Quantum Network Simulation

Project Report – SIP 2023

Submitted by:
Shivam Sawarn
Delhi University

28/07/2023

Supervisor:
Prof. R. P. Singh
AMOPH Division
Physical Research Laboratory
Ahmedabad, Gujarat - 380059

Contents

1 Introduction

Classical networks, the backbone of today's communication infrastructure, use encryption algorithms to secure information transmission. These algorithms transform the original message or data into an incomprehensible format, thus safeguarding it from unauthorized access. However, these encryption methods could become vulnerable in the face of fault-tolerant quantum computers. Once these quantum computers become feasible, they could potentially break these classical encryption codes, threatening the privacy and safety of transmitted information.

As a response to this emerging vulnerability, researchers have introduced the concept of Quantum Key Distribution (QKD). This quantum-based approach deviates significantly from traditional encryption methods. In QKD, two communicating parties can generate a shared key that, in theory, is invulnerable to any form of unauthorized interception. This shared key is then used for encoding and decoding the transmitted information, providing a sturdy shield against potential eavesdroppers.

Quantum Key Distribution forms the foundation of quantum networks. A quantum network is composed of multiple nodes that can share encryption keys amongst themselves using the principles of QKD, while still maintaining the overall system security. The underlying advantage of these quantum networks is the ability to provide enhanced data transmission security. It is expected that with the tactical use of quantum networks, communication systems would become significantly more secure and resistant to potential threats.

In this summer research project, we work on Simulator of QUantum Network Communication (SeQUeNCe), a customizable discrete-event quantum network simulator. This tool models the building blocks of quantum hardware and network protocols, allowing us to experiment with them. Our goal here is to demonstrate the importance of quantum network simulations. They can help build scalable quantum networks, fine-tune quantum devices, and boost network performance. The idea is not to replace the networks we have now, but to establish an extra security layer.

Throughout this project, we aim to contribute to these challenging goals and acquire deep insights into quantum networking. Moreover, we have used real experimental data to illustrate the BB84 protocol, which is a key component in quantum communication.

The project is divided into four section. Section 2 introduces the concept of quantum networks and explores their novel capabilities and potential applications, as well as the inherent challenges associated with establishing these networks. Section 3 discusses the significance of simulation in quantum networking and outlines key requirements and choices for a quantum network simulator. Section 4 delves into the inner workings of SeQUeNCe, detailing its modular design, key protocols, and its implementation. Section 5 presents the simulation results and in-depth analysis, examining how throughput, error rates, and latency varied for both short and long-range distances. The project concludes with summarizing key findings and future directions for quantum network simulation.

2 Quantum Networks

Quantum networks are collection of nodes that is able to exchange Qubits and distribute entangled states amongst themselves. It represents an advanced realm of quantum information science that exploit quantum mechanics principles to produce ultra-secure, large-capacity communication systems. Unlike traditional networks which employ classical bits for information transfer, quantum networks utilise entangled pairs of qubits.

The method of information transmission also differs significantly between quantum networks and classical networks. Classical networks transfer data via an established connection that depends on network bandwidth. Quantum networks, meanwhile, leverage the principle of quantum entanglement for instantaneous information exchange. This phenomenon also yields significant benefits in terms of security, as the generation of shared cryptographic keys in quantum networks do not rely on algorithmic complexity. This contrasts with traditional cryptographic methods and can provide robust, long-term protection against advancements in computing hardware.

In addition to ultra-secure communication systems, quantum networks present novel and transformative applications. These applications extend beyond the capabilities of classical networks, with potential uses in quantum computing, next-generation secure communication, quantum teleportation, and more. The implications of this technology are vast and varied, extending across different sectors and industries. As such, users can expect significantly increased data-carrying capacities and ultra-secure communication systems, providing a significant edge over the capabilities of classical networks.

Challenges

Building quantum networks isn't as straightforward as simply turning classical models into their quantum equivalents. Quantum networks face several technological and fundamental challenges that make classical approaches unsuitable in a quantum context.

1. **The Problem with Measurement:** In classical computers and networks, data bits stored in memory can be read at any time. This ability is useful for copying data, detecting errors, and more. However, qubits, which are the backbone of quantum networks, can't be read out the same way. The state of a qubit is destroyed once it's measured, and it loses any superposition or entanglement it had. Due to this, quantum networks cannot use classical methods for error detection and correction.

2. **The No-Cloning Theorem:** Quantum mechanics includes a no-cloning theorem, which means we cannot create an identical copy of an arbitrary, unknown quantum state. This makes it impossible to use classical methods for signal amplification, retransmission, and more. All these methods rely on the ability to copy data, which is not possible in quantum mechanics. As such, connecting nodes within a quantum network, where the physical channel is always lossy, is a challenging task.

3. **Fidelity:** Classical data packets are expected to arrive at their destination without any errors introduced by hardware noise. However, error detection and correction in quantum networks are more complicated due to our inability to read or copy a quantum state. A physical quantity called "fidelity" measures the quality of a quantum state on a scale of 0 to 1. High fidelity indicates that a state is close to the one we tried to create. Although quantum applications don't need perfect fidelity to operate, how a network manages errors will significantly impact its architecture.

4. **The Inadequacy of Direct Transmission** Sending entangled quantum states directly from one end to another across multiple nodes while performing sufficient forward quantum error correction to bring down losses to an acceptable level is not feasible. Quantum networks rely on different mechanisms, such as entanglement swapping and teleportation. These networks fundamentally differ from classical networks, requiring different management techniques and protocol designs. The concept of "store and forward" and "store and swap" in quantum networks also differ. Quantum networks perform a "store and swap" operation where Bell pairs are stored and then an entanglement swapping operation is performed, instead of forwarding in the data plane. These distinctions make control algorithms and optimization of quantum networks different from classical ones.

Overall, creating a quantum network embodies a significant challenge with its unique nature and requirements due to the distinct and delicate properties of quantum mechanics.

3 Simulation

With the unique and complex nature of quantum networks, the need for a dedicated simulator that can accurately model these networks is crucial for their development and evolution. A quantum network simulator could assist researchers and developers in designing and testing new quantum network architectures and protocols, without the need for physical implementation. It would provide insights into the behaviors of quantum networks under various scenarios, and possible area of optimization.

3.1 Requirements

To maximally aid quantum network research, a quantum network simulator must meet several key requirements. These essential characteristics enable realistic, robust, and insightful simulations, assisting in the development of efficient and high-performing quantum networks. They are enumerated as follows:

(a) **Accuracy and Realism:** The simulator should generate results that can be validated and recreate real-world scenarios as closely as possible, including faithfully capturing the behaviors of quantum states. Precise tracking of timing and execution of events, down to picosecond precision, is also crucial given the time-sensitivity of quantum networks.

(b) **Extensibility and Flexibility:** The chosen simulator should allow for seamless upgrades, easy implementation of new models in QKD and similar technologies, and flexibility in simulating alternative network architectures and new protocols. It should have a modular design to facilitate reprogramming of protocols, and quick testing of different scenarios through parameter alterations. This is required as there are no standard protocols defined by *IETF* for quantum networks.

(c) **Usability:** The simulator should be user-friendly and able to integrate with existing solutions. Comprehensive software support including active developer cooperation, regular updates, error reporting, and clear instructions are essential in evaluating a simulator's usability.

(d) **Availability:** Preference should be given to open-source simulators that are freely available for extensions to foster integration and testing of the QKD technology with varied existing network solutions at minimal cost.

(e) **Scalability:** The simulator should be capable of performing large-scale studies on wide area networks with numerous components. It should effectively handle high-frequency photon tracking and manage sizable simulation events. Features like high-performance computing systems compatibility and efficient parallel simulation methods are desirable for enhanced scalability.

3.2 Available Simulators

I had the opportunity to explore various quantum network simulators, understand their workings, and evaluate them in terms of flexibility, effectiveness, and scope. In this process, I became familiar with options like QKDNetSim, QuISP, NetSquid, QuNetSim, SQUANCH, SeQUeNCe and SimulaQron. The details of each simulator tried are as follows:

(i) **QKDNetSim:** It is specifically designed for QKD networks based on network simulator of version 3 (NS-3) with a support of web interface[1]. It can operate both in an overlay mode or as a network with a single TCP/IP stack, providing flexibility for different network setups. QKDNetSim is mainly used to test existing solutions and implement new ones in the field of QKD networks. It proves useful in prioritizing traffic originated from the routing protocol in the allocation of network resources, thus avoiding bringing QKD links into a locked position. Its virtual TCP/IP stack allows for easy simulation of overlay networks, and its inbuilt QKD buffers simplify simulations involving symmetrical cryptographic keys. Despite its numerous benefits, it has base files which depends on outdated versions of python, possibly limiting its scope.

(ii) **QuISP:** QuISP aims to comprehensively simulate full-stack quantum networks. It supports a variety of protocols such as entanglement purification, link tomography,

[1] https://open-qkd.eu

and entanglement swapping. The main emphasis of QuISP is scalability, with a goal to simulate a quantum internet of up to 100 networks with 100 nodes each. However, its main drawback keeping in mind our aim is its prime focus on scalibility rather than accuracy to a small functional network.

(iii) **NetSquid:** It is a software available upon request[2] that focuses on simulation using Nitrogen-Vacancy (NV) centres in diamond and ensemble memories. NetSquid offers a modular framework to allow modeling a range of architectures. It also proposes a five-layer quantum network stack including physical, link, network, transmission, and application layers. However, its main focus remains on physical and link layer protocols.

(iv) **QuNetSim:** QuNetSim can handle smaller-scale simulations of 5-10 nodes, employing a network layering framework inspired by the OSI(Open System Interconnection) model. While it successfully simulates entanglement swapping, it lacks support for entanglement purification, and does not specify physical layer protocols, making it less comprehensive for wide-ranging projects.

(v) **SQUANCH:** The unique feature of SQUANCH is its agent-based modeling that allows a natural form of parallelization by running each agent on a different process. However, it does not model detailed interactions of protocols or physical-layer descriptions, limiting its utility for comprehensive simulations.

(vi) **SimulaQron:** Specifically designed to facilitate the development of network applications, SimulaQron does not model time-dependent noise and uses a classical-quantum combined interface with sockets mimicking information transmission over simulated quantum channels. Its main limitation is a lack of comprehensive noise modeling.

(vii) **SeQUeNCe:** SeQUeNCe stands out with its comprehensive modularized design that allows for great simulation flexibility. The simulator focuses on single rare-earth ion memories while providing the possibility to study other technologies. The key strength of SeQUeNCe is its flexible approach to quantum network simulations and accommodation of comprehensive technological and processing variations. It is open-source framework with support of Graphical User Interface(GUI) locally.

4 Simulator of Quantum Network Communication (SeQUeNCe)

The SeQUeNCe simulator stands out for its modular design, based on the main principles identified from many types of quantum network designs we've researched. The design is specifically constructed for quickly adapting to changing architecture principles and

[2] https://netsquid.org

simulating a wide range of possible future quantum network architectures. Figure 1 shows the modularized design of SeQUeNCe.

The central aspect of SeQUeNCe is the **Simulation Kernel**, which has an event scheduler for managing discrete-event simulation and a quantum manager for keeping watch on quantum states throughout the simulation process. This two-component system provides users with an exceptional level of control over when events are executed, which is key for maintaining accurate timelines and expanding the system's capacity.

Hardware Module includes models of essential hardware components involved in a quantum network, while the **Entanglement Management Module** handles protocols for reliable and high-fidelity distribution of entangled qubit pairs between network nodes.

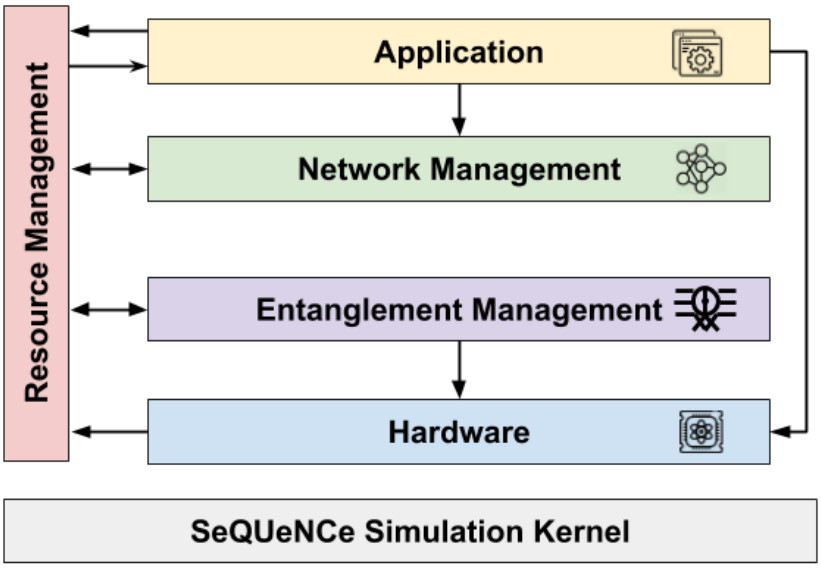

Figure 1: Modularized design of SeQUeNCe (*Credits: SeQUeNCe*)

Resource Management Module takes care of managing local resources within one network node. It records the state of hardware, allocates resources to applications, and regains control of the hardware when the resources are released. Shifted slightly from the primary dynamics of the other modules, the Network Management Module extends its operability across the local applications and remote network managers and liaises with the resource manager for optimal utilization of local resources.

Application Module, finally, represents a varied spectrum of quantum network applications, all of which rely fundamentally on entanglement. The module is capable of triggering distribution of entanglement between itself and another network node based on specific requisites.

A key advantage of SeQUeNCe is its high degree of reconfigurability, offered through the user's ability to specify network topology and a variety of network and protocol parameters. The modular design allows advanced users to introduce their models of new quantum hardware and network protocols. This degree of flexibility and adaptability was a critical determinant in SeQUeNCe being our simulator of choice for quantum networks.

4.1 Design of modules

The modular design of SeQUeNCe encompasses numerous distinct components, each playing a vital role in the simulation of quantum networks. At the center of the operations is the *Simulation Kernel*, managing the execution of events and tracking of quantum states. The *Hardware Module* integrates models of fundamental hardware components, creating an empirical environment for the simulation. Supporting these core modules, the *Entanglement Management* and *Resource Management Modules* oversee the protocols for entangled qubit distribution and the effective allocation of local resources. Lastly, the *Network Management* and *Application Modules* deal with quantum network services and represent a variety of quantum network applications. Together these modules create an extensive portfolio for accurate and scalable quantum network simulations which is discussed below in details.

4.1.1 Hardware

Integral to SeQUeNCe's design is the Hardware Module. This module encapsulates various smaller yet essential components that cumulatively contribute to a detailed and faithful simulation of quantum network dynamics. By simulating the vital facets of a quantum network setup, the Hardware module enables users to have practical and relatable insights. The description of how we model the key hardware elements in SeQUeNCe are as follows:

- **Quantum and classical channels:** Quantum channels use standard communication fiber/free-space to transmit quantum data while classical channels transmit classical information. The quantum channel has two main functions - scheduling transmission times and transmitting single photons. Propagation delay is modeled using the length of the fiber and the speed of light. Our simulator synchronizes all photon sources sharing a channel in order to ensure proper spacing of photons from different sources.

- **Single photon detectors (SPD):** These are used to detect individual photon arrivals by generating an electrical signal and recording the arrival time. The detector's efficiency determines the probability of a successful detection. The detector resolution determines the precision of time stamps. The SPD model also takes into account 'dead time' and the average number of false positives.

- **Quantum memories:** These are used to store quantum information in the form of matter-based qubits. The quantum memory model in this study is based on single atom memories (*Erbium ions in solids*). The quantum state and the efficiency of memory decides the probability of photon emission. Along with storing quantum states, the quantum memory also maintains the fidelity of entanglement. The model considers physical parameters such as photon collection efficiency, atom-cavity cooperativity, optical decay rate, optical pure dephasing rate, and difference between the optical transition frequencies in quantum memory simulations.

- **Nodes:** The model follows the outline from a recent *IETF* draft. It refers to two types of network nodes - router nodes and Bell State Measurement (BSM) nodes. Router node has all the functionality of a *quantum repeater* to overcome photon loss. BSM nodes are crucial for the Barrett-Kok entanglement generation protocol.

4.1.2 Entanglement Management

The entanglement management module includes models of protocols for undertakings such as entanglement generation, purification, and swapping. It is vital for the functioning of the quantum network as it interfaces with both resource and hardware management modules for smooth operation.

- **Barrett–Kok Entanglement Generation Protocol:** This protocol is used as it can tolerate significant hardware errors such as detector loss and spontaneous emission making it more resilient. It uses matter qubits and linear optics and prepares quantum memories in a specific state that ensures entanglement between two qubits. This protocol has two rounds; the first round generates entanglement between two qubits while the second round improves the entangled state.

- **BBPSSW Purification Protocol:** The purification protocol is used to enhance the quality of entangled qubits which may have been affected by factors such as imperfections and decoherence in quantum memories. It utilizes various pairs of low-quality entangled qubits to produce a pair of high-quality entangled qubits. This process involves the use of quantum gates for the purification process.

- **Swapping Protocol:** The swapping protocol is used to transform multiple short-distance entanglements into single long-distance entanglement. This transformation is achieved by performing a Bell State Measurement (BSM) at the intermediate node, resulting in the entanglement of memories at either end. When successfully implemented, the success probability and degradation of swapping are defined by the operations performed at the intermediate node.

All these protocols are powered by existing algorithms and methods, leveraging state machines and exchanging classical messages to operate. Importantly, the entanglement management module manages all the resources well when notified of entanglement expiration, freeing up the resources used by terminated protocols.

4.1.3 Resource Management

The resource manager module is essential for managing local resources in a quantum network and operates by using an internal set of rules for managing quantum memories.

Three interfaces provide crucial points of interaction:

1. **Interface 1** communicates with the network management module and application module. This interface retrieves the state of local resources and installs rules, which

instruct the resource manager on how to allocate hardware resources to various entanglement protocols.

2. **Interface 2** exchanges with the hardware and entanglement management modules to update the hardware state and engage or release resources. This ensures the resource manager has an accurate count of resource locations, their internal states, and fidelity.

3. **Interface 3** communicates with resource managers on other nodes. It aids the control of entanglement protocols and ensures a smooth operation by sending relevant messages to create or terminate corresponding protocol instances.

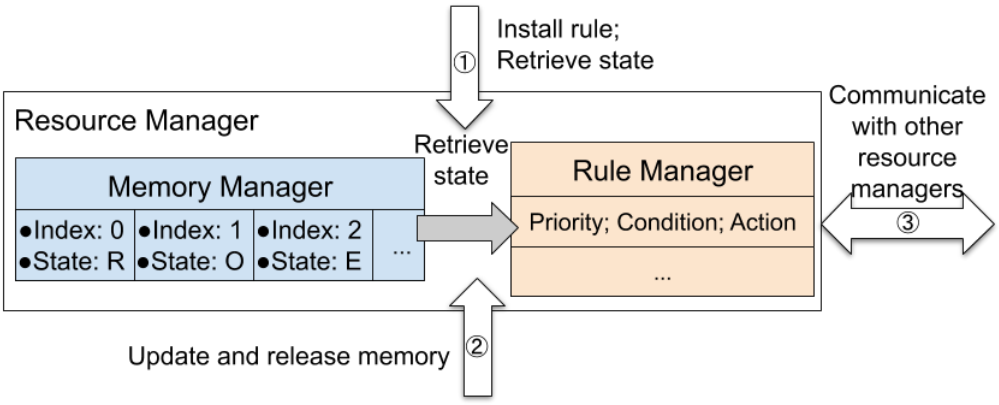

Figure 2: Resource Management Module(*Credits: SeQUeNCe*)

Figure 2 shows the resource management module of the simulator. The internal structure of the resource manager features a memory manager that traces the states of quantum memories. Quantum memories can be *raw* (not entangled and not in use), *entangled (entang*led to another memory or memories, with corresponding entanglement information also recorded), or *occupied* (allocated to an entanglement protocol).

The rule manager, another component of the resource manager, uses a rule set to manage hardware resources. These rules are sorted by priority and consist of three parts: *priority*, *condition*, and *action*. Rules help in decision-making for hardware resource allocation based on the priorities set.

4.1.4 Network Management

The network management module in a quantum network is crucial for applications to reserve network resources, employing a reservation-based approach. It assumes two primary responsibilities - reservation and routing.

1. **Reservation:** This process is responsible for reserving local resources. The reservation phases require an application to create a reservation request, which includes the initiator (the one sending the reservation request), the responder (the one receiving the request), desired fidelity, memory size, start time, and end time. If resources are

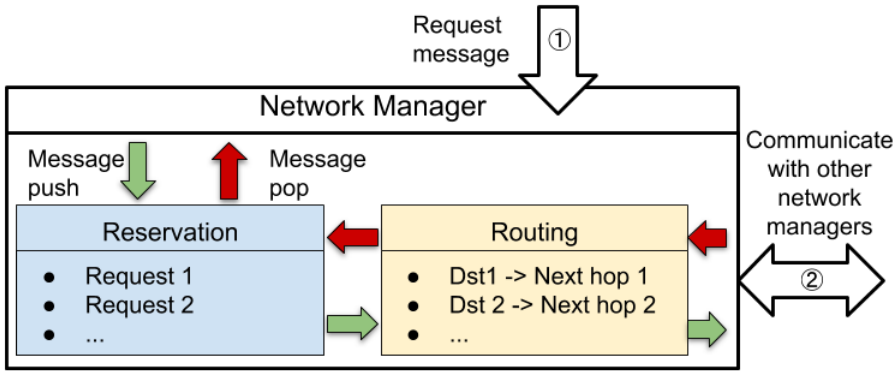

Figure 3: Network Management Module (*Credits: SeQUeNCe*)

available, the network manager pushes the request to a reservation instance, which reserves these resources.

2. **Routing:** This provides a path for entanglement distribution to fulfill the end-to-end reservation. When the reservation instance approves a request, it assigns local information and pushes it back to the routing instance to send the request to the next hop. Routing involves determining the next hop based on a forwarding table and calculating the shortest path using the length of quantum channels.

Figure 3 shows the Network Management Module of the simulator. The network management module operates in coordination with network managers on other nodes and has interfaces for communication with applications and neighboring nodes. It also has the capability to accommodate an additional authentication protocol to augment the network's security. Overall, It streamlines the process of resource reservation and routing, providing a seamlessly functioning quantum network.

4.1.5 Applications

The application module within a quantum network essentially consumes entanglement. The applications within a quantum network can vary greatly, each with unique requirements for aspects like throughput and fidelity. Along with this abstract application model, the SeQUeNCe simulator also includes a reference implementation of quantum key distribution. It consumes and operates on entangled states as needed, serving as an end-user entity within the quantum network architecture. Therefore, studying its operation, even with this abstract model, is crucial to understanding overall network dynamics.

4.2 BB84 Protocol

The BB84 protocol is a cornerstone in the field of quantum cryptography, enabling two remote parties to exchange encryption keys securely. Initiated by scientists *Charles Bennett* and *Giles Brassard* in 1984, this quantum key distribution (QKD) protocol harnesses elements of quantum mechanics to allow for protected communication over unsecured channels.

The BB84 protocol operates on a principle that involves the usage of two orthogonal quantum states as bases. The process commences when the sender, often referred to as Alice, sends a sequence of quantum bits or qubits to the receiver, known as Bob. Each qubit is prepared in one of the two different measurement bases, either rectilinear or diagonal, chosen at random. Upon receipt, Bob randomly selects a basis to measure each received qubit. It's important to note that the selection of the measurement base and qubit state is completely random, enhancing the protocol's security.

If Bob's chosen basis coincides with Alice's, an accurate qubit measurement occurs. By comparing their chosen bases without revealing the actual measured qubits, Alice and Bob can discard instances where they used differing bases. Post this 'sifting' process, they can establish a shared secret key for encrypting their communication.

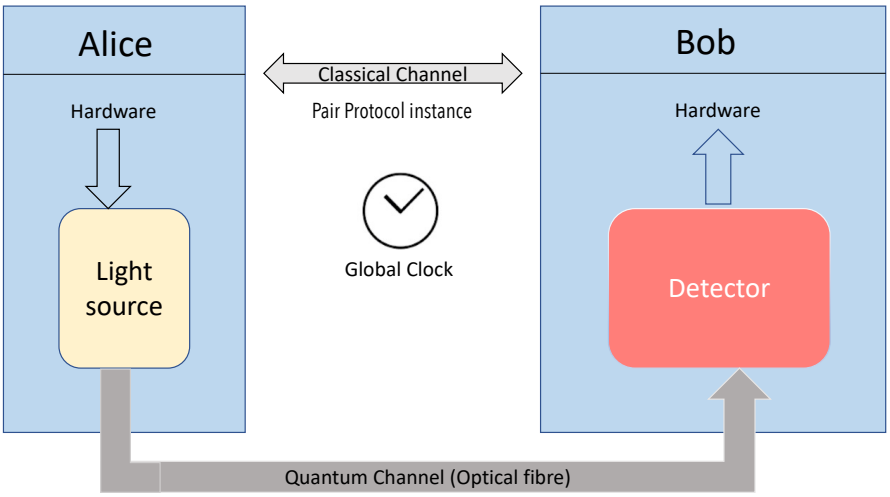

Figure 4: BB84 Topology

Figure 4 shows the BB84 topology used to simulate in the SeQUeNCe simulator. BB84, as the pioneering quantum cryptographic protocol, has asserted its position with its unconditional security, though the assumptions of the perfect quantum channel and flawless quantum detectors have been critiqued due to implementation challenges. Despite these critiques, BB84 remains a crucial and field-defining technique in quantum cryptography. The protocol also acts as a guiding model for subsequent research and development undertaken to improve real-world quantum communication networks.

4.3 Implementation

The simulation provides an implementation of the BB84 protocol for quantum key distribution. It will be later on attached to a node with a `QKDNode`. Some of the key `classes` used for the simulation of the protocol are:

- `Pair protocol instances` connects two instances of a BB84 protocol: one in the role of sender (Alice) and the other in the receiver's role (Bob).

- `Push method` is responsible for receiving requests for key generation in the BB84 protocol. It accepts parameters indicating the desired key length, the number of keys to generate, and the maximum time allowed for the key generation process. It initiates the key generation process if the conditions are met.

- `Start Protocol` initializes the BB84 Quantum Key Distribution protocol by calculating necessary parameters like *light_time* and *start_time*. It sends/schedules a *BEGIN_PHOTON_PULSE* message to the secondary party, carrying vital parameters for key generation. On completion of all key lengths, the protocol is marked as ready, indicating potential for re-initiation.

- `set_measure_basis_list` sets the basis for measurement. The function calculates the number of pulses based on *light_time* and *light source frequency*. It then generates a random basis list of 0s and 1s corresponding to these pulses. This basis list is added to the overall basis lists and is also set in the quantum state detector, determining the reference framework for qubit measurement during the simulation of the BB84 protocol.

`BB84 class` includes methods to initiate the protocol, send and receive messages, set the measurement basis for qubits, and convert a list of bits to an integer key. This class also features a function that pairs BB84 instances prior to transmission to ensure correct communication as discussed above.

Key attributes of the class include node details, flags distinguishing node roles and work status, the photon's pulse time, and the generated keys as both an integer and list of bits. For measuring the protocol's performance, attributes like *latency*, *throughput*, and *error rates* are maintained. Upon receiving requests for key generation, the *push* method can potentially initiate the protocol. The *received message* method ensures proper responses to different message types during the protocol's operation for its effective functioning.

5 Results and Discussion

In our extensive summer project, we focused on simulating quantum and classical channels over a wide range from 50 m to 99 km. The `QuantumChannel`, `ClassicalChannel`, `LightSource`, and `Detector` modules were parameterized extensively to reflect real-world conditions, as shown in Table 1.

The simulator was fed with data of various distances, with two distinct segments: short range (50 m to 2 km) and long range (1 km to 100 km). The simulation aimed to understand how *throughput*, *error rates*, and *latency* are influenced over varying distances in a quantum network. Table 1 shows the value of parameters for different modules to simulate BB84 protocol in the simulator.

Module	Parameter	Values
`QuantumChannel`	polarization_fidelity	1
	attenuation	0.0002 dB/m
	light_speed (speed of light within fiber)	2×10^{-4} m/ps
`ClassicalChannel`	delay	10 ms
`lightsource`	frequency	5 MHz
	mean_photon_number	0.3
	wavelength	795.6 nm
	encoding_type	polarization
`detector`	efficiency	0.65
	dark_count	1000 Hz
	time_resolution	0.03 ps
	Maximum count rate	10 MHz

Table 1: Parameter values for different modules used in the SeQUeNCe simulator

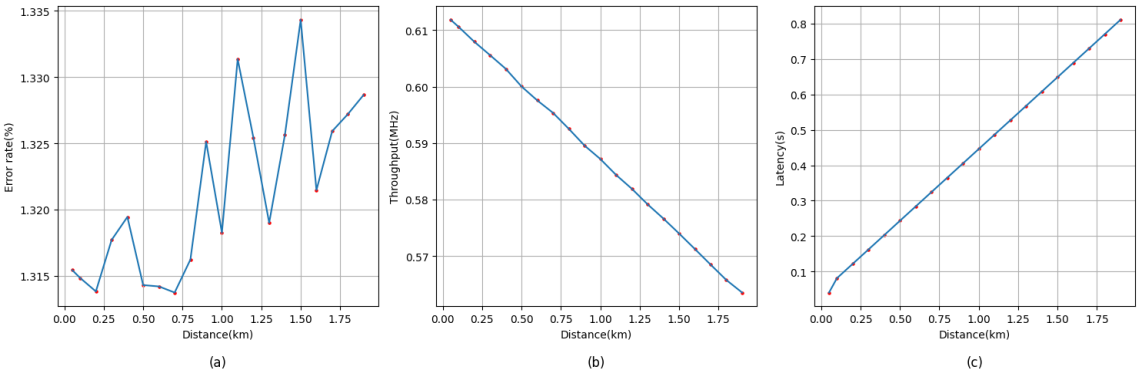

Figure 5: Graphs showing the relationship between Distance (in km) and (a) Error rate (in %), (b) Throughput (in MHz), and (c) Latency (in sec) for a long distance range of 50 m to 2 km

5.1 Short Range Results (50 m - 2 km)

Our quantum network simulation study primarily focused on exploring network performance over a variety of distances. This section discuss about short-range simulations, focusing on distances ranging from 50 m to 2 km. Figure 5 shows the graphs between variation of Distance with Error rate, Throughput and Latency in the short range distance.

Despite the nascent nature of quantum networking and the inherent complexities involved, short-range simulations surfaced several promising outcomes.

Throughput: A crucial measure of network performance, the throughput, modestly decreased from 6.1 MHz at 50 m to 5.6 MHz at the 2 km mark. High throughput is indicative of an efficient and well-functioning network. The demonstrated levels pointed towards the potential capacity of quantum networks to maintain an impressive rate of data

flow, essential for rapid information exchange. It's to be noted that actual throughput in a real-world quantum network may deviate because of yet-not-included factors like hardware limitations and physical noise.

Error Rate: A key gauge of the reliability and stability of a network, the error rate registered a small hike from 1.315% at 50 m to 1.329% at 2 km. Such slight fluctuations is yet to be find out by meticulous study of the modules involved. An actual quantum network would have to fortify its error correction systems to keep error levels low and maintain the integrity of quantum information.

Latency: Predictably, latency displayed a linear increase from 0.0405 sec at 50 m to 0.8112 sec at 2 km. When considering the speed of light in fibre as 2×10^8 m/s, the latency over distances of 50 m and 2 km would be approximately 250 ns and 10 μs, respectively.

However, the latencies observed in our simulator are higher than these values. This discrepancy is due to the fact that our latency measurement also incorporates the processing time at both the source and destination nodes, along with the transmission time. Processes such as computational delays, component inefficiencies within the quantum communication system, and network congestion may attribute to these delays. As a result, the actual latency experienced in quantum communication will likely exceed the values calculated based solely on the speed of light.

5.2 Long Range Results (1 km - 99 km)

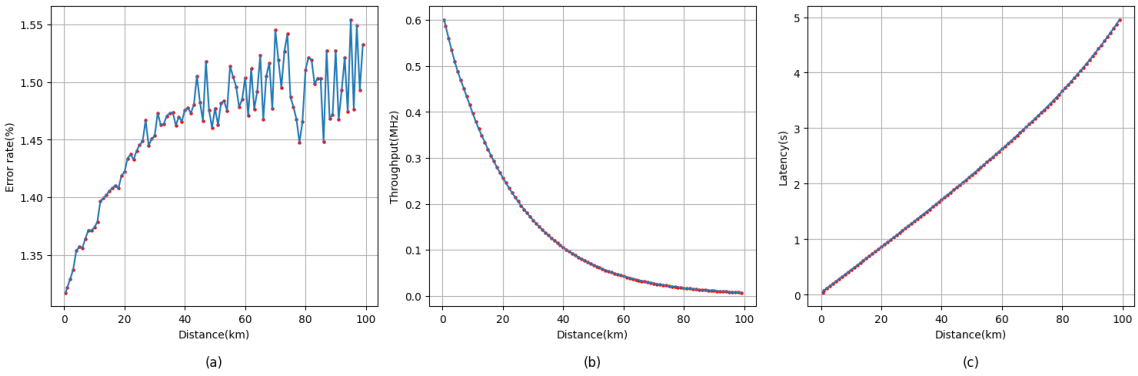

Figure 6: Graphs showing the relationship between Distance (in km) and (a) Error rate (in %), (b) Throughput (in MHz), and (c) Latency (in sec) for a long distance range of 1 km to 99 km

When observing the results over longer distances, there are several key points to note regarding throughput, error rate, and latency. Figure 6 shows the graphs between variation of Distance with Error rate, Throughput and Latency in the short range distance.

Throughput: This is a measure of how much quantum information can be transmitted over the network in a given time. In the long-range results, the throughput is observed to decrease as the distance increases from 1 km to 99 km. At 1 km, the throughput stands at 5.8 MHz, dropping to 0.7 MHz at the 99 km mark. This is a significant drop, illustrating the difficulties faced in maintaining high data rates over larger distances.

The decrease in throughput with distance can be attributed to increased photon loss and decoherence effects in the quantum channel. As the distance increases, the likelihood of a photon being absorbed or scattered out of the guideway increases and photonic qubits face more chances of losing their quantum states, both combining to lower the data rate. One more to be noted in the figure 6 is that it decreases exponentially contrary to short distance rance shown in figure 5.

Error Rate: This represents the proportion of transmitted quantum bits (qubits) that are received incorrectly. Over the greater distances, the error rate sees a slight rise, moving from 1.321% at 1 km to 1.532% at 99 km. This minor increase in error rate is yet to be examined in case of long distance range. However, protocols such as Quantum Error Correction (QEC) might be needed for practical long-distance quantum communication.

Latency: In our simulator, latency is the delay between a sender transmitting a key and a receiver receiving it. As expected, latency increases with distance. Our simulations show latency starting from 0.0812 s for 1 km and linearly increasing to 4.9530 s for 99 km. This increase in latency is due to the finite speed of light in fiber, approximately 2×10^8 m/s, which establishes a minimum limit on information transmission time from sender to receiver.

However, our simulator's latencies are higher than what light speed and distance would suggest. This happens because our latency calculation includes processing times at both source and destination nodes, computational delays, and time needed to handle various system and network inefficiencies. Thus, the actual latency in quantum communication tends to be greater than the calculated values.

Overall, long distance transmission presents more challenges for quantum networking, with reduced throughput, increased error rates, and greater latency. To tackle these issues, considerable development and improvement in quantum repeaters, correction protocols, and efficient quantum cryptographic protocols are needed. Continued research in these realms is essential for advancing long-distance quantum networking and making it practical for real world use.

6 Conclusion

This project has primarily focused on simulating quantum network behaviors, demonstrating the critical role of simulations in advancing quantum network technology. The continuing refinement of both simulation technology and quantum network design holds enormous potential for the advancement of secure, high-capacity communication systems. By implementing the BB84 protocol within the SeQUeNCe simulator across various distances, we have generated insights into the capabilities and limitations of quantum networks. While the simulated quantum network demonstrated promising capabilities, it invites further investigation and development to perfect its design for reliability.

In conclusion, based on the simulation results, quantum networks hold promising potential for efficient and secure communication, especially over short distances. The high throughput, low error rates, and low latency levels from simulation results paint an

optimistic picture of what theoretically seems achievable.

While simulations offer valuable data, it is crucial to remember that they are simplified representations of complex real-world systems. The performance characteristics observed in our simulations, such as throughput, error rates, and latency, provide essential starting points for theoretical refinement and practical exploration. Despite some challenges encountered in long range transmission simulations, such as reduced throughput and fluctuating error rates, the overall outcomes suggest significant potential inherent in quantum network simulations. Nonetheless, it is also important to acknowledge the difference between simulated environments and real-world scenarios. To optimise the performance of quantum networks in practical applications, further targeted, real-world experimental research data must be used in simulation predictions.

Nonetheless, it's essential to appreciate that these results are outcomes of a simulator designed to model theoretical expectations. Translating these results into actual quantum networks will involve grappling with physical and technological challenges that haven't been fully addressed here. Continued research and development will be needed to bridge the gap between these simulation results and the realities of experimental quantum network implementation.

Acknowledgements

I extend my sincere thanks to my supervisor, Prof. R. P. Singh, for his unwavering guidance and support throughout this research project. His deep understanding of quantum networking has greatly enhanced my learning experience.

I am also truly grateful to Dr. Shashi Prabhakar for his valuable input and dynamic discussions. My appreciation also goes to the members of the Physical Research Laboratory and fellow researchers of the Quantum Technologies group for their invaluable contributions to this endeavour. Thank you all for making this project a remarkable journey.

References

[1] Wu, X. et al. (2021) 'SeQUeNCe: a customizable discrete-event simulator of quantum networks', Quantum Science and Technology, 6(4), p. 045027. Available at: https://doi.org/10.1088/2058-9565/ac22f6.

[2] X. Wu, A. Kolar, J. Chung, D. Jin, T. Zhong, R. Kettimuthu and M. Suchara. "SeQUeNCe: Simulator of QUantum Network Communication." GitHub repository, https://github.com/sequence-toolbox/SeQUeNCe, 2021.

[3] Biswas, A. et al. (2021) 'Experimental Side Channel Analysis of BB84 QKD Source', IEEE Journal of Quantum Electronics, 57(6), pp. 1–7. Available at: https://doi.org/10.1109/JQE.2021.3111332.

[4] Biswas, A. et al. (2022a) 'Quantum key distribution with multiphoton pulses: an advantage', Optics Continuum, 1(1), pp. 68–79. Available at: https://doi.org/10.1364/OPTCON.445727.

[5] Donahoo, M.J. and Calvert, K.L. (2009) TCP/IP sockets in C: practical guide for programmers. 2nd ed. Amsterdam Boston: Morgan Kaufmann (The Morgan Kaufmann practical guides series).

[6] Dür, W. and Briegel, H.J. (2007) 'Entanglement purification and quantum error correction', Reports on Progress in Physics, 70(8), p. 1381. Available at: https://doi.org/10.1088/0034-4885/70/8/R03.

[7] Gunkel, M., Wissel, F. and Poppe, A. (2019) 'Designing a Quantum Key Distribution Network - Methodology and Challenges', in Photonic Networks; 20th ITG-Symposium. Photonic Networks; 20th ITG-Symposium, pp. 1–3.

[8] Kozlowski, W. et al. (2023) Architectural Principles for a Quantum Internet. Request for Comments RFC 9340. Internet Engineering Task Force. Available at: https://doi.org/10.17487/RFC9340.

[9] Mehic, M. et al. (2017) 'Implementation of quantum key distribution network simulation module in the network simulator NS-3', Quantum Information Processing, 16(10), p. 253. Available at: https://doi.org/10.1007/s11128-017-1702-z.

[10] Mishra, S. et al. (2022) 'BBM92 quantum key distribution over a free space dusty channel of 200 meters', Journal of Optics, 24(7), p. 074002. Available at: https://doi.org/10.1088/2040-8986/ac6f0b.

[11] Specht, H.P. et al. (2011) 'A single-atom quantum memory', Nature, 473(7346), pp. 190–193. Available at: https://doi.org/10.1038/nature09997.

[12] Zang, A. (2022) 'Simulation of Entanglement Generation and Distribution - Towards Practical Quantum Networks', in Proceedings of the 2022 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation. New York, NY, USA: Association for Computing Machinery (SIGSIM-PADS '22), pp. 53–54. Available at: https://doi.org/10.1145/3518997.3534992.

[13] Zang, A. et al. (2022) 'Simulation of Entanglement Generation between Absorptive Quantum Memories', in 2022 IEEE International Conference on Quantum Computing and Engineering (QCE). 2022 IEEE International Conference on Quantum Computing and Engineering (QCE), pp. 617–623. Available at: https://doi.org/10.1109/QCE53715.2022.00084.

[14] An Illustrated Introduction to Quantum Networks and Quantum Repeaters — AWS Quantum Technologies Blog (2022). Available at: https://aws.amazon.com/blogs/quantum-computing/an-illustrated-introduction-to-quantum-networks-and-quantum-repeaters/ (Accessed: 26 July 2023).

7 Appendix

Data Visualizations

Following code was used to plot Figure 5 and Figure 6

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the data
data200m = pd.read_csv("distance_bb84.csv")

fig, axs = plt.subplots(1, 3, figsize=(15,5))

axs[0].plot(data200m['Distance']/1000, data200m['Error_rate']*100)
axs[0].scatter(data200m['Distance']/1000, data200m['Error_rate']*100,
      color='red', marker='.', s=20)
axs[0].set_xlabel('Distance(km)')
axs[0].set_ylabel('Error rate(%)')
axs[0].grid()
axs[0].text(0.5, -0.2, "(a)", size=12, ha="center", transform=axs[0].
      transAxes)

axs[1].plot(data200m['Distance']/1000, data200m['Throughput']/(10**6))
axs[1].scatter(data200m['Distance']/1000, data200m['Throughput']/(10**6)
      , color='red', marker='.', s=20)
axs[1].set_xlabel('Distance(km)')
axs[1].set_ylabel('Throughput(MHz)')
axs[1].grid()
axs[1].text(0.5, -0.2, "(b)", size=12, ha="center", transform=axs[1].
      transAxes)

axs[2].plot(data200m['Distance']/1000, data200m['Latency'])
axs[2].scatter(data200m['Distance']/1000, data200m['Latency'], color='
      red', marker='.', s=20)
axs[2].set_xlabel('Distance(km)')
axs[2].set_ylabel('Latency(s)')
axs[2].grid()
axs[2].text(0.5, -0.2, "(c)", size=12, ha="center", transform=axs[2].
      transAxes)

plt.tight_layout()
plt.show()
```