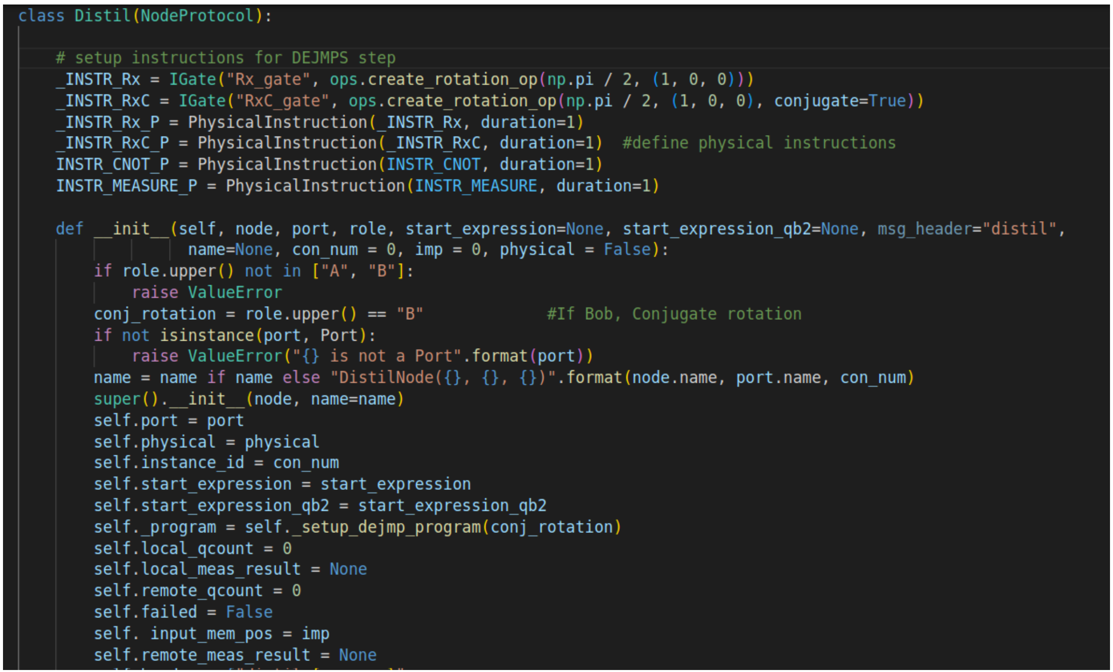
The protocol described above is the "EntangleNodes" protocol. It is only used in cases where 1 qubit is to be sent due to the way NetSquid models channels. When there is more than 1 qubit in the channel, the noise and loss models are applied to all qubits, meaning that the states of each photon propagating become mixed. A more simplistic "EntangleBase" protocol defines two seperate channels to fix this bug.

3.1.7 Distillation Protocol

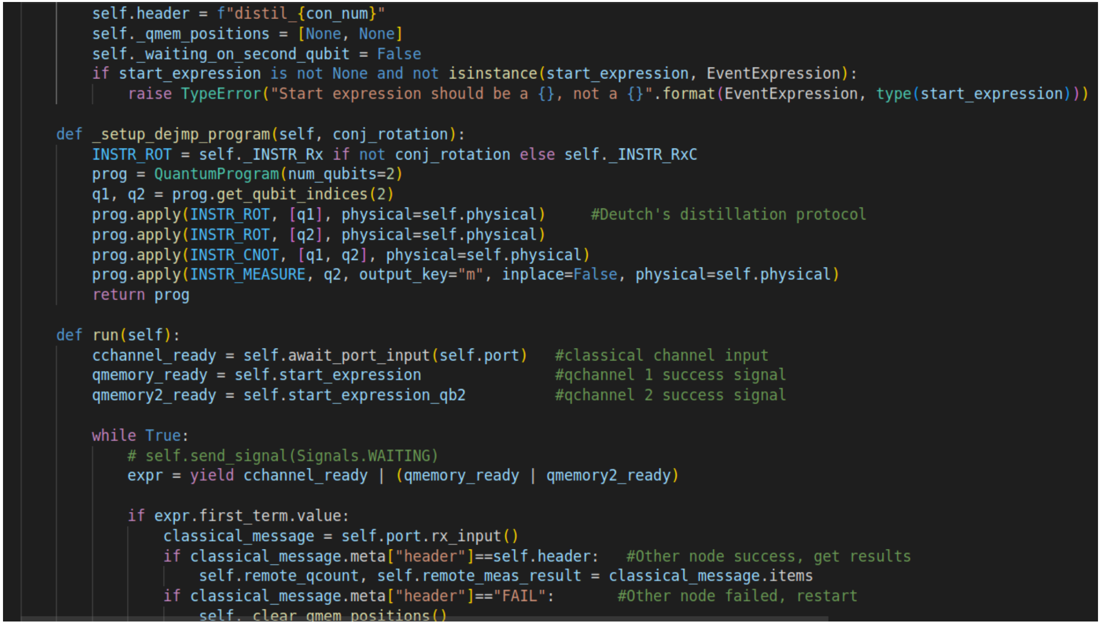
The commented code for this protocol is included below. This protocol includes most of the aspects that make NetSquid difficult to understand and give an overview of the order of operations due the scheduling of events on the discrete event engine.

Heralding

When a SUCCESS signal is received from the entangle protocol in either node, an entangled qubit is ready and so the local distillation protocol begins. This signal informs the quantum processor which memory position the ψ is located in. Before entering the *handle_new_qubit* method, the protocol checks if the quantum memory of the current position received a qubit. If it has not received a qubit a FAIL signal is sent to the parent

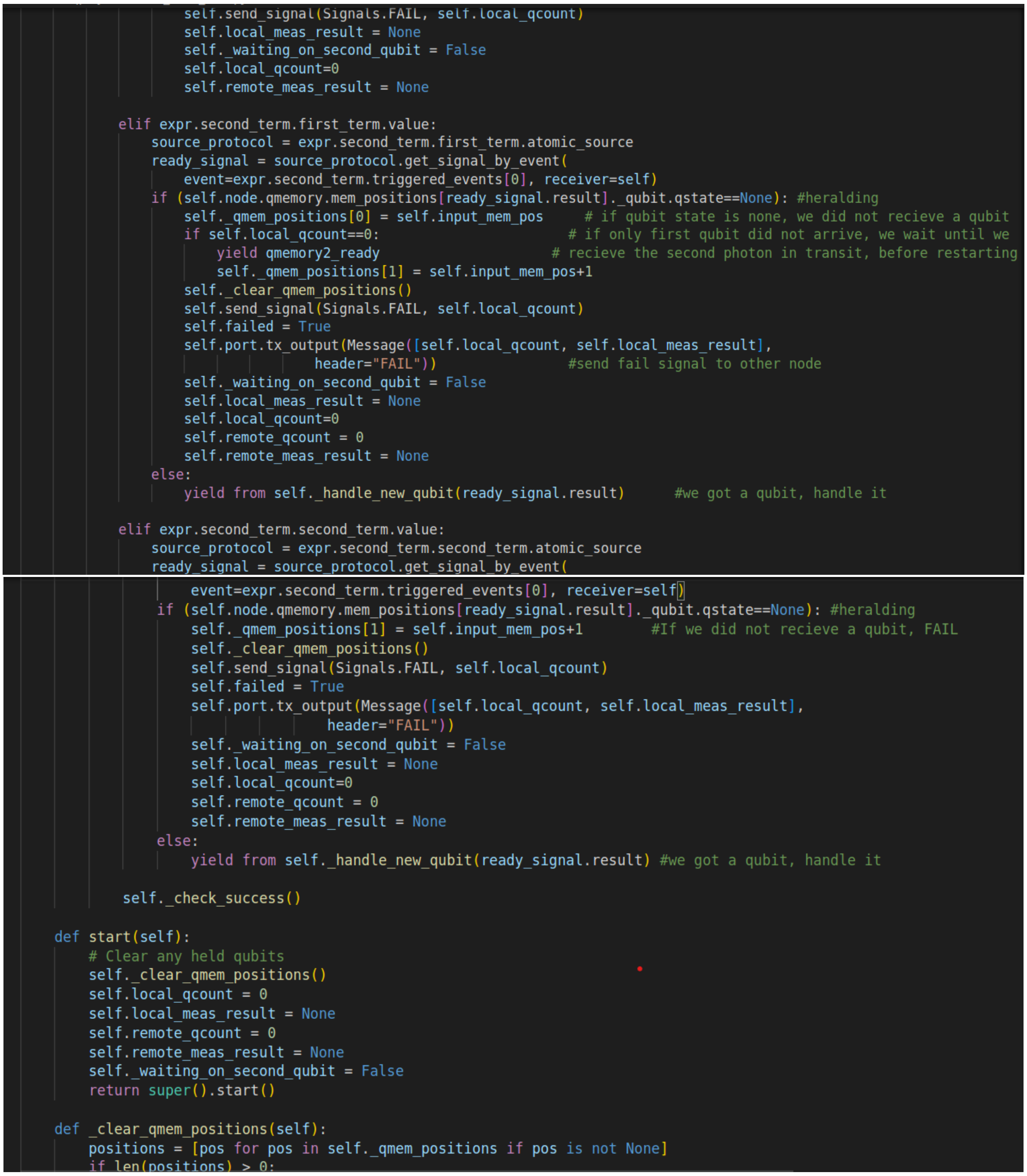


```
class Distil(NodeProtocol):

    # setup instructions for DEJMPS step
    _INSTR_Rx = IGate("Rx_gate", ops.create_rotation_op(np.pi / 2, (1, 0, 0)))
    _INSTR_RxC = IGate("RxC_gate", ops.create_rotation_op(np.pi / 2, (1, 0, 0), conjugate=True))
    _INSTR_Rx_P = PhysicalInstruction(_INSTR_Rx, duration=1)
    _INSTR_RxC_P = PhysicalInstruction(_INSTR_RxC, duration=1)  #define physical instructions
    INSTR_CNOT_P = PhysicalInstruction(INSTR_CNOT, duration=1)
    INSTR_MEASURE_P = PhysicalInstruction(INSTR_MEASURE, duration=1)

    def __init__(self, node, port, role, start_expression=None, start_expression_qb2=None, msg_header="distil",
                 name=None, con_num = 0, imp = 0, physical = False):
        if role.upper() not in ["A", "B"]:
            raise ValueError
        conj_rotation = role.upper() == "B"             #If Bob, Conjugate rotation
        if not isinstance(port, Port):
            raise ValueError("{} is not a Port".format(port))
        name = name if name else "DistilNode({}, {}, {})".format(node.name, port.name, con_num)
        super().__init__(node, name=name)
        self.port = port
        self.physical = physical
        self.instance_id = con_num
        self.start_expression = start_expression
        self.start_expression_qb2 = start_expression_qb2
        self._program = self._setup_dejmp_program(conj_rotation)
        self.local_qcount = 0
        self.local_meas_result = None
        self.remote_qcount = 0
        self.failed = False
        self. input_mem_pos = imp
        self.remote_meas_result = None
```



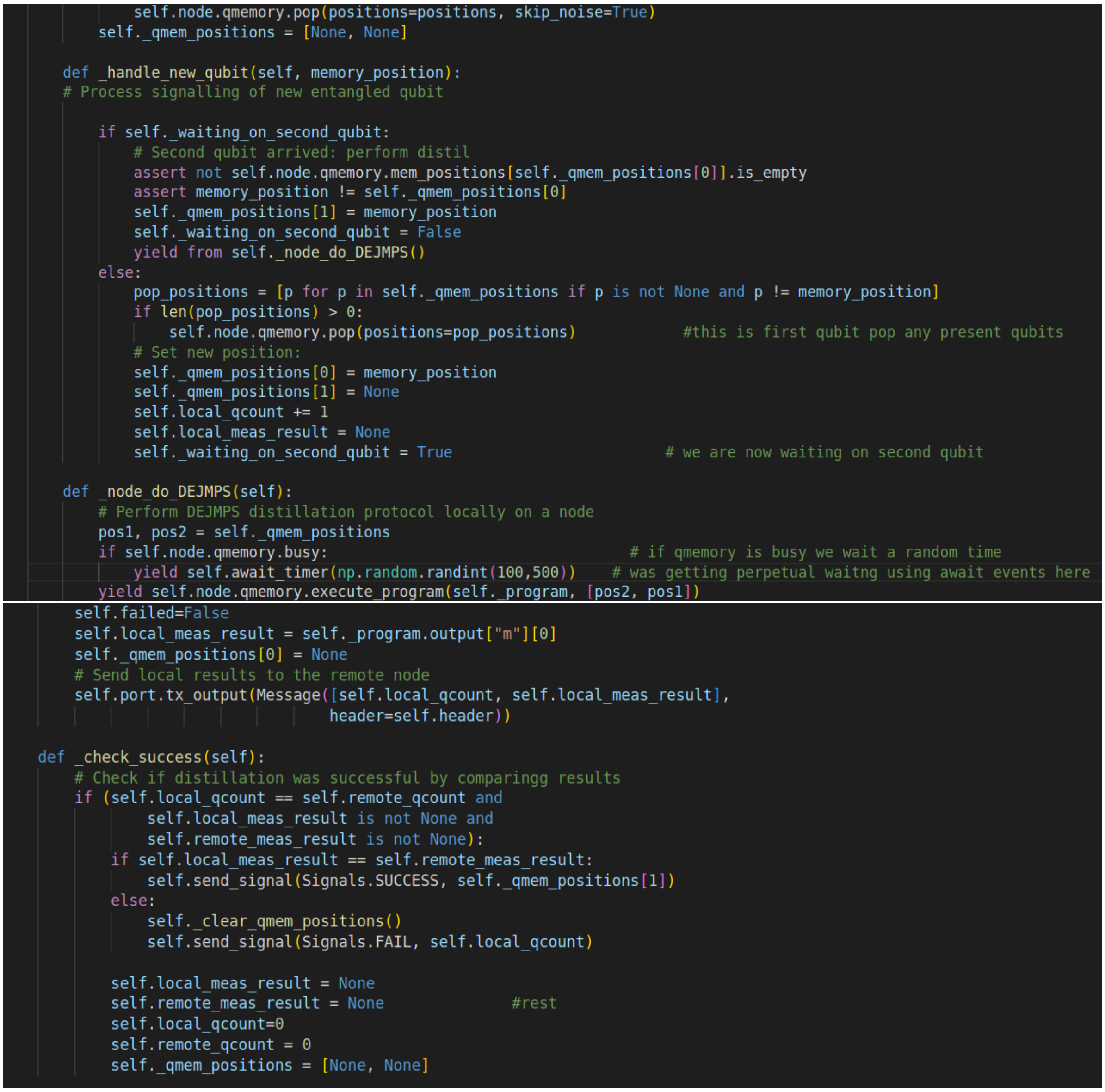
```
        self.header = f"distil_{con_num}"
        self._qmem_positions = [None, None]
        self._waiting_on_second_qubit = False
        if start_expression is not None and not isinstance(start_expression, EventExpression):
            raise TypeError("Start expression should be a {}, not a {}".format(EventExpression, type(start_expression)))

    def _setup_dejmp_program(self, conj_rotation):
        INSTR_ROT = self._INSTR_Rx if not conj_rotation else self._INSTR_RxC
        prog = QuantumProgram(num_qubits=2)
        q1, q2 = prog.get_qubit_indices(2)
        prog.apply(INSTR_ROT, [q1], physical=self.physical)     #Deutch's distillation protocol
        prog.apply(INSTR_ROT, [q2], physical=self.physical)
        prog.apply(INSTR_CNOT, [q1, q2], physical=self.physical)
        prog.apply(INSTR_MEASURE, q2, output_key="m", inplace=False, physical=self.physical)
        return prog

    def run(self):
        cchannel_ready = self.await_port_input(self.port)   #classical channel input
        qmemory_ready = self.start_expression               #qchannel 1 success signal
        qmemory2_ready = self.start_expression_qb2          #qchannel 2 success signal

        while True:
            # self.send_signal(Signals.WAITING)
            expr = yield cchannel_ready | (qmemory_ready | qmemory2_ready)

            if expr.first_term.value:
                classical_message = self.port.rx_input()
                if classical_message.meta["header"]==self.header:   #Other node success, get results
                    self.remote_qcount, self.remote_meas_result = classical_message.items
                if classical_message.meta["header"]=="FAIL":       #Other node failed, restart
                    self._clear_qmem_positions()
```

Figure 3.2: (Above) The _init _method initialises all of the classes variables that will be needed to perform distilation on two nodes. (Below) The distillation program is defined so that the processor can execute the program when called. When the run method is called by the event engine, this protocol will yeild on three expressions. If the first expression returns an input, expr.first_term will have a value. If the second expression is signalled expr.second_term.first_term will have a value. If the third wxpression is signalled expr.second_term.second_term will have a value.



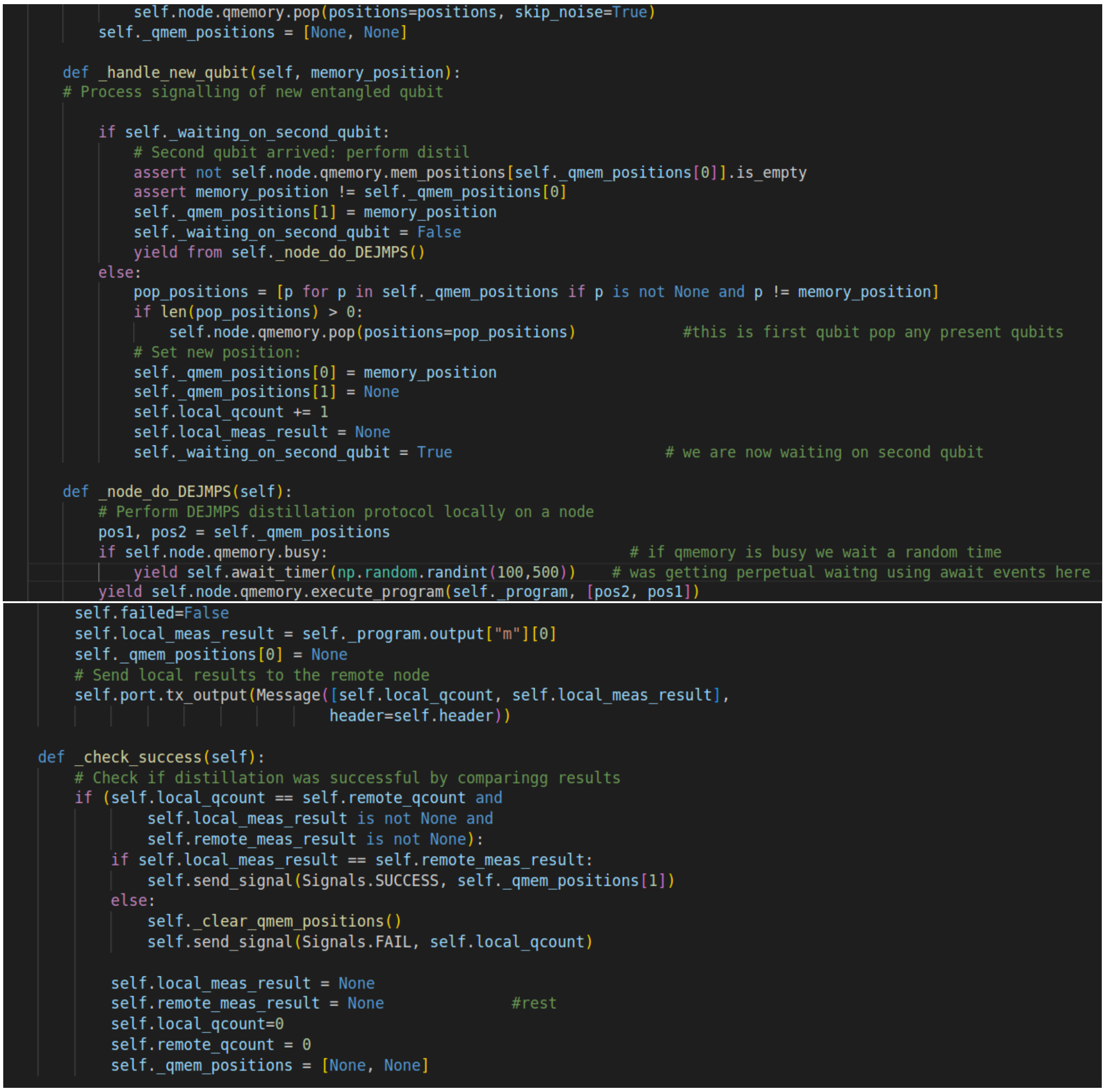
```
                    self.send_signal(Signals.FAIL, self.local_qcount)
                    self.local_meas_result = None
                    self._waiting_on_second_qubit = False
                    self.local_qcount=0
                    self.remote_meas_result = None

            elif expr.second_term.first_term.value:
                source_protocol = expr.second_term.first_term.atomic_source
                ready_signal = source_protocol.get_signal_by_event(
                    event=expr.second_term.triggered_events[0], receiver=self)
                if (self.node.qmemory.mem_positions[ready_signal.result]._qubit.qstate==None): #heralding
                    self._qmem_positions[0] = self.input_mem_pos     # if qubit state is none, we did not recieve a qubit
                    if self.local_qcount==0:                       # if only first qubit did not arrive, we wait until we
                        yield qmemory2_ready                     # recieve the second photon in transit, before restarting
                        self._qmem_positions[1] = self.input_mem_pos+1
                    self._clear_qmem_positions()
                    self.send_signal(Signals.FAIL, self.local_qcount)
                    self.failed = True
                    self.port.tx_output(Message([self.local_qcount, self.local_meas_result],
                                    header="FAIL"))                #send fail signal to other node
                    self._waiting_on_second_qubit = False
                    self.local_meas_result = None
                    self.local_qcount=0
                    self.remote_qcount = 0
                    self.remote_meas_result = None
                else:
                    yield from self._handle_new_qubit(ready_signal.result)      #we got a qubit, handle it

            elif expr.second_term.second_term.value:
                source_protocol = expr.second_term.second_term.atomic_source
                ready_signal = source_protocol.get_signal_by_event(
                    event=expr.second_term.triggered_events[0], receiver=self)
                if (self.node.qmemory.mem_positions[ready_signal.result]._qubit.qstate==None): #heralding
                    self._qmem_positions[1] = self.input_mem_pos+1        #If we did not recieve a qubit, FAIL
                    self._clear_qmem_positions()
                    self.send_signal(Signals.FAIL, self.local_qcount)
                    self.failed = True
                    self.port.tx_output(Message([self.local_qcount, self.local_meas_result],
                                    header="FAIL"))
                    self._waiting_on_second_qubit = False
                    self.local_meas_result = None
                    self.local_qcount=0
                    self.remote_qcount = 0
                    self.remote_meas_result = None
                else:
                    yield from self._handle_new_qubit(ready_signal.result) #we got a qubit, handle it

            self._check_success()

    def start(self):
        # Clear any held qubits
        self._clear_qmem_positions()
        self.local_qcount = 0
        self.local_meas_result = None
        self.remote_qcount = 0
        self.remote_meas_result = None
        self._waiting_on_second_qubit = False
        return super().start()

    def _clear_qmem_positions(self):
        positions = [pos for pos in self._qmem_positions if pos is not None]
        if len(positions) > 0:
```

Figure 3.3: Run method continued. If no qubit arrives when the protocol is waitng for its first qubit, it waits until the second qubit arrives/does not arrive before sending a fail message and restarting. If no qubit arrives when the protocol is waitng for its second qubit, a fail message is sent and the protocol restarts. If the protocol recieves a qubit, the handle qubit function is called. The start method must be called before the protocol can run, it ensures the protocol is properly initialised.



```
            self.node.qmemory.pop(positions=positions, skip_noise=True)
        self._qmem_positions = [None, None]

    def _handle_new_qubit(self, memory_position):
    # Process signalling of new entangled qubit

        if self._waiting_on_second_qubit:
            # Second qubit arrived: perform distil
            assert not self.node.qmemory.mem_positions[self._qmem_positions[0]].is_empty
            assert memory_position != self._qmem_positions[0]
            self._qmem_positions[1] = memory_position
            self._waiting_on_second_qubit = False
            yield from self._node_do_DEJMPS()
        else:
            pop_positions = [p for p in self._qmem_positions if p is not None and p != memory_position]
            if len(pop_positions) > 0:
                self.node.qmemory.pop(positions=pop_positions)        #this is first qubit pop any present qubits
            # Set new position:
            self._qmem_positions[0] = memory_position
            self._qmem_positions[1] = None
            self.local_qcount += 1
            self.local_meas_result = None
            self._waiting_on_second_qubit = True                        # we are now waiting on second qubit

    def _node_do_DEJMPS(self):
        # Perform DEJMPS distillation protocol locally on a node
        pos1, pos2 = self._qmem_positions
        if self.node.qmemory.busy:                                  # if qmemory is busy we wait a random time
            yield self.await_timer(np.random.randint(100,500))    # was getting perpetual waitng using await events here
        yield self.node.qmemory.execute_program(self._program, [pos2, pos1])
        self.failed=False
        self.local_meas_result = self._program.output["m"][0]
        self._qmem_positions[0] = None
        # Send local results to the remote node
        self.port.tx_output(Message([self.local_qcount, self.local_meas_result],
                                    header=self.header))

    def _check_success(self):
        # Check if distillation was successful by comparingg results
        if (self.local_qcount == self.remote_qcount and
                self.local_meas_result is not None and
                self.remote_meas_result is not None):
            if self.local_meas_result == self.remote_meas_result:
                self.send_signal(Signals.SUCCESS, self._qmem_positions[1])
            else:
                self._clear_qmem_positions()
                self.send_signal(Signals.FAIL, self.local_qcount)

            self.local_meas_result = None
            self.remote_meas_result = None               #rest
            self.local_qcount=0
            self.remote_qcount = 0
            self._qmem_positions = [None, None]
```

Figure 3.4: handle_new _qubit performs the relavant action on incoming qubits. If the new qubit is the first qubit to be processed, its memory position is recorded, and the waiting_on_second_qubit flag is set to true. If the qubit to be handled is the second qubit the DEJMPS program is called immediately. _do_DEJMPS calls the DEJMPS program on the two memory positions and sends the results via the classical channel. check_success compares local and remote measument results.



```
            self.node.qmemory.pop(positions=positions, skip_noise=True)
        self._qmem_positions = [None, None]

    def _handle_new_qubit(self, memory_position):
    # Process signalling of new entangled qubit

        if self._waiting_on_second_qubit:
            # Second qubit arrived: perform distil
            assert not self.node.qmemory.mem_positions[self._qmem_positions[0]].is_empty
            assert memory_position != self._qmem_positions[0]
            self._qmem_positions[1] = memory_position
            self._waiting_on_second_qubit = False
            yield from self._node_do_DEJMPS()
        else:
            pop_positions = [p for p in self._qmem_positions if p is not None and p != memory_position]
            if len(pop_positions) > 0:
                self.node.qmemory.pop(positions=pop_positions)            #this is first qubit pop any present qubits
            # Set new position:
            self._qmem_positions[0] = memory_position
            self._qmem_positions[1] = None
            self.local_qcount += 1
            self.local_meas_result = None
            self._waiting_on_second_qubit = True                        # we are now waiting on second qubit

    def _node_do_DEJMPS(self):
        # Perform DEJMPS distillation protocol locally on a node
        pos1, pos2 = self._qmem_positions
        if self.node.qmemory.busy:                                  # if qmemory is busy we wait a random time
            yield self.await_timer(np.random.randint(100,500))    # was getting perpetual waitng using await events here
        yield self.node.qmemory.execute_program(self._program, [pos2, pos1])
        self.failed=False
        self.local_meas_result = self._program.output["m"][0]
        self._qmem_positions[0] = None
        # Send local results to the remote node
        self.port.tx_output(Message([self.local_qcount, self.local_meas_result],
                                    header=self.header))

    def _check_success(self):
        # Check if distillation was successful by comparingg results
        if (self.local_qcount == self.remote_qcount and
                self.local_meas_result is not None and
                self.remote_meas_result is not None):
            if self.local_meas_result == self.remote_meas_result:
                self.send_signal(Signals.SUCCESS, self._qmem_positions[1])
            else:
                self._clear_qmem_positions()
                self.send_signal(Signals.FAIL, self.local_qcount)

            self.local_meas_result = None
            self.remote_meas_result = None               #rest
            self.local_qcount=0
            self.remote_qcount = 0
            self._qmem_positions = [None, None]
```

Figure 3.5: handle_new _qubit performs the relavant action on incoming qubits. If the new qubit is the first qubit to be processed, its memory position is recorded, and the waiting_on_second_qubit flag is set to true. If the qubit to be handled is the second qubit the DEJMPS program is called immediately. _do_DEJMPS calls the DEJMPS program on the two memory positions and sends the results via the classical channel. check_success compares local and remote measument results.

protocol, and a classical message is sent to the source node with a header of ”*FAIL_DISTILsrcdestconnection_number*”. It is important to note that heralding does not occur like this in a physical system, but this has little effect on the outcome of the simulations performed in this paper.

If the there is indeed a qubit in the memory position signalled by the entangleNodes protocol, the *handle_new_qubit* method is called to check which of the two qubits needed for distillation is currently being considered. The "*waiting_on_second_qubit*" flag, which is initialised to false, is checked here. If it is false, it is set to true, and the protocol waits for the next qubit to arrive. When "*waiting_on_second_qubit*" is true, the *do_DEJPMS* method is called.

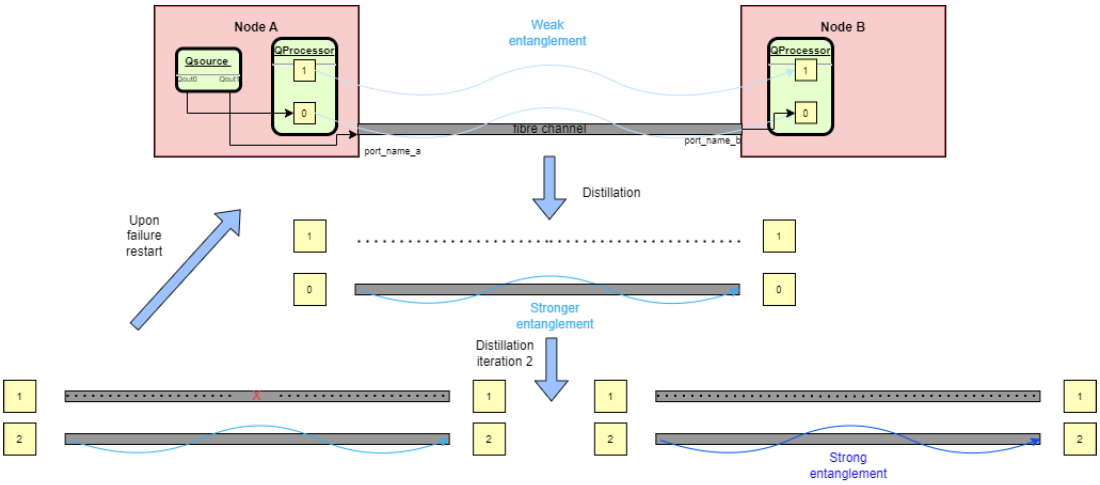


Figure 3.6: Diagram indicating the distillation process.

DEJMPS

The Distributed Entanglement Joint Measurement Protocol(DEJMPS) was proposed by Bennet et. al [61] and allows for two copies of a mixed entangled state to be converted into a smaller number of higher-quality entangled states. It is a simple protocol with the following steps:

1. The protocol must be consistent with which qubit pair should be kept as the memory qubit pair after distillation. Here, we always use position$2N$
2. The source node performs a $R(\pi/2)$ operation on both qubits, and destination node performs $R(\pi/2)$ gates on both qubits.
3. Both nodes apply a CNOT gate on their qubits. Here, they choose positions $2N$ as their control qubits and $2N-1$ as target qubits
4. Both nodes measure the target qubits.

These instructions are modelled as physical gates, meaning that their operation takes a fixed duration and applies the non linear noise model to the qubits when they are executed. The times for each operation are as follows. [33]

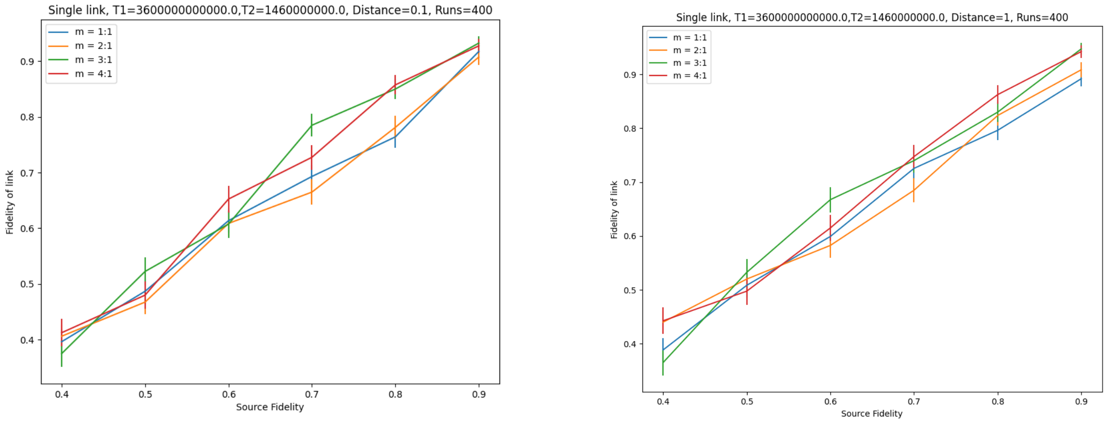
Operation	Duration (ns)
ROT	5
CNOT	20
SWAP	20
MEASURE	35

Once the measurement has been made on the target qubit, each node shares the result of that measurement via a classical channel. If the measurement results are the same, the distillation is successful, and the qubit pair is stored as state ρ_out. If the measurement results are different, the distillation process has failed and the control qubit pair will be discarded. Upon failure each node will send a FAIL signal to the parent protocol.

DistilSetup

DistilSetup is the name of the parent protocol that initialises the EntangleNodes and Distil protocols. It also sets the start expressions for each subprotocol. "Entangle_A" awaits a FAIL signal from "Purify_A" or a WAITING signal which is sent once the sub-protocols are started. "Entangle_B" does not have a start expression, and is constantly awaiting a port input from the source node until the success or failure of the entire protocol. "Purify_A" and "Purify_B" await SUCCESS signals from "Entangle_A" and "Entangle_B", respectively.

The run method in this protocol starts each protocol and yields the SUCCESS signal of both "Purify_A" and "Purify_B". Upon success a dictionary called "results" is created containing the meaningful data from the overall distillation run, such as the memory position of ρ_{out} for the source and destination node, the time taken from this distillation run, and the number of pairs generated by the source node. The results dictionary is included in SUCCESS event expression of the DistilSetup protocol.



(a) d = 0.1km

(b) d = 1km

Figure 3.7: Figure plotting Output fidelity vs input fidelity for varying distillation runs in dynamical decoupling NV-centres

A netsquid.util.DataCollector is defined before the distillation run that collects data upon the success of the DistilSetup protocol. This triggers the record_run function, which calculates the fidelity F of the two qubits, and is stores it as well as the rest of the data in a pandas DataFrame with the keys as column names.

Further distillation

For further distillation to be applied, DistilTwice and DistilThrice are defined as parent protocols and are similar to DistilSetup. DistilTwice is initialised in the same way DistilSetup, except additional sub-protocols "Entangle_A1","Entangle_B1","Purify_A1" and "Purify_B1" are initialised. "Entangle_A1" and "Entangle_B1" perform the Entanglenodes protocol for 1 additional qubit on the next available memory position. "Purify_A1" and "Purify_B1", perform a modified version of the Distil protocol on the ρ_{out} state and the newly entangled qubit from "Entangle_A1" and "Entangle_B1". This protocol runs in the same way as DistilSetup, until the SUCCESS signal from "Purify_A" and "Purify_B" is received. Upon receiving these signals, the protocol attempts to entangle the next qubit for distillation. The reason this step was not done in parallel is that entangling a single qubit takes significantly shorter than the distillation protocol. Therefore, this qubit would be sitting in the quantum memory, decohering while waiting for the first distillation process to complete.

3.2 Scaling up the system

Connecting more than just two nodes is possible in NetSquid, as an arbitrary number of i/o ports and channels can be defined. However, there is no mechanism for dynamically routing

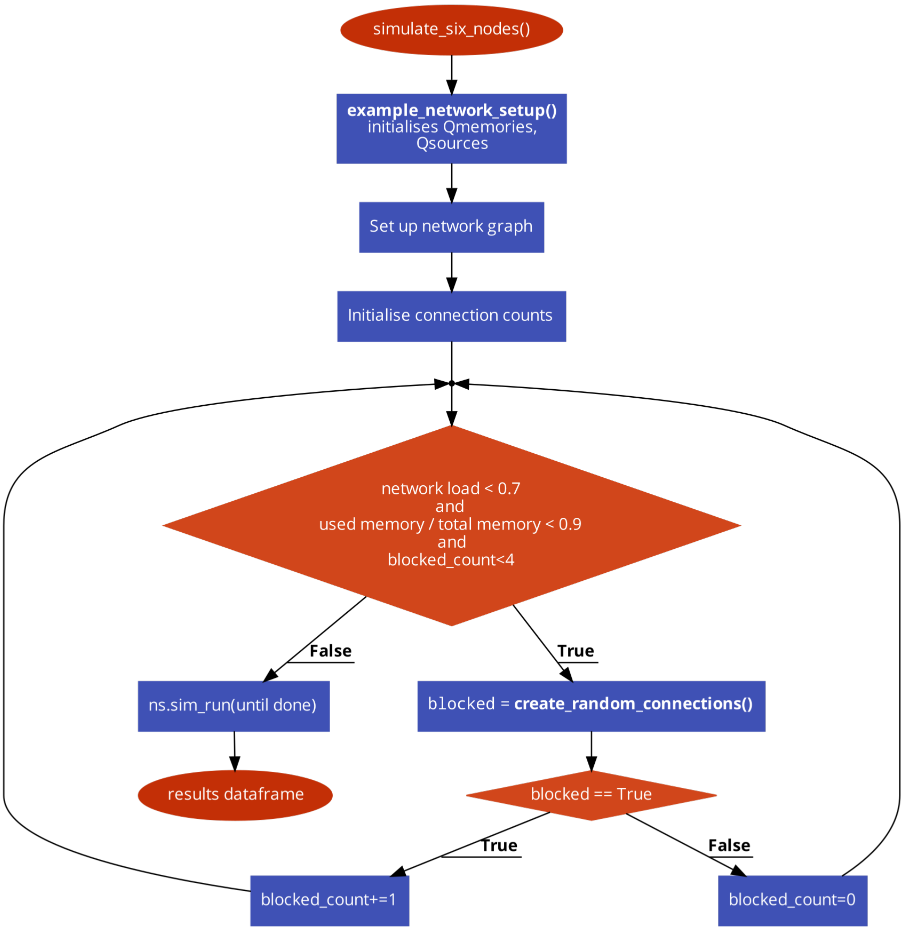


Figure 3.8: Flow chart of the simulate_six_nodes function.

connections between interconnected networks. This issue was addressed by defining a network graph for routing, a "create_random _connections" function and a "add_connection" function.

The network graph is defined as a dictionary of dictionaries, where the first set of keys are the source nodes. Within each outer dictionary is a list of keys indicating the destination node, whose value is the weight (in this case this is distance) from source to destination. Initially, the network graph was defined as a matrix but was changed to allow for the potential addition of additional data within each source/node link such another weight for node loss.

The dimensions and weights of this graph can be defined arbitrarily, however the graph considered for the medium sized network is designed to represent the topology shown in figure 3.9, where the weights are equal indicating equal distances between adjacent nodes.

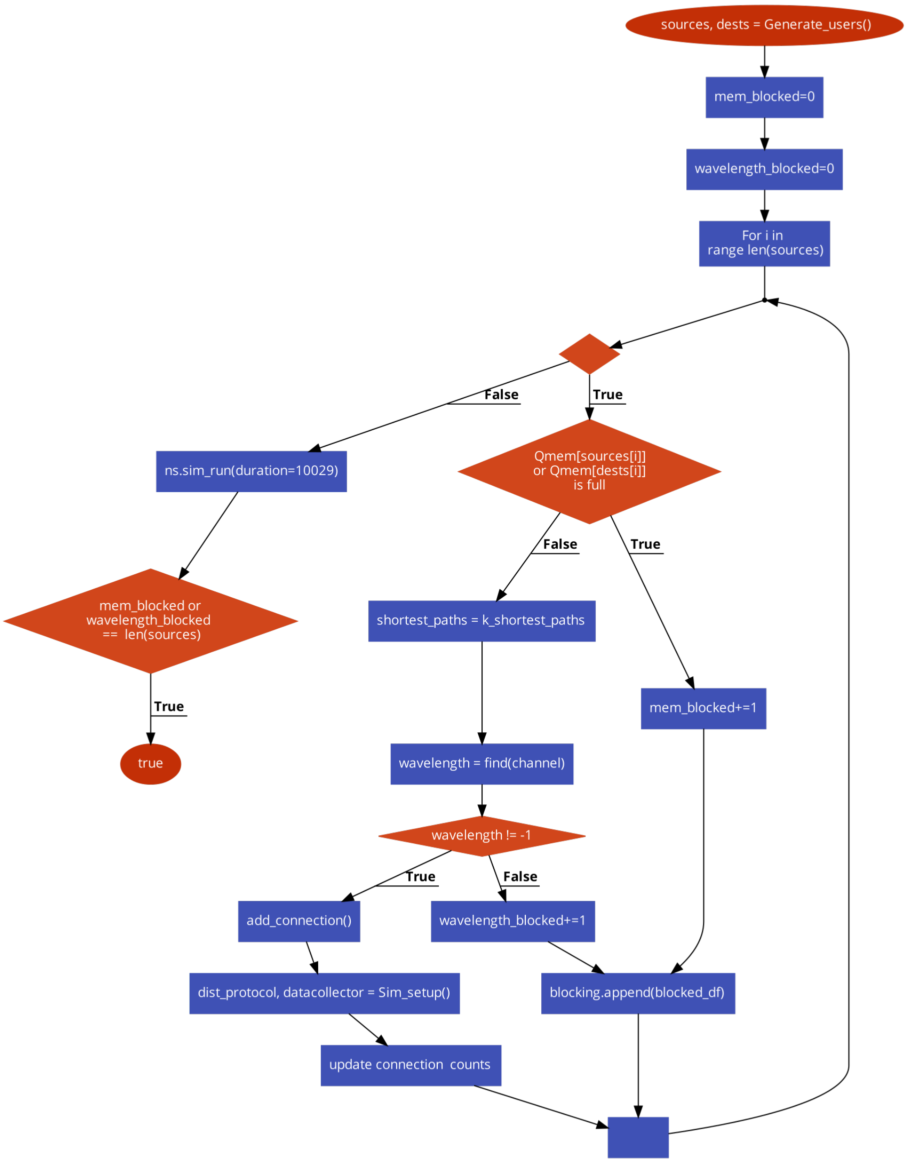


Figure 3.9: Flow chart of the create_random_connections function.

3.2.1 Route Finding

Now that the network graph has been defined, a routing algorithm can be used to route connections from A to B. The "add_connection" function is used to interface between the routing aspect and quantum simulation aspect of this codebase. Once the routing algorithm returns a path to be connected, a distance parameter is calculated from the path returned, and loss injection due to passing through the ROADMs is calculated from the length of the path. This function creates a QuantumChannel and a ClassicalChannel and adds them as connections to the Network.

This algorithm is powerful and can find the shortest path from a source to any node on a network, however the shortest path will not always be available for use in this project so for cases when we want to consider alternative paths a different algorithm must be used.

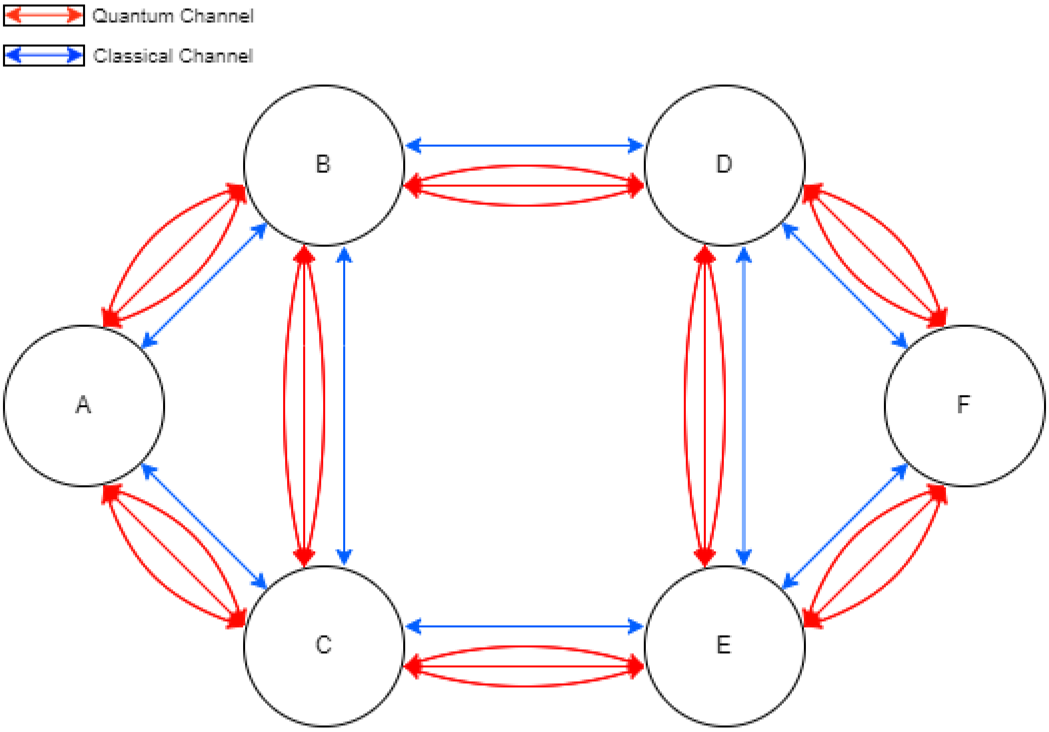


Figure 3.10: 6 node network topology.

3.2.2 Wavelength Blocking

In this model the quantum channels are distinct wavelength bands in optical-fibre, and thus are subject the same limitations associated with routing of classical photonic signals. One such limitation is wavelength blocking. What this means is that for a multi-hop connection, there needs to exist a continuous, single wavelength channel from the source to the destination node. To illustrate this point, consider figure 3.10. Say in this case only two wavelength channels exist on each connection between nodes. Say A has established a connection with B on wavelength one and B has established a connection with D on wavelength 2. If node A wants to connect to node D, it is not possible for this connection to be made along the shortest path. In short, a common wavelength must be available from source to destination. This affect can be managed more easily in classical networks through the use of transponders for wavelength conversion [35]. However these transponders incur significant loss penalties [72], which are unacceptable in the context of quantum communications.

Wavelength continuity in ensured between links in this system by iterating through the K-shortest paths and three-dimensional "channels" array. The shortest wavelength is chosen and each hop is checked sequentially to see if that wavelength is available. If it is not available for all hops, the next wavelength is checked. If no wavelengths are available the next shortest path will be checked. If no wavelengths are available for any path up to and including K, the connection attempt is noted and aborted. The "channels" matrix is a global data structure that is available to all nodes on the network.

3.2.3 K-Shortest Paths

The algorithm used by this codebase is loosely based off Yen's algorithm and uses priority queues to iteratively determine the shortest path. The graph is represented as a dictionary of dictionaries, where each key represents a node, and the corresponding value is another dictionary that maps the node's neighbors to their edge weights. The function initializes an empty list called paths and a heap (a priority queue) that contains a tuple, (0, [source]). The first element in the tuple is the distance from the source node to the current node, which is 0 since it starts from the source node, and the second element is a list containing only the source node, since the algorithm starts from it.

The function then enters a loop that continues while there are still items in the heap and the number of paths found so far is less than k. In each iteration of the loop, it pops the smallest element from the heap. It then checks the last node in the path and if it is the target or destination node and adds it to the list of paths. If the last node does not match the target node, the algorithm continues by examining the neighbors of the last node and for each neighbor, it checks if it's not already in the path. If the neighbor is not in the path, the function creates a new path by adding the neighbor to the end of the current path, and computes the new distance. The new path and distance are then added to the heap as a tuple.

Once the loop completes, the function returns the list of paths that have been found, which is of maximum length, k. The pseudo code is as follows:

Algorithm 2 K-Shortest paths

```
$heap \leftarrow [(0, [source])]$
while $heap, len(paths) < k$ do
    $d, path \leftarrow$ heappop(heap)
    $node \leftarrow$ path[-1]
    if $node = target$ then paths.append(path)
    end if
    for each neighbour v of u do:
        if $v \notin path$ then
            $newpath \leftarrow path + neighbour$
            $newdistance \leftarrow d + G.edges(u, v)$
            heappush(heap,( new distance, new path))
        end if
    end for
end while
```

3.3 Traffic model

Nature generates may random processes that follow Poisson distributions. The work of Erdos and Renyi shows that network departures follow the same distribution [73]. Let N be a set randomly distributed, unconnected nodes. At t = 0, two nodes are selected and connected. After this connection is made, again two nodes are chosen for connection. This process is repeated M times. This process results in M connections. If M is large compared to N, the graph will be highly connected. If M is small relative to N, there will be many isolated nodes.

Let N_p be the total number of connections that can be formed from N nodes,

$$N_p = \frac{N}{2}(N-1) \tag{3.8}$$

Since M connections have been selected, the probability P_e of a randomly selected connection being made is:

$$P_e \frac{M}{N_p} = \frac{2M}{N(N-1))} \tag{3.9}$$

Focusing on a specific node v, that is chosen randomly, the total number of connections that could include v as a source or destination is $N-1$ (as it can not connect to itself). However within the M connections node n will only appear k times. Thus the probability of v being present in k of the $N-1$ pairs is given by:

$$P(k) = \binom{N-1}{k}(p_e)^k(1-p_e)^{N-1}) \tag{3.10}$$

In the limit where $N \rightarrow \inf$, the expression collapses to the following expression, which is the Poisson distribution:

$$P(k) = \frac{\lambda^k}{k!}e^{-}\lambda \tag{3.11}$$

Thus, the number of requests made by a given node at time step can be sampled from the Poisson distribution for a given λ, using NumPy.Poisson. For each connection request a random node is then chosen to be the destination node. The "Generate_users" function samples the Poisson distribution for each node, then for whatever number of connections are to be made for that node a, chooses a destination node is chosen for each connection. The function returns a list of all nodes to be connected.

3.4 Wattson-Strogatz topology

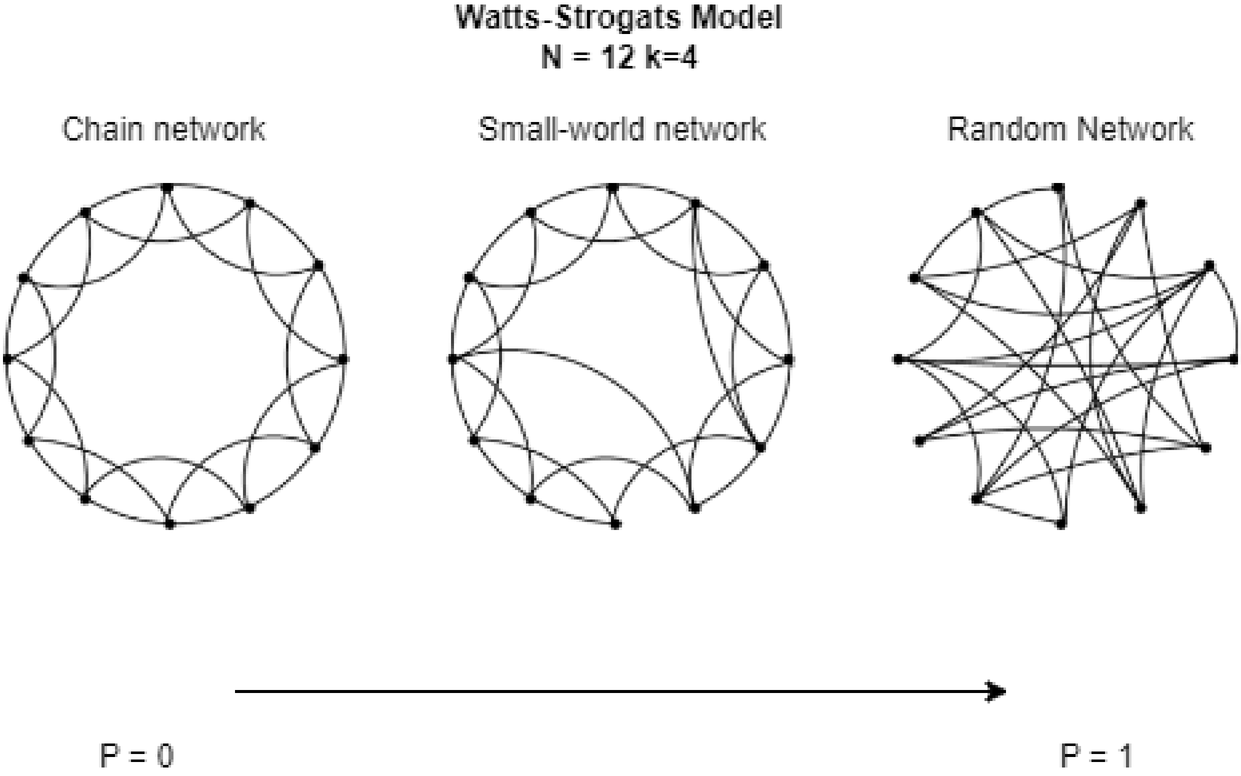


Figure 3.11: Watts-Strogatz network with N=12 and $k_{ws} =$

A Watts-Strogatz network is a type of small-world network with ability to keep a network globally connected, while still generating local clustering [74].

Real-world networks exhibit a degree of randomness in their connectivity, meaning that connections are not predictable. The Watts-Strogatz model includes a random rewiring step that captures this aspect of real-world networks. The model also allows for a smooth transition between a highly clustered network and a more random network. This is pertinent to model as real-world networks also exhibit varying levels of clustering depending on the context.

A key aspect of Watts-Strogatz networks that make them suitable for this thesis is its degree distribution property [74]. The degree of a node is the number of distinct nodes it is connected in a network. A Watts-Strogatz network has parameters N being the number of nodes, k_{ws} being the order of the network and p_{ws} being the probability of a node rewiring. For such a network, when $p_{ws} = 0$, the network defaults to a ring lattice network where each node is interconnected with k_{ws} nearest neighbours(see figure 3.5 left). In this scenario, the degree of each node is equal to k_{ws}. As the network is increasingly rewired, the degree of each node changes but the average degree of the system as a whole remains unchanged.

Nodes of different degrees are physically different and require different switching hardware. To demonstrate this, the node shown in figure 3.1 is of degree 2. The degree distribution in the case of the ring lattice is simply a Dirac delta function centered at K. The degree distribution for 0<p_{ws} <1 can be written as, [75]

$$P(k) \approx \sum_{n=0}^{f(k,K)} \binom{K/2}{n} (1-p_{ws})^n p_{ws}^{K/2-n} \frac{(p_{ws}K/2)^{k-K/2-n}}{(k-K/2-n)!} e^{-p_{ws}K/2} \tag{3.12}$$

where k_{ws} is a nodes has or its degree. Watts-Strogatz networks are relatively Homogenous meaning that each node shares a similar degree.

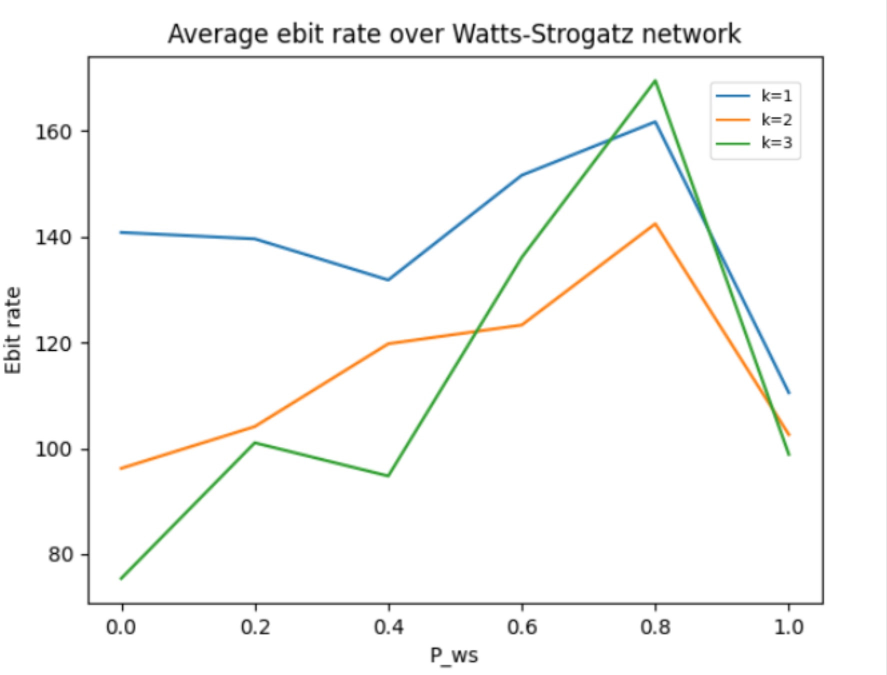


Figure 3.12: Diagram indicating average ebit rate across a Watts-Strogatz network with parameters M=8 k_{WS}=3, d_n=0.2, T_1=2.68e6, T_2=3.3e3

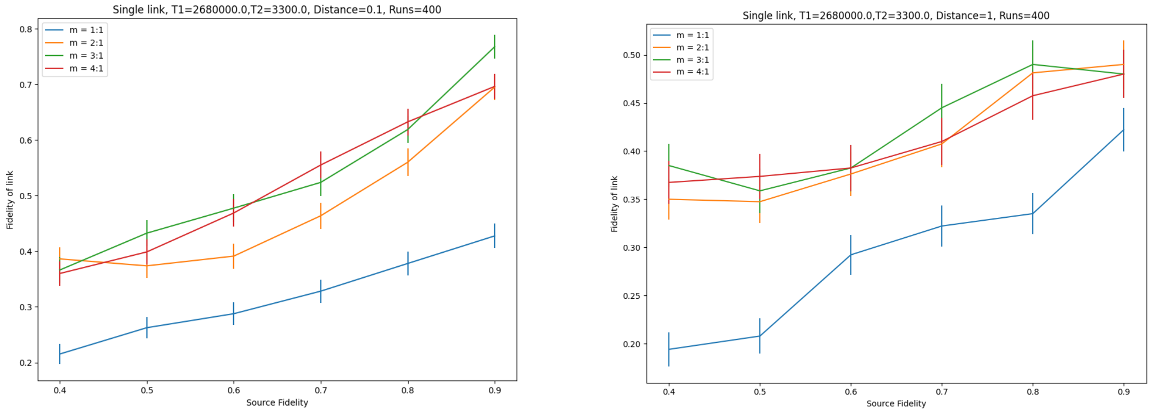
This project was scaled up from a six node case to a Watts Strogatz network of varying sizes. The implementation of the Watts-Strogatz network uses NetworkX's watts_strogatz_graph() from the networkx.generators.random_graphs module. A utility function called "create_WS_graph" is defined that returns the output of the NetworkX function as an adjacency list for ease of interoperability with the rest of the codebase. The function takes four arguments:

- N: the total number of nodes in the network
- k_{ws}: the number of nearest neighbors to connect each node to in the initial ring topology
- p_{ws}: the probability of rewiring for each edge

- seed (optional): the random seed to use for the network generation

4 Results

4.1 Distillation protocol



(a) d = 0.1km

(b) d = 1km

Figure 4.1: Figures showing distillation for dynamical electron spin NV-centres

Before scaling up the system to a a complicated topology, it had to be tested thoroughly for the two node case. Figure 3.7 plots fidelity vs distance for the two node case for the large T_1 and T_2 times associated with dynamical decoupling. It is clear that the distillation protocol results in increased fidelity for the output entangled state ρ compared to input state $\psi = F_i|\psi^+\rangle + (1 - F_i)|00\rangle$. Below an input fidelity of 0.5, attempts of distillation do not increase the output fidelity. This is as predicted by Deutcsch et al [61]. Increasing the number of links combined in the distillation process has the effect of increasing the output fidelity when compared to distillation protocols that combine fewer links. The 3:1 and 4:1 cases tend to have similar improvements in fidelity, and pure entanglement has a similar output fidelity as the 2:1 case. Over short distances, a NV centre that incorporates dynamical decoupling would not be expected to have a large amount of decoherence, hence why the direct entanglement performs well.

Figure 4.1 demonstrates the effect of distillation when the shorter T_1 and T_2 times

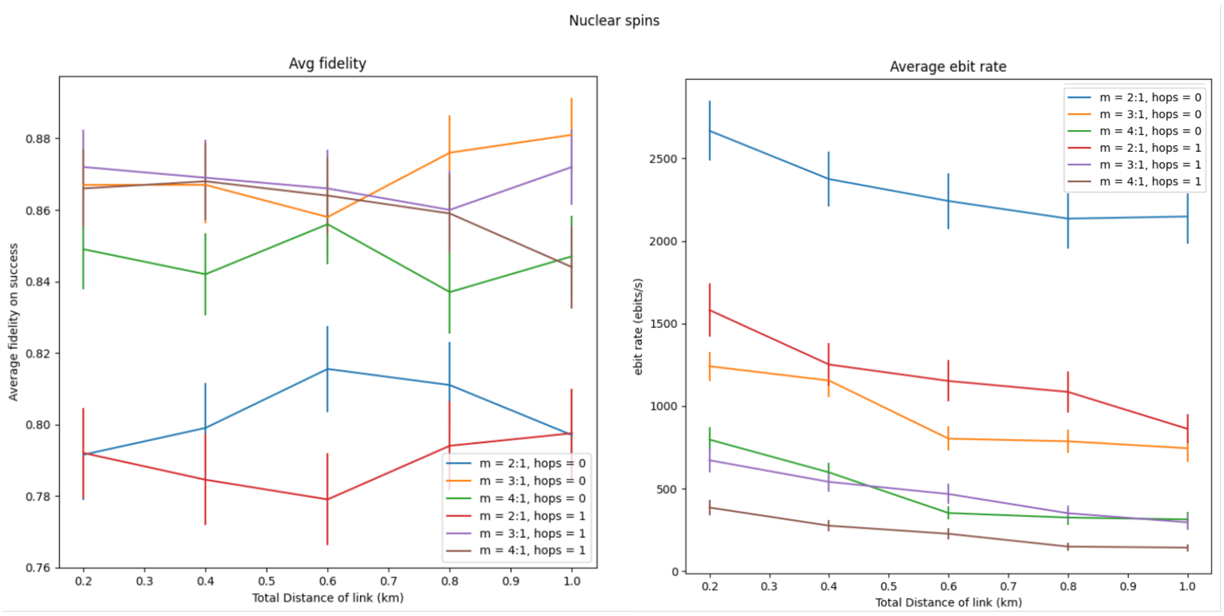


Figure 4.2: Plot showing varying numbers of hops and distillation runs on fidelity and ebit rate. Here $F_i = 0.8$

associated with electron spins of NV centres are used. Sub figure A) demonstrates an increase in fidelity for each distillation case, with the 3:1 and 4:1 cases performing the best. The increase in fidelity is quite dramatic here compared to entanglement with no distillation, as this is the first case shown where significant amounts of decoherence is occurring. In sub figure b, seemingly random fidelities are observed with a slight uptrend in output fidelity F with increasing source fidelity, F_i. As the input fidelity values begin below 0.5, the output fidelity can not rise above 0.5. After this, the output fidelities begin to improve but not significantly regardless of the number of distillation runs. The decoherence of the quantum memories is the limiting factor caused by the additional time taken for the qubits to propagate from node A to node B. Comparing with figure 3.7 demonstrates that the key factor causing the loss of fidelity is the type quantum memory in use. For the remaining experiments F_i is set to 0.994, unless otherwise stated [76].

Figures 4.2 and 4.3 show the impact of different numbers of distillation runs for two node link, with artificially injected loss for different number of ROADM "hops". Here, (A$\rightarrow$B) would be 0 hops and (A$\rightarrow$C$\rightarrow$B) would be 1 hop. This graph considers the large T_1 and T_2 times associated with the dynamically decoupled nuclear spins of an NV centre. The fidelity plots show that the 3:1 and 4:1 cases perform significantly better than the 2:1 case in terms of improvement of fidelity. It should also be noticed that distance has little effect on output fidelity, with some lines increasing with distance and others decreasing. The extra distance would not be expected to have significant impact on fidelity here as the T_1 and T_2 times are orders of magnitude greater than any propagation time. Another noticeable result on the fidelity graphs is that the 3:1 case performs either the same or slightly better than the 4:1

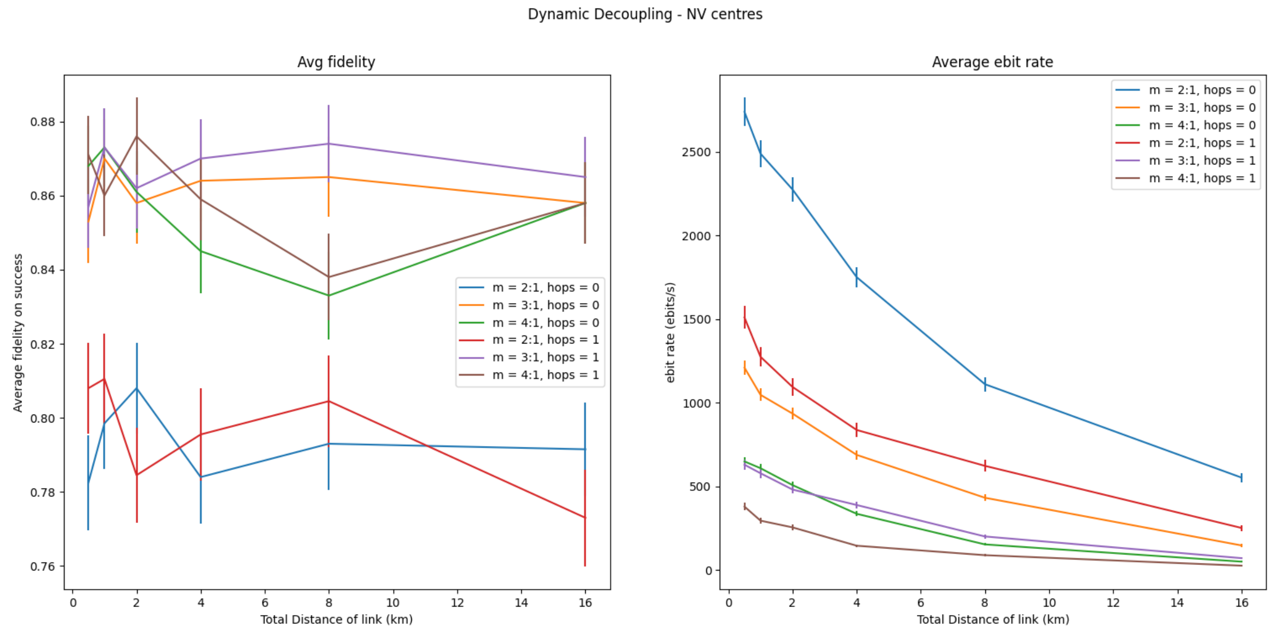


Figure 4.3: Extends 4.2 to long distances. Plot showing varying numbers of hops and distillation runs on fidelity and ebit rate. $F_i = 0.8$

case. It is possible that this is due to the asymmetric nature of the distillation protocol. Most implementations of distillation first distil the 4:1 case by first producing two 2:1 distilled connections and then combining those connections to create a 4:1 distilled link [12–14]. However, the implemented method iteratively adds links until 4 links have been distilled together.

Similarly, to how distance has little bearing on the output fidelity, the number of hops, and node loss by extension, has little impact on output fidelity with some outliers. Node loss should decrease ebit rate but not fidelity. This is the expected result as an increased number of hops injects loss into the system, not noise. The distillation protocol restarts when a photon is is not received, so no additional decoherence occurs than if the protocol succeeded on its first run. It is important to note that decoherence is caused by a particles interaction with its environment [77], so there would be some degree of decoherence when a photon interacts with the ROADM. However, for the purposes of modelling large quantum networks, only the most significant decoherence factors were considered.

The difference between the plotted cases is more apparent on the graph of ebit rate and distance. The clear winner in terms of ebit rate is the 0 hop case with two links distilled into one. This ebit rate drops with distance, but not as much as the fidelity loss with distance. Here, there is very large decrease in ebit rate with the number of hops. Comparing each n:1 case, it can be seen that he ebit rate drops by approximately 1/2 with the addition of 1 hop.

In both the 1 hop and 0 hop cases, the decrease in ebit rate from the 3:1 to the 4:1 case is

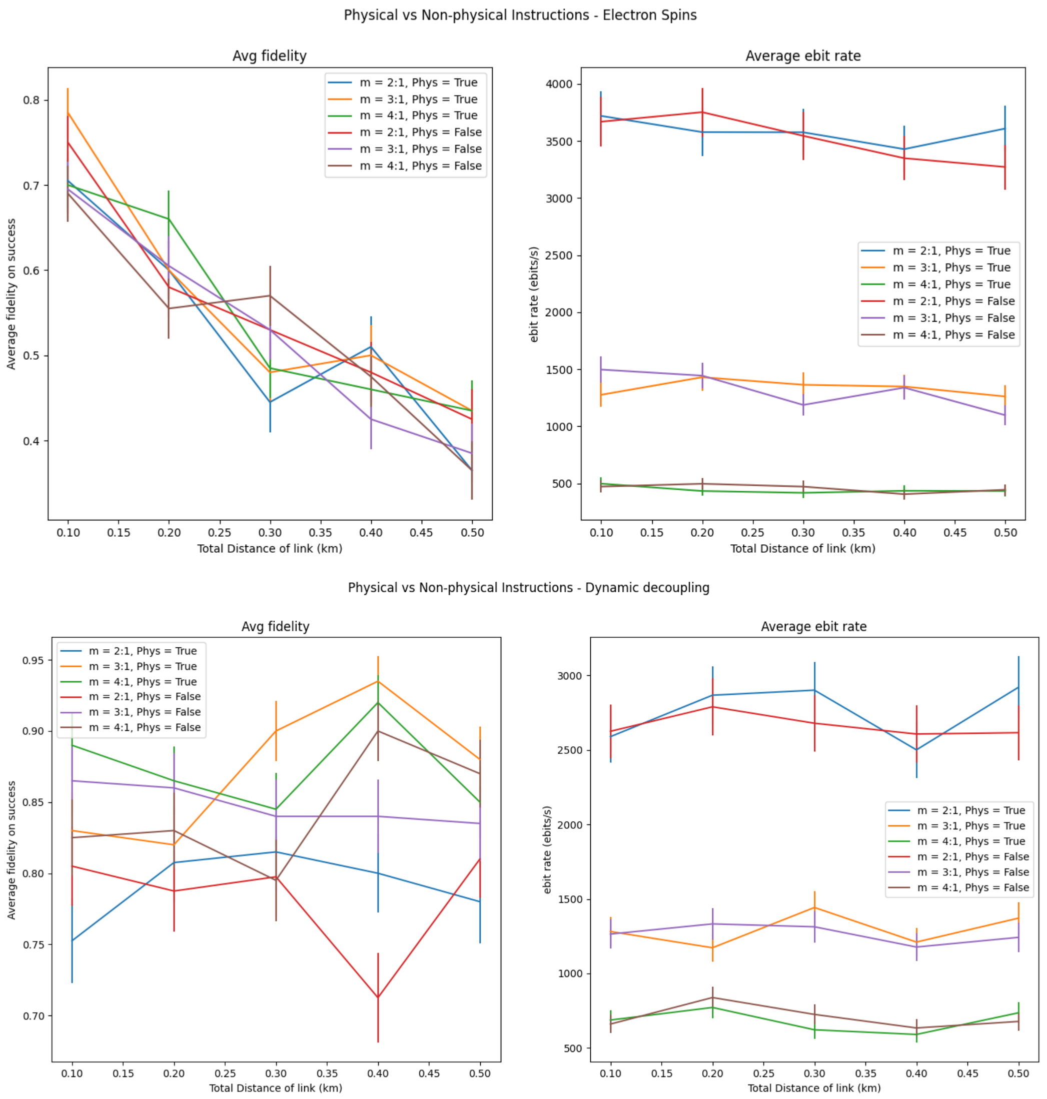


Figure 4.4: Diagram highlighting the effect of ideal vs physical instructions for electron spins (top) and Nuclear spins with dynamical decoupling(bottom).

significantly less than the decrease from the 2:1 to the 3:1 case. While the fidelity of the 2:1 distillation protocol is worse than its counterparts, due to its clear lead in ebit rate, it is used for the remaining experiments with the larger six-node and Watts-Strogatz networks.

NetSquid allows the modelling of Idealised gate operations (non-physical) or operations that take a predetermined time to complete apply the noise model to the quantum memories with the Δt parameter being the time since the quantum memory was last interacted with. The results here are not conclusive for either the electron or nuclear spins. It does appear that changing the instructions from physical to non physical has some very minor effect particularly on the 2:1 case on both graphs, but this difference lies within the standard error of the means for both fidelity and ebit rate plots.

This result is reasonable as the time frame of the instructions is in the 10s of nanoseconds, so delays incurred would be very minimal. Similarly noise model used is unlikely to add noise

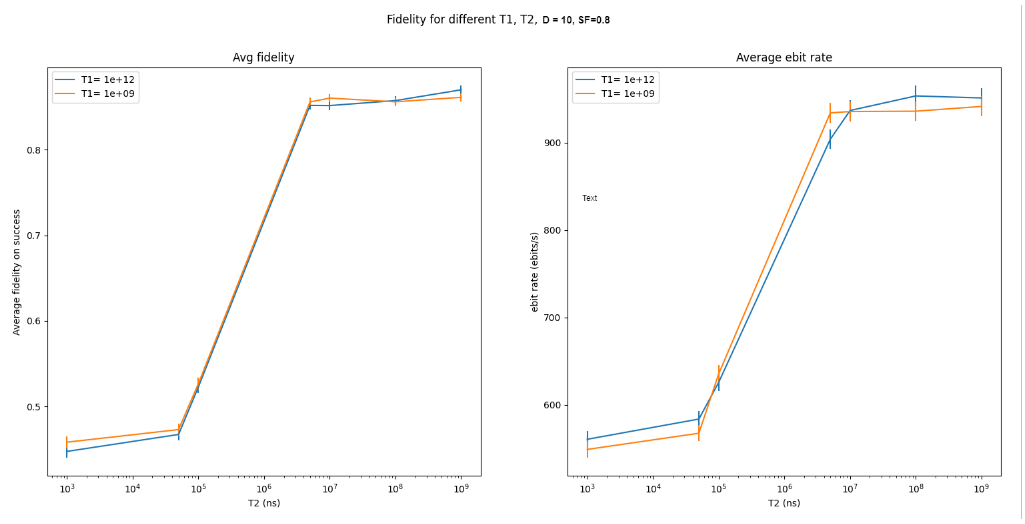


Figure 4.5: Diagram that plots T2 against fidelity and ebit rate over a distance of 10km.

over such small time frames. Due to the little difference between these two instruction types, physical instructions were used for further experiments as it more accurately models a physical quantum memory.

Plotting distance against fidelity and ebit rate is informative for a given architecture. However for the very large values of T_1 and T_2 associated with dynamical decoupling, there is little effect on the output fidelity unless the distance, and thus propagation time of the connection is very large. In such scenarios the photon loss is so high that an unmanageable amount of runs must be performed to get accurate results. Figure 4.5 increases T_2 logarithmically over a fixed distance of 10km and plots the results for $T_1 = 10^9, 10^{12}$. In this plot the output fidelity is limited to below the 0.5 threshold before $T_2 > 5 \times 10^4$. This threshold is precisely as expected for this distance. As the speed of light in fibre for these simulations is 200,000km/s, $\frac{10}{200{,}000 \times 10^9} = 5 \times 10^4$. There are other delays to account for, but the propagation time is the most significant. The fidelity and ebit rate both grow at similar rates until $T_2 > 5 \times 10^6$. The effect of T1 is seemingly negligible on both plots, with a slight increase of ebit rate for higher T_1 values, although, the standard error of the mean for these data points is overlapping.

The T_2 values of interest are further investigated in figure 4.6. This plot considers the T_2 values where the greatest amount of change in output occurred in finer detail, with the additional plot of a much lower $T_1 = 5 \times 10^6 ns$. This plot shows that T_1 does in fact affect the output fidelity of the network, although its effect is considerably less potent than that of T_2. The relative effect of T_1 on the output fidelity is strongest when $T_1 \rightarrow T_2$. When looking at $T_2 = 5 \times 10^6 ns$ in figure 4.5, it can be seen that the ebit rate for the shorter T_1 time is actually higher than the the ebit rate of the shorter T_1 time. This only occurs for this single data point but is an interesting result. This result is corroborated by figure 4.6, where the same effect can be seen at $T_2 = 5 \times 10^6 ns$. It does not hold that increased decoherence

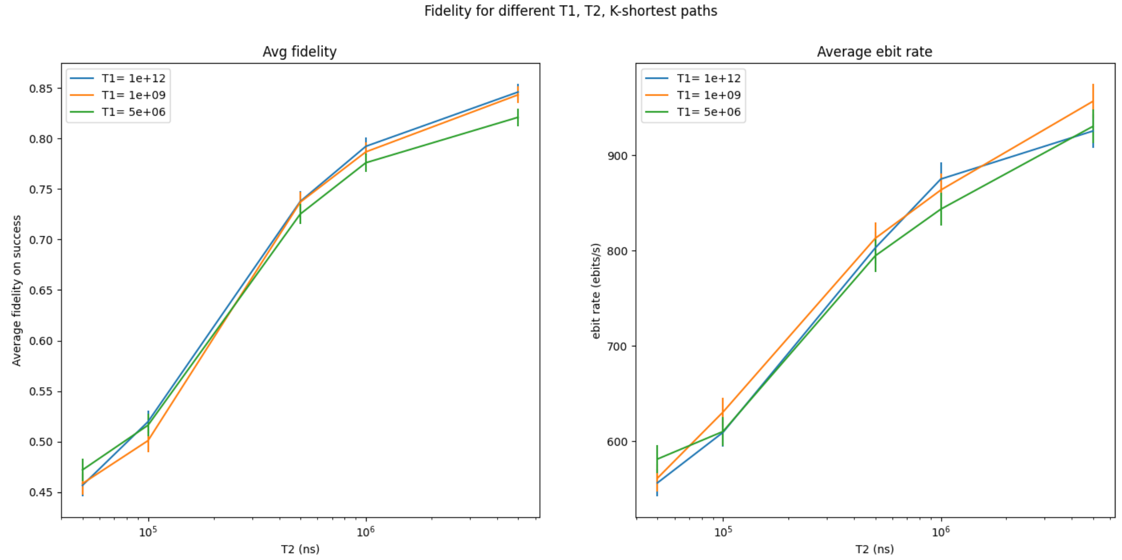
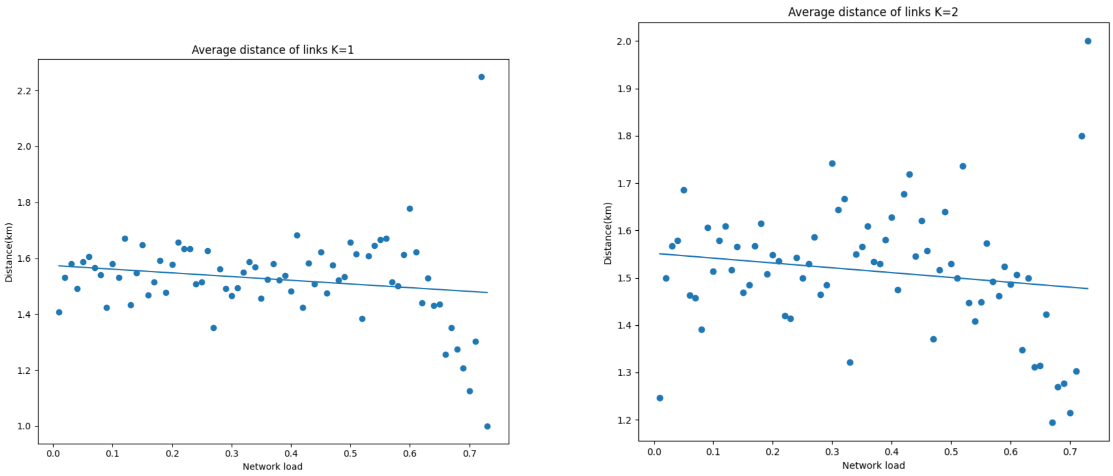


Figure 4.6: "Zoomed in" plot of 4.5. Compares T2 against fidelity and ebit rate over a distance of 10km for the values of interest.

would increase r_e at any point with the implemented "T1T2NoiseModel", and futher investigation is needed into this phenomenon. It is possible that this is a strange coincidence due to the a classical computer's "pseudo-random" pseudo-random nature.

4.2 6 node network



(a) K=1

(b) K=2

Figure 4.7: Diagram plotting average connection distance against network load for K=2 and k =3

In the six node network, interactions between nodes and shared resources become apparent. When k=1 is considered in figure 4.7 a), the average distance remains mostly constant. As the network becomes increasingly blocked, less long distance connections become available and the only connections that can be established are the short ones. When multiple shortest paths are considered, a more diverse array of distances can be chosen by the routing algorithm, and the data points become more spread out as a result. Overall, the average distance of a link is about the same but slightly smaller in case b). In both cases there are one or two outliers of very large distances that get through at the end.

The relationship between memory blocking and wavelength blocking in the six node network is demonstrated by the histograms in figures 4.8 and 4.9. As there is a total 8 links between the nodes (see 3.10), the total number of channels available to the six node network is 8N, where N is the number of distinct wavelength channels between adjacent nodes. As two memory positions on two nodes are tied up for each distillation process, there will be guaranteed memory blocking if $8N > \frac{Numnodes \times M}{4}$, where M is the memory size at each node. In cases where $8N >> \frac{Numnodes \times M}{4}$ there is be a lot of memory blocking and very little wavelength blocking. The opposite can be said for when $8N < \frac{Numnodes \times M}{4}$. Figures4.8 and 4.9 show circumstances where both memory and wavelength blocking are prominent in the network.

In these circumstances, It can be seen that increasing the number of channels, N, while leaving M constant increases the amount of memory blocks, but also shifts back the peak of the memory blocking distribution so that most of the memory blocking actually occurs significantly earlier. In cases where $8N \leq \frac{Numnodes \times M}{4}$, the memory blocking ramps up quickly

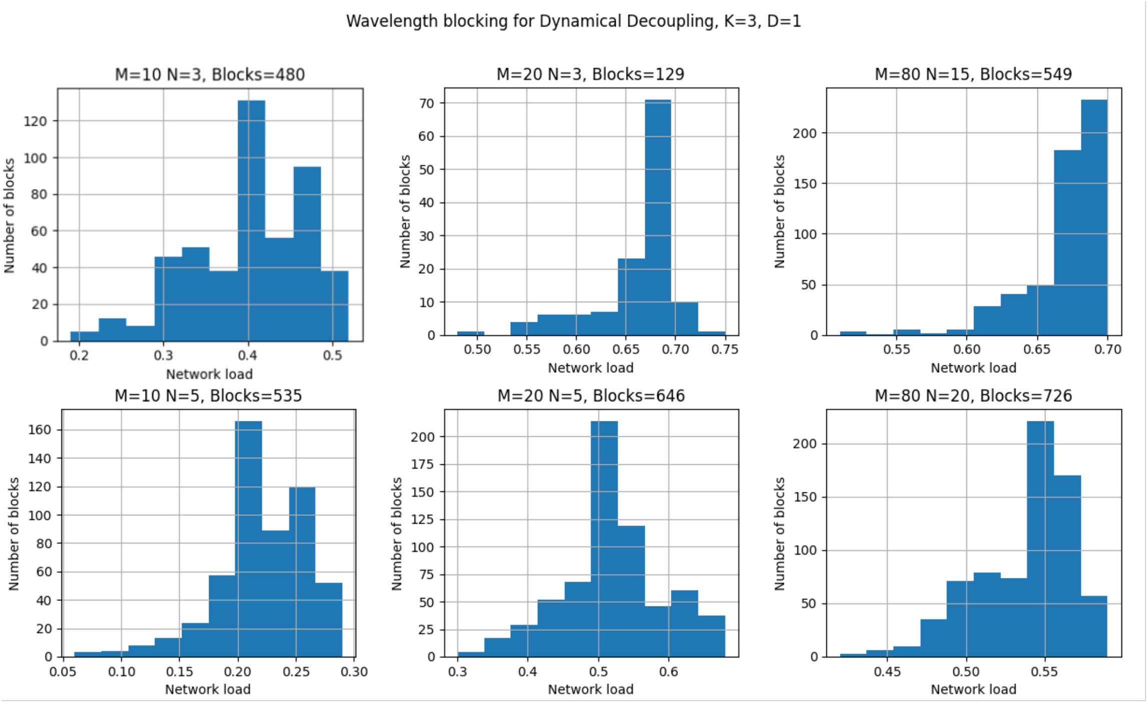


Figure 4.8: Histogram showing memory blocking for a 6 node network

to a peak, until all memory positions are fully blocked up and the simulation ends.

Interestingly, in figure 4.9 increasing the number of channels available also pushes back the peak of the distribution of wavelength blocks in figure 4.9, although it drops the frequency of this type of blocking. This makes sense, as although there an abundance of wavelengths available in comparison to memory positions, certain paths can become blocked quickly depending on network traffic. It is important to note that the x axis is not the same for all of these histograms. This is due to the varying end conditions of the simulations shown in3.8 for when memory or wavelength blocking occurs.

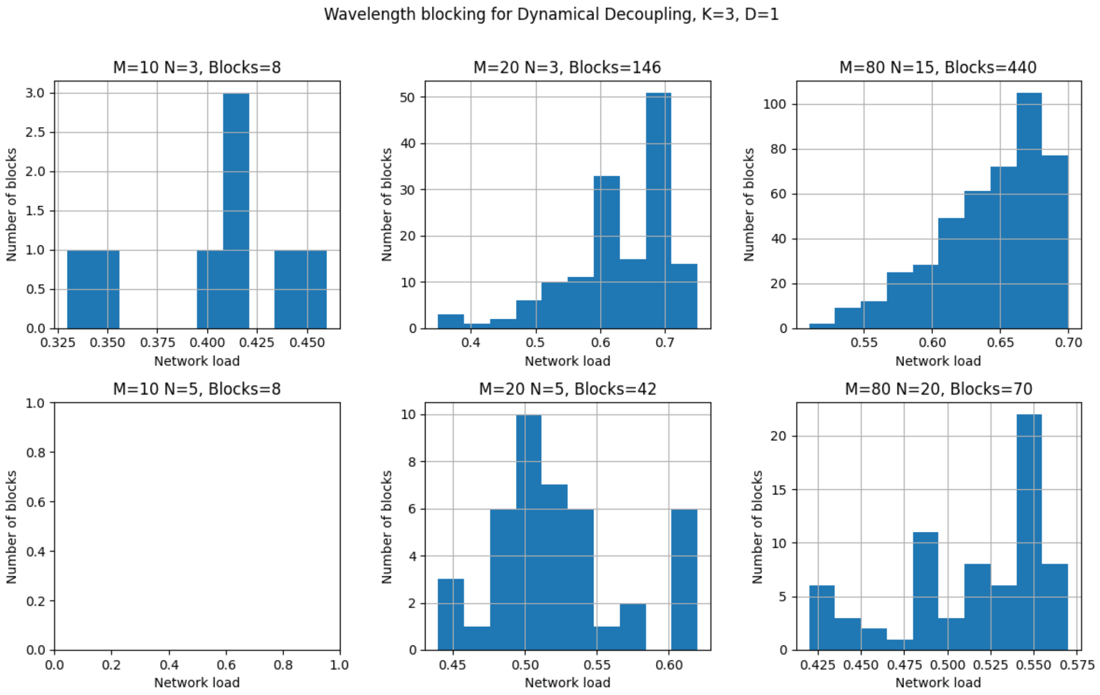


Figure 4.9: Histogram showing wavelength blocking for a 6 node network

4.3 Watts-Strogatz Network

For the simulations of the Watts-Strogatz network, the results are compared from more of a macro level as there are 3 additional parameters: N, k_{ws} and p_{ws} to consider for each Watts-Strogatz graph. These simulations forcus on how the probability of a link being rewired, p_{ws}, impacts the overall performance of a network.

Figure 4.11 a) shows that p_{ws} has no impact on the output fidelity of a network when only the shortest path is considered in routing. This is logical as even though areas that are

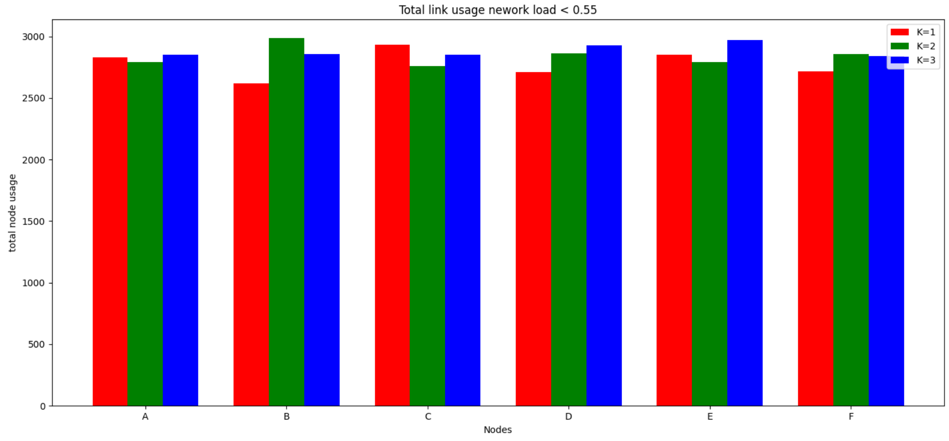
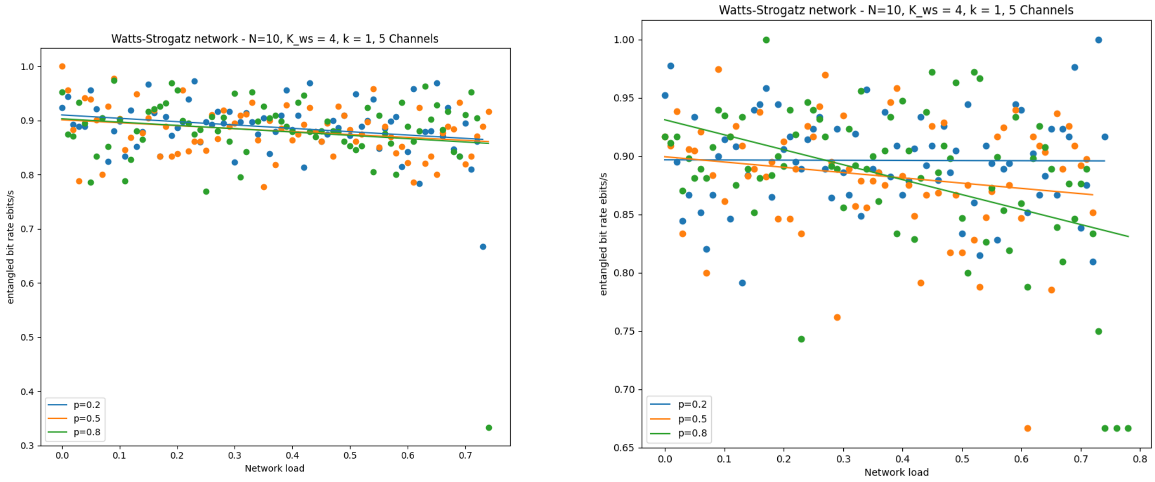


Figure 4.10: Diagram showing overall node usage up for a six node network up to network load <0.55

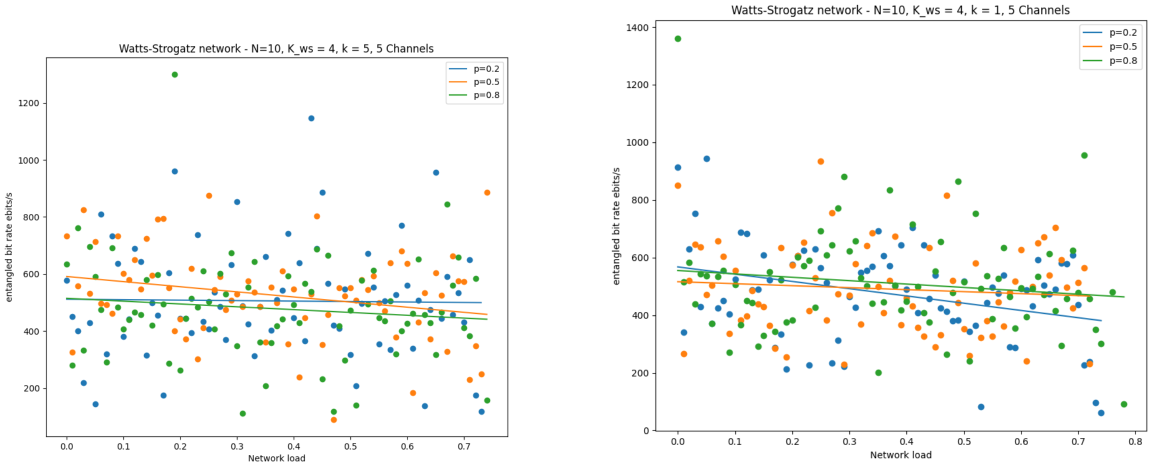


(a) K = 1 (b) K = 5

Figure 4.11: Fidelity vs Network load for a Watts-Strogatz graph with parameters: N = 10, k_{ws}=4, M = 100, C = 5, T_1 = 1e12, T_1= 1e6, d_l = 10

highly connected will become blocked first, once they become considerably blocked, no connections will make it through and the network will simply allow connections elsewhere to be made.

4.11 b) presents a different narrative. Nodes in graphs with large $p_{ws} = 0.8$ have increased options to get from one side of the graph to the other (see 3.11), and thus initially choose shorter paths, resulting in a higher fidelity. However. these "shortcut connections" quickly become filled, and what remains is a graph that is similar to a one with a smaller p_{ws} value, that has less links available. This leads to a lower output fidelity as the network becomes increasingly loaded. For $p_{ws} = 0.5$ a similar but less substantial effect can be seen.



(a) K = 1 (b) K = 5

Figure 4.12: Average ebit rate, r_e vs Network load for a Watts-Strogatz graph with parameters: N = 10, $k_w s$=4, M = 100, C = 5, T_1 = 1e12, T_1= 1e6, d_l = 10

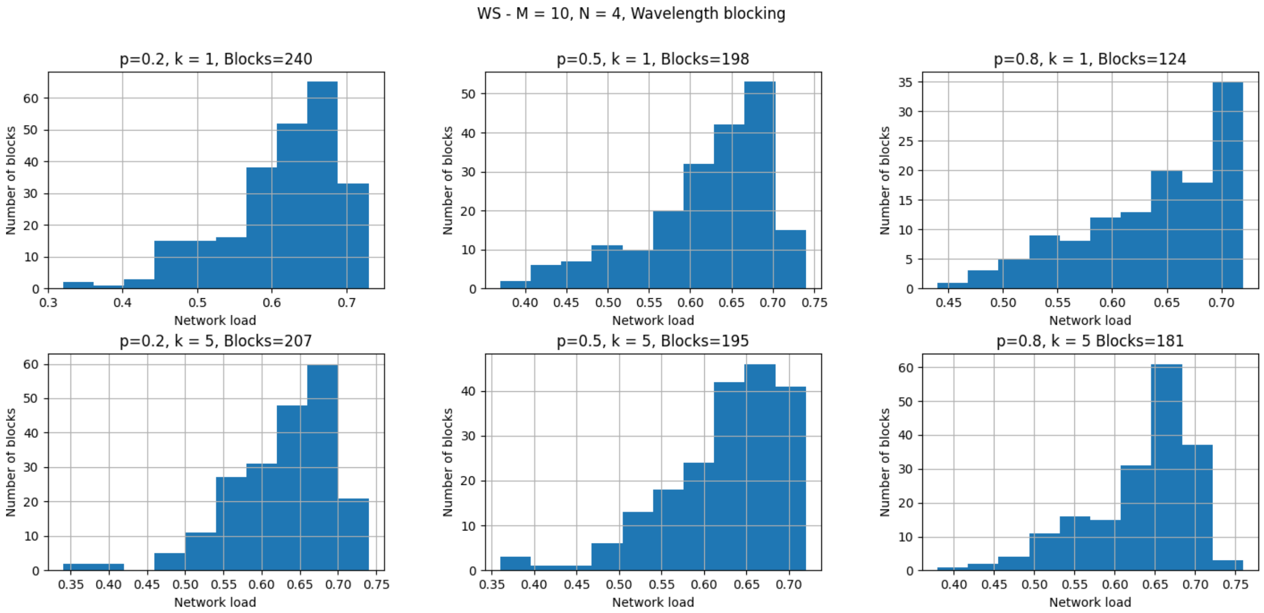


Figure 4.13: Histogram showing the disribution of wavelength blocking for a Watts-Strogatz graph with parameters: N = 10, k_{ws}=4, M = 100, C = 5, T_1 = 1e12, T_1= 1e6, d_l = 10

Figure 4.13 shows cases of wavelength blocking in a Watts-Strogatz network for varying paramaters of p_{ws} and K. For the less random graphs where $p_{ws} = 0.2$ the amount of blocking is less in the graphs that consider the k=5 shortest paths. This is due to the fact that for the k = 1 case, once the few "shortcut paths" are used up, the routing algorithm by the way it is written will still consider these shortcut paths to be the shortest path, even though it is blocked. In the k = 5 case, these shortcut paths will be chosen sooner, but once they are unavailable, will not necessarily result in a blocking signal from the routing protocol.

As the network approaches a random graph where $p_w s = 0.8$, the k = 5 graph begins to see much more blocking than its k = 1 counterpart. This is likely due to the increased clustering and isolated nodes in this type of graph. If the routing algorithm considers up to 5 paths to get to a node in a network with only 10 nodes, there is a likelihood that an isolated node will be routed through. If this happens enough times a node can become completely cut off from the graph and wavelength blocking becomes a certainty whenever that node is chosen. This explanation lines up with the histogram in the bottom left showing a large spike at a network load of 0.65, where a lot of blocking suddenly occurs despite there being a large number channels available in this network relative to the quantum memories.

Considering each value of k on its own, the amount of blocking always decreases as $p_{ws} = 0.8$ increases. This is logical as graphs that are more interconnected should have less blocking.

Figure 3.12 shows the average ebit rate of a Watts-Strogatz graph with parameters M=8

k_{WS}=3, d_n=0.2, T_1=2.68e6, T_2=3.3e3. It is interesting to see that larger numbers of K decrease the ebit rate, but the explanation of the previous figure accurately sums up why this is the case. Once the shortcut paths are all blocked, the routing algorithm can only choose short paths for connections which makes the average connection shorter, which therefore has a higher throughput. As the Watts-Strogatz graph becomes increasingly random, the larger K values perform relatively better than they did before.

Attempting to simulate Watts-Strogatz graphs with N>12 caused the device used for this thesis to become unusable. This is likely a limitation of netsquid as the quantum formalism used for this project has a limited number of qubits it can keep track of at one time. However, the important aspects of a Watts-Strogatz network were still simulated with a 10 node network.

5 Conclusion

Simulating single links and a six-node mesh network provided the best insight into the limitations of the scalable components this thesis focused on. Quantum states stored in NV centres that make use of dynamical decoupling proved to be able to maintain their source fidelity of 0.8 at 750 ebits/s over distances of 16km. The use of ROADMs as routing switches for quantum connections proved to be functional but very lossy. When passing through a single ROADM "hop" the ebit rate of the same link of 16km had its ebit rate reduced by approximately a factor of two. However, it managed to maintain close to its source fidelity. This one example is indicative of a trend present in all of the results that is a very high loss in throughput associated with ROADMs. However, in circumstances where low throughput is acceptable, this solution is effective at both maintaining fidelity and bypassing the need to waste memory resources on interconnecting nodes.

The overall goal or vision of the thesis was to progress towards a realistic implementation of a large-scale quantum internet with minimal loss in fidelity that is capable of generating entanglement faster than it is lost. Furthermore, a network architecture was created that is capable of simulating complex protocols requiring the inter-operation of classical and quantum channels, while capturing the general effect of decoherence on a quantum network through approximation, and identifying challenges associated with variable-scale quantum networks.

In this thesis, quantum networks of varying size and specification were analysed with respect to their critical metrics: fidelity and the rate of entanglement. As the networks grew in size and complexity, it became apparent that there were many more metrics of significant importance, such as network load, blocking and clustering. Only by thoroughly testing a network to the limits of a wide array of these metrics can a network protocol be said to be robust. Although a simulation can not capture all of the behaviours of a system, simulating a network in such a way can provide insight and direction to engineers and scientists on the path to travel when physically realising such networks.

Bibliography

[1] P. Kómár, E. M. Kessler, M. Bishof, L. Jiang, A. S. Sørensen, J. Ye, and M. D. Lukin, "A quantum network of clocks," *Nature Physics*, vol. 10, no. 8, pp. 582–587, Aug. 2014, number: 8 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/nphys3000

[2] R. Kumar, H. Qin, and R. Alléaume, "Coexistence of continuous variable QKD with intense DWDM classical channels," *New Journal of Physics*, vol. 17, no. 4, p. 043027, Apr. 2015, publisher: IOP Publishing. [Online]. Available: https://dx.doi.org/10.1088/1367-2630/17/4/043027

[3] M. Pittaluga, M. Minder, M. Lucamarini, M. Sanzaro, R. I. Woodward, M.-J. Li, Z. Yuan, and A. J. Shields, "600-km repeater-like quantum communications with dual-band stabilization," *Nature Photonics*, vol. 15, no. 7, pp. 530–535, Jul. 2021, number: 7 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41566-021-00811-0

[4] J. I. Cirac, A. K. Ekert, S. F. Huelga, and C. Macchiavello, "Distributed quantum computation over noisy channels," *Physical Review A*, vol. 59, no. 6, pp. 4249–4254, jun 1999. [Online]. Available: https://doi.org/10.1103%2Fphysreva.59.4249

[5] A. Dahlberg, M. Skrzypczyk, T. Coopmans, L. Wubben, F. Rozpędek, M. Pompili, A. Stolk, P. Pawełczak, R. Knegjens, J. d. O. Filho, R. Hanson, and S. Wehner, "A Link Layer Protocol for Quantum Networks," in *Proceedings of the ACM Special Interest Group on Data Communication*, Aug. 2019, pp. 159–173, arXiv:1903.09778 [quant-ph]. [Online]. Available: http://arxiv.org/abs/1903.09778

[6] S. Das, M. S. Rahman, and M. Majumdar, "Design of a Quantum-Repeater using Quantum-Circuits and benchmarking its performance on an IBM Quantum-Computer," Oct. 2021, arXiv:2009.04584 [quant-ph]. [Online]. Available: http://arxiv.org/abs/2009.04584

[7] J.-W. Ji, Y.-F. Wu, S. C. Wein, F. K. Asadi, R. Ghobadi, and C. Simon, "Proposal for room-temperature quantum repeaters with nitrogen-vacancy centers and

optomechanics," *Quantum*, vol. 6, p. 669, Mar. 2022, arXiv:2203.06611 [quant-ph]. [Online]. Available: http://arxiv.org/abs/2203.06611

[8] D. M. Irber, F. Poggiali, F. Kong, M. Kieschnick, T. Lühmann, D. Kwiatkowski, J. Meijer, J. Du, F. Shi, and F. Reinhard, "Robust all-optical single-shot readout of nitrogen-vacancy centers in diamond," *Nature Communications*, vol. 12, no. 1, p. 532, Jan. 2021, number: 1 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41467-020-20755-3

[9] G. de Lange, Z. H. Wang, D. Ristè, V. V. Dobrovitski, and R. Hanson, "Universal Dynamical Decoupling of a Single Solid-State Spin from a Spin Bath," *Science*, vol. 330, no. 6000, pp. 60–63, Oct. 2010, publisher: American Association for the Advancement of Science. [Online]. Available: https://www.science.org/doi/10.1126/science.1192739

[10] M. H. Abobeih, J. Cramer, M. A. Bakker, N. Kalb, M. Markham, D. J. Twitchen, and T. H. Taminiau, "One-second coherence for a single electron spin coupled to a multi-qubit nuclear-spin environment," *Nature Communications*, vol. 9, no. 1, p. 2552, Jun. 2018, number: 1 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41467-018-04916-z

[11] N. Gisin, N. Linden, S. Massar, and S. Popescu, "Error filtration and entanglement purification for quantum communication," *Physical Review A*, vol. 72, no. 1, jul 2005. [Online]. Available: https://doi.org/10.1103%2Fphysreva.72.012338

[12] B. Regula and R. Takagi, "Fundamental limitations on distillation of quantum channel resources," *Nature Communications*, vol. 12, no. 1, p. 4411, Jul. 2021, number: 1 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41467-021-24699-0

[13] P. C. Humphreys, N. Kalb, J. P. J. Morits, R. N. Schouten, R. F. L. Vermeulen, D. J. Twitchen, M. Markham, and R. Hanson, "Deterministic delivery of remote entanglement on a quantum network," *Nature*, vol. 558, no. 7709, pp. 268–273, Jun. 2018, arXiv:1712.07567 [quant-ph]. [Online]. Available: http://arxiv.org/abs/1712.07567

[14] L. Chen, K. Xue, J. Li, N. Yu, R. Li, J. Liu, Q. Sun, and J. Lu, "A Heuristic Remote Entanglement Distribution Algorithm on Memory-Limited Quantum Paths," *IEEE Transactions on Communications*, vol. 70, no. 11, pp. 7491–7504, Nov. 2022, conference Name: IEEE Transactions on Communications.

[15] K. C. Chen, P. Dhara, M. Heuck, Y. Lee, W. Dai, S. Guha, and D. Englund, "Zero-added-loss entangled photon multiplexing for ground- and space-based quantum networks," 2022. [Online]. Available: https://arxiv.org/abs/2206.03670

[16] X. Liu, J. Hu, Z.-F. Li, X. Li, P.-Y. Li, P.-J. Liang, Z.-Q. Zhou, C.-F. Li, and G.-C. Guo, "Heralded entanglement distribution between two absorptive quantum memories," *Nature*, vol. 594, no. 7861, pp. 41–45, Jun. 2021, number: 7861 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41586-021-03505-3

[17] R. S. Tessinari, O. Alia, S. K. Joshi, D. Aktas, M. Clark, E. Hugues-Salas, G. T. Kanellos, J. Rarity, R. Nejabati, and D. Simeonidou, "Towards Co-Existence of 100 Gbps Classical Channel Within a WDM Quantum Entanglement Network," in *2021 Optical Fiber Communications Conference and Exhibition (OFC)*, Jun. 2021, pp. 1–3.

[18] E. Hugues-Salas, O. Alia, R. Wang, K. Rajkumar, G. T. Kanellos, R. Nejabati, and D. Simeonidou, "11.2 Tb/s Classical Channel Coexistence With DV-QKD Over a 7-Core Multicore Fiber," *Journal of Lightwave Technology*, vol. 38, no. 18, pp. 5064–5070, Sep. 2020, conference Name: Journal of Lightwave Technology.

[19] D. D. Awschalom, R. Hanson, J. Wrachtrup, and B. B. Zhou, "Quantum technologies with optically interfaced solid-state spins," *Nature Photonics*, vol. 12, no. 9, pp. 516–527, Sep. 2018, number: 9 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41566-018-0232-2

[20] A. Faraon, C. Santori, Z. Huang, V. M. Acosta, and R. G. Beausoleil, "Coupling of nitrogen-vacancy centers to photonic crystal cavities in monocrystalline diamond," *Phys. Rev. Lett.*, vol. 109, p. 033604, Jul 2012. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.109.033604

[21] T. Schröder, S. Mouradian, J. Zheng, M. E. Trusheim, M. Walsh, E. H. Chen, L. Li, I. Bayn, and D. Englund, "Review article: Quantum nanophotonics in diamond," 2016.

[22] A. Sipahigil, R. E. Evans, D. D. Sukachev, M. J. Burek, J. Borregaard, M. K. Bhaskar, C. T. Nguyen, J. L. Pacheco, H. A. Atikian, C. Meuwly, R. M. Camacho, F. Jelezko, E. Bielejec, H. Park, M. Lončar, and M. D. Lukin, "An integrated diamond nanophotonics platform for quantum-optical networks," *Science*, vol. 354, no. 6314, pp. 847–850, Nov. 2016, publisher: American Association for the Advancement of Science. [Online]. Available: https://www.science.org/doi/10.1126/science.aah6875

[23] M. Atatüre, D. Englund, N. Vamivakas, S.-Y. Lee, and J. Wrachtrup, "Material platforms for spin-based photonic quantum technologies," *Nature Reviews Materials*, vol. 3, pp. 38–51, 2018.

[24] C. Berger, A. Di Paolo, T. Forrest, S. Hadfield, N. Sawaya, M. Stęchły, and K. Thibault, "Quantum technologies for climate change: Preliminary assessment," Jun. 2021, arXiv:2107.05362 [quant-ph]. [Online]. Available: http://arxiv.org/abs/2107.05362

[25] G. Keiser, *Optical Fiber Communications*, ser. McGraw-Hill Series in Electrical and Computer Engineering. McGraw-Hill, 2000. [Online]. Available: https://books.google.ie/books?id=lANTAAAAMAAJ

[26] W. Dür and H. J. Briegel, "Entanglement purification and quantum error correction," *Reports on Progress in Physics*, vol. 70, no. 8, pp. 1381–1424, Aug. 2007, arXiv:0705.4165 [quant-ph]. [Online]. Available: http://arxiv.org/abs/0705.4165

[27] J. Y. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. 712–716, 1971. [Online]. Available: http://www.jstor.org/stable/2629312

[28] B. A. Myers, "Quantum decoherence of near-surface nitrogen-vacancy centers in diamond and implications for nanoscale imaging," Ph.D. dissertation, UC Santa Barbara, 2016. [Online]. Available: https://escholarship.org/uc/item/8bj3x7nr

[29] T. Coopmans, R. Knegjens, A. Dahlberg, D. Maier, L. Nijsten, J. de Oliveira Filho, M. Papendrecht, J. Rabbie, F. Rozpędek, M. Skrzypczyk, and et al., "Netsquid, a network simulator for quantum information using discrete events," *Communications Physics*, vol. 4, no. 1, 2021.

[30] A. Einstein, "The foundation of the general theory of relativity," 1916.

[31] F. Toudeh-Fallah, M. Pistoia, Y. Kawakura, N. Moazzami, D. H. Kramer, R. I. Woodward, G. Sysak, B. John, O. Amer, A. O. Polychroniadou, J. Lyon, S. Shetty, T. D. Movva, S. Upadhyay, M. R. Behera, J. A. Dolphin, P. A. Haigh, J. F. Dynes, and A. J. Shields, "Paving the way towards 800 gbps quantum-secured optical channel deployment in mission-critical environments," 2022. [Online]. Available: https://arxiv.org/abs/2202.07764

[32] S.-H. Wei, B. Jing, X.-Y. Zhang, J.-Y. Liao, C.-Z. Yuan, B.-Y. Fan, C. Lyu, D.-L. Zhou, Y. Wang, G.-W. Deng, H.-Z. Song, D. Oblak, G.-C. Guo, and Q. Zhou, "Towards real-world quantum networks: a review," *Laser & Photonics Reviews*, vol. 16, no. 3, p. 2100219, Mar. 2022, arXiv:2201.04802 [quant-ph]. [Online]. Available: http://arxiv.org/abs/2201.04802

[33] J. Ghosh, A. G. Fowler, and M. R. Geller, "Surface code with decoherence: An analysis of three superconducting architectures," *Physical Review A*, vol. 86, no. 6, p. 062318, Dec. 2012, publisher: American Physical Society. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.86.062318

[34] M. Schlosshauer, "Quantum Decoherence," *Physics Reports*, vol. 831, pp. 1–57, Oct. 2019, arXiv:1911.06282 [quant-ph]. [Online]. Available: http://arxiv.org/abs/1911.06282

[35] M. D. Feuer, D. C. Kilper, and S. L. Woodward, "ROADMs and their system applications," in *Optical Fiber Telecommunications VB*. Elsevier, 2008, ch. 8.

[36] R. Wang, R. S. Tessinari, E. Hugues-Salas, A. Bravalheri, N. Uniyal, A. S. Muqaddas, R. S. Guimaraes, T. Diallo, S. Moazzeni, Q. Wang, G. T. Kanellos, R. Nejabati, and D. Simeonidou, "End-to-End Quantum Secured Inter-Domain 5G Service Orchestration Over Dynamically Switched Flex-Grid Optical Networks Enabled by a q-ROADM," *Journal of Lightwave Technology*, vol. 38, no. 1, pp. 139–149, Jan. 2020, conference Name: Journal of Lightwave Technology.

[37] I. McGregor, "The relationship between simulation and emulation," in *Proceedings of the Winter Simulation Conference*, vol. 2, Dec. 2002, pp. 1683–1688 vol.2.

[38] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," in *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing*, India, 1984, p. 175.

[39] C. Elliott, A. Colvin, D. Pearson, O. Pikalo, J. Schlafer, and H. Yeh, "Current status of the DARPA Quantum Network," Mar. 2005, arXiv:quant-ph/0503058. [Online]. Available: http://arxiv.org/abs/quant-ph/0503058

[40] F. Toudeh-Fallah, M. Pistoia, Y. Kawakura, N. Moazzami, D. H. Kramer, R. I. Woodward, G. Sysak, B. John, O. Amer, A. O. Polychroniadou, J. Lyon, S. Shetty, T. D. Movva, S. Upadhyay, M. R. Behera, J. A. Dolphin, P. A. Haigh, J. F. Dynes, and A. J. Shields, "Paving the Way towards 800 Gbps Quantum-Secured Optical Channel Deployment in Mission-Critical Environments," Feb. 2022, arXiv:2202.07764 [physics, physics:quant-ph]. [Online]. Available: http://arxiv.org/abs/2202.07764

[41] J. Yin, Y.-H. Li, S.-K. Liao, M. Yang, Y. Cao, L. Zhang, J.-G. Ren, W.-Q. Cai, W.-Y. Liu, S.-L. Li, R. Shu, Y.-M. Huang, L. Deng, L. Li, Q. Zhang, N.-L. Liu, Y.-A. Chen, C.-Y. Lu, X.-B. Wang, F. Xu, J.-Y. Wang, C.-Z. Peng, A. K. Ekert, and J.-W. Pan, "Entanglement-based secure quantum cryptography over 1,120 kilometres," *Nature*, vol. 582, no. 7813, pp. 501–505, Jun. 2020, number: 7813 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41586-020-2401-y

[42] S. Pirandola, R. Laurenza, C. Ottaviani, and L. Banchi, "Fundamental limits of repeaterless quantum communications," *Nature Communications*, vol. 8, no. 1, p. 15043, Apr. 2017, number: 1 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/ncomms15043

[43] J. Hofmann, M. Krug, N. Ortegel, L. Gérard, M. Weber, W. Rosenfeld, and H. Weinfurter, "Heralded entanglement between widely separated atoms," *Science*, vol. 337, no. 6090, pp. 72–75, 2012. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.1221856

[44] M. J. Burek, Y. Chu, M. S. Z. Liddy, P. Patel, J. Rochman, S. Meesala, W. Hong, Q. Quan, M. D. Lukin, and M. Lončar, "High quality-factor optical nanocavities in bulk single-crystal diamond," *Nature Communications*, vol. 5, no. 1, p. 5718, Dec. 2014, number: 1 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/ncomms6718

[45] B. Hensen, H. Bernien, A. E. Dréau, A. Reiserer, N. Kalb, M. S. Blok, J. Ruitenberg, R. F. L. Vermeulen, R. N. Schouten, C. Abellán, W. Amaya, V. Pruneri, M. W. Mitchell, M. Markham, D. J. Twitchen, D. Elkouss, S. Wehner, T. H. Taminiau, and R. Hanson, "Experimental loophole-free violation of a Bell inequality using entangled electron spins separated by 1.3 km," Oct. 2015, arXiv:1508.05949 [quant-ph]. [Online]. Available: http://arxiv.org/abs/1508.05949

[46] G. Balasubramanian, P. Neumann, D. Twitchen, M. Markham, R. Kolesov, N. Mizuochi, J. Isoya, J. Achard, J. Beck, J. Tissler, V. Jacques, P. R. Hemmer, F. Jelezko, and J. Wrachtrup, "Ultralong spin coherence time in isotopically engineered diamond," *Nature Materials*, vol. 8, no. 5, pp. 383–387, May 2009, number: 5 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/nmat2420

[47] J. Cramer, N. Kalb, M. A. Rol, B. Hensen, M. S. Blok, M. Markham, D. J. Twitchen, R. Hanson, and T. H. Taminiau, "Repeated quantum error correction on a continuously encoded qubit by real-time feedback," *Nature Communications*, vol. 7, no. 1, p. 11526, May 2016, arXiv:1508.01388 [cond-mat, physics:quant-ph]. [Online]. Available: http://arxiv.org/abs/1508.01388

[48] Z.-H. Wang, G. de Lange, D. Ristè, R. Hanson, and V. V. Dobrovitski, "Comparison of dynamical decoupling protocols for a nitrogen-vacancy center in diamond," *Phys. Rev. B*, vol. 85, p. 155204, Apr 2012. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.85.155204

[49] D. Sukachev, A. Sipahigil, C. Nguyen, M. Bhaskar, R. Evans, F. Jelezko, and M. Lukin, "Silicon-Vacancy Spin Qubit in Diamond: A Quantum Memory Exceeding 10 ms with Single-Shot State Readout," *Physical Review Letters*, vol. 119, no. 22, p. 223602, Nov. 2017, publisher: American Physical Society. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.119.223602

[50] A. Faraon, C. Santori, Z. Huang, V. M. Acosta, and R. G. Beausoleil, "Coupling of Nitrogen-Vacancy Centers to Photonic Crystal Cavities in Monocrystalline Diamond," *Physical Review Letters*, vol. 109, no. 3, p. 033604, Jul. 2012, publisher: American Physical Society. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.109.033604

[51] P. C. Maurer, G. Kucsko, C. Latta, L. Jiang, N. Y. Yao, S. D. Bennett, F. Pastawski, D. Hunger, N. Chisholm, M. Markham, D. J. Twitchen, J. I. Cirac, and M. D. Lukin, "Room-Temperature Quantum Bit Memory Exceeding One Second," *Science*, vol. 336, no. 6086, pp. 1283–1286, Jun. 2012, publisher: American Association for the Advancement of Science. [Online]. Available: https://www.science.org/doi/10.1126/science.1220513

[52] M. E. Trusheim, L. Li, A. Laraoui, E. H. Chen, H. Bakhru, T. Schröder, O. Gaathon, C. A. Meriles, and D. Englund, "Scalable fabrication of high purity diamond nanocrystals with long-spin-coherence nitrogen vacancy centers," *Nano Letters*, vol. 14, no. 1, pp. 32–36, 2014, pMID: 24199716. [Online]. Available: https://doi.org/10.1021/nl402799u

[53] C. P. Slichter and C. P. Slichter, *Principles of magnetic resonance / C.P. Slichter*, 3rd ed. Springer-Verlag Berlin ; New York, 1990.

[54] A. Dahlberg and S. Wehner, "SimulaQron - A simulator for developing quantum internet software," *Quantum Science and Technology*, vol. 4, no. 1, p. 015001, Sep. 2018, arXiv:1712.08032 [quant-ph]. [Online]. Available: http://arxiv.org/abs/1712.08032

[55] S. Diadamo, J. Nötzel, B. Zanger, and M. M. Beşe, "QuNetSim: A Software Framework for Quantum Networks," *IEEE Transactions on Quantum Engineering*, vol. 2, pp. 1–12, 2021, conference Name: IEEE Transactions on Quantum Engineering.

[56] N. Killoran, J. Izaac, N. Quesada, V. Bergholm, M. Amy, and C. Weedbrook, "Strawberry Fields: A Software Platform for Photonic Quantum Computing," *Quantum*, vol. 3, p. 129, Mar. 2019, arXiv:1804.03159 [physics, physics:quant-ph]. [Online]. Available: http://arxiv.org/abs/1804.03159

[57] T. Coopmans, R. Knegjens, A. Dahlberg, D. Maier, L. Nijsten, J. d. O. Filho, M. Papendrecht, J. Rabbie, F. Rozpędek, M. Skrzypczyk, L. Wubben, W. de Jong, D. Podareanu, A. Torres-Knoop, D. Elkouss, and S. Wehner, "NetSquid, a NETwork Simulator for QUantum Information using Discrete events," *Communications Physics*, vol. 4, no. 1, p. 164, Dec. 2021, arXiv:2010.12535 [quant-ph]. [Online]. Available: http://arxiv.org/abs/2010.12535

[58] C. Antonelli, M. Shtaif, and M. Brodsky, "Distance limitations on the entanglement distribution over optical fiber due to chromatic and polarization mode dispersion," May 2011.

[59] L. Gyongyosi and S. Imre, "Opportunistic Entanglement Distribution for the Quantum Internet," *Scientific Reports*, vol. 9, no. 1, p. 2219, Feb. 2019, number: 1 Publisher:

Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41598-019-38495-w

[60] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," *Theoretical Computer Science*, vol. 560, pp. 7–11, Dec. 2014, arXiv:2003.06557 [quant-ph]. [Online]. Available: http://arxiv.org/abs/2003.06557

[61] D. Deutsch, A. Ekert, R. Jozsa, C. Macchiavello, S. Popescu, and A. Sanpera, "Quantum privacy amplification and the security of quantum cryptography over noisy channels," *Physical Review Letters*, vol. 77, no. 13, pp. 2818–2821, Sep. 1996, arXiv:quant-ph/9604039. [Online]. Available: http://arxiv.org/abs/quant-ph/9604039

[62] G. Iñesta, G. Vardoyan, L. Scavuzzo, and S. Wehner, "Optimal entanglement distribution policies in homogeneous repeater chains with cutoffs," Aug. 2022, arXiv:2207.06533 [quant-ph]. [Online]. Available: http://arxiv.org/abs/2207.06533

[63] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[64] F. A. Mele, L. Lami, and V. Giovannetti, "Restoring quantum communication efficiency over high loss optical fibres," *Physical Review Letters*, vol. 129, no. 18, p. 180501, Oct. 2022, arXiv:2204.13128 [quant-ph]. [Online]. Available: http://arxiv.org/abs/2204.13128

[65] S. R. Bickham, P. G. Hebgen, P. Tandon, H. B. Matthews, J. T. Prater, D. A. Lewis, H. M. De Pedro, D. A. Stainer, I. I. Kouzmina, S. L. Wager, J. R. Pace, B. W. Wakefield, J. D. Downie, and J. E. Hurley, "Ultra-Low-Loss Reduced-Clad Fiber with Attenuation less than 0.16 dB/km," in *2022 27th OptoElectronics and Communications Conference (OECC) and 2022 International Conference on Photonics in Switching and Computing (PSC)*, Jul. 2022, pp. 1–3.

[66] N. Bar-Gill, L. M. Pham, A. Jarmola, D. Budker, and R. L. Walsworth, "Solid-state electronic spin coherence time approaching one second," *Nature Communications*, vol. 4, no. 1, p. 1743, Apr. 2013, number: 1 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/ncomms2771

[67] E. D. Herbschleb, H. Kato, Y. Maruyama, T. Danjo, T. Makino, S. Yamasaki, I. Ohki, K. Hayashi, H. Morishita, M. Fujiwara, and N. Mizuochi, "Ultra-long coherence times amongst room-temperature solid-state spins," *Nature Communications*, vol. 10, no. 1, p. 3766, Aug. 2019, number: 1 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41467-019-11776-8

[68]

[69] V. Giovannetti and R. Fazio, "Information-capacity description of spin-chain correlations," *Physical Review A*, vol. 71, no. 3, mar 2005. [Online]. Available: https://doi.org/10.1103%2Fphysreva.71.032314

[70] J. Preskill, "Lecture notes for ph219/cs219: Quantum information, chapter 3," October 2018.

[71] N. Kalb, P. C. Humphreys, J. J. Slim, and R. Hanson, "Dephasing mechanisms of diamond-based nuclear-spin memories for quantum networks," *Physical Review A*, vol. 97, no. 6, p. 062330, Jun. 2018. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.97.062330

[72] K. Suzuki, O. Moriwaki, K. Hadama, K. Yamaguchi, H. Taniguchi, Y. Kisaka, D. Ogawa, M. Takeshita, S. Camatel, Y. Ma, and M. Fukutoku, "A transponder aggregator with efficient use of filtering function for transponder noise suppression," 2022.

[73] P. Erdös and A. Rényi.

[74] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[75] A. Barrat and M. Weigt, "On the properties of small-world network models," 1999.

[76] D. A. Hopper, J. D. Lauigan, T.-Y. Huang, and L. C. Bassett, "Real-Time Charge Initialization of Diamond Nitrogen-Vacancy Centers for Enhanced Spin Readout," *Physical Review Applied*, vol. 13, no. 2, p. 024016, Feb. 2020, arXiv:1907.08741 [quant-ph]. [Online]. Available: http://arxiv.org/abs/1907.08741

[77] J. E. Martinez, P. Fuentes, P. M. Crespo, and J. Garcia-Frias, "Approximating decoherence processes for the design and simulation of quantum error correction codes on classical computers," *IEEE Access*, vol. 8, pp. 172 623–172 643, 2020.