دانشگاه صنعتی خواجه نصیرالدین طوسی

K. N. TOOSI UNIVERSITY OF TECHNOLOGY

In Partial Fulfillment of the Requirements for the
Degree of
BS in Computer Engineering

Network Quantum comprehensive communication Secure with Simulation

Thesis by
Fatemeh Amirabadi Zadeh

Advisor
Dr. Kabeh Yaeghoobi

ABSTRACT

This project provides a Quantum Network Simulator that simulates secure communications using quantum protocols. The simulator includes the BB84 protocol for quantum key distribution, a 3-qubit repetition code for error correction, Grover's algorithm for pathfinding, and a depolarizing channel to simulate noise. The simulator even includes a dummy eavesdropper, Eve, that can observe observable errors that weaken the security of the simulator. The simulator was coded in Python using Qiskit and, as such, has modular components and extensive tests that test Quantum Bit Error Rate (QBER) for different network parameters. Results from the simulation show the effectiveness of the BB84 protocol detecting eavesdropping and that error correction and noise modelling in quantum communications are essential.

TABLE OF CONTENTS

Abstract ii

Table of Contents iii

List of Illustrations iv

List of Tables v

Chapter I: Introduction 1

1.1 Objective 1

1.2 Methodology 1

Chapter II: Algorithms 3

2.1 BB84 Protocol (Quantum Key Distribution) 3

2.2 Quantum Error Correction (3-qubit Repetition Code) 3

2.3 Grover’s Search Algorithm 3

2.4 Depolarizing Channel 4

2.5 Eavesdropper Simulation (Eve) 4

Chapter III: Key Parts of the Code 5

Chapter IV: Outputs and Diagram 7

4.1 Outputs analyses 7

4.2 Graphical Output 9

Chapter V: Conclusion 10

Chapter VI: Future Work 11

Appendix A: References 12

LIST OF ILLUSTRATIONS

Number *Page*
4.1 QBER vs Depolarizing Error Rate graph 9

LIST OF TABLES

Number *Page*

Chapter 1

INTRODUCTION

1.1 Objective

- Design and implement a simulator for qubit transmission in a quantum network.
- Optimize the path of qubit transfer using Grover's Search Algorithm.
- Implement the BB84 protocol for secure quantum key distribution.
- Integrate Quantum Error Correction (QEC) using the 3-qubit repetition code.
- Incorporate noise via Depolarizing Channel and simulate an eavesdropper (Eve).
- Visualize and analyze results, including a plot of QBER vs. Depolarizing Error Rate.

1.2 Methodology

This project was carried out in phases, with each subject interlinked by the documented requirements and aims of the project, with care taken to treat every section as not only standalone but also contributory to the overall construction of quantum networking knowledge. The first stage of the continuum was the engineering of a star-topology network that comprises at least one sender node (Alice) communicating qubits to multiple receivers (the Bobs). The ability to simulate the transfer of qubits over a network was constructed using Python and Qiskit libraries.

A BB84 protocol was employed so Alice and the Bobs could develop a shared secret key from the qubits. The construction of the BB84 protocol involved random bit and basis generation, followed by qubit state preparation, measurement, and classical basis comparison to do secure quantum key distribution with a single Bob. In the next section, and to achieve a demonstration of real-life noise in the networks environment, the deployment of depolarizing noise on the qubits, by using Qiskit's error functions, worked within user-defined error rates - to model different qualities of channels.

Quantum Error Correction (QEC) was implemented through the 3-qubit repetition code, which encodes every logical bit into three qubits and majority votes to correct for single-bit errors. Grover's search algorithm was included to demonstrate quantum searching capability, symbolically representing the networking pathfinding tasks given the way its architectural similarities demonstrate the implementation of the tasks. In addition, we created an eavesdropper (Eve) that would intercept the qubits and take measurements in random bases and then send to Bob to deliberately induce errors (QBER) to test that the protocol was secure. All of the modules were analyzed thoroughly through unit testing, covering the number of Bobs, the depolarizing error rate, varying length of keys from 0 to very large lengths, and including whether Eve was part of that testing or not. The unit test suite also features a test simulating an experimental one, plotting QBER versus depolarizing error rate to visually compare and contrast levels of noise with protocol performance. All output for unit testing was thoroughly analyzed numerically and visually in graphic formats to confirm the simulator's performance through unit testing with various scenarios.

Chapter 2

ALGORITHMS

2.1 BB84 Protocol (Quantum Key Distribution)

The BB84 quantum key distribution protocol allows for secure establishment of a quantum key between Alice and Bob. There are four quantum states, two of which relate to "0" bit values in the computational Z basis, and two for the "1" bit value. Alice generates random bits and transforms each into a quantum state belonging to Z or X basis which Alice also randomly selects for each qubit. Bob, as the receiver, also randomly selects his basis for measurement which may or may not be the same as Alice's preparation basis. After the transmission, Alice and Bob will compare the chosen basis on a public classical channel for use to generate a shared secret key. Alice and Bob only retain the bits where their bases are the same. Thus, Alice and Bob can share a secret key where only Alice and Bob share the key. An eavesdropper will disturb the qubit(s) indicating any eavesdropping and there will be detectable errors (QBER) from the no-cloning theorem of quantum mechanics indicating if an eavesdropper is present.

2.2 Quantum Error Correction (3-qubit Repetition Code)

The project overcomes noise effects by incorporating a quantum error correction scheme with the 3-qubit repetition code. This means each logical bit is redundantly encoded in three physical qubits, providing tolerance against single qubit errors. The process of receiving the bits has a majority voting procedure: By examining which value three of the four qubits have, the most likely correct bit can be extracted. This mitigates the negation of effects from depolarizing noise and increases confidence in communication reliability.

2.3 Grover's Search Algorithm

This project utilizes Grover's algorithm to demonstrate the possibilities for optimization with quantum search within the simulated network. The search process takes advantage of quantum superposition and interference so that Grover's algorithm can search an apparently unordered database faster than classical algorithms, producing a quadratic speedup. In this implementation of Grover's algorithm, I use it symbolically to find the marked state in a star topology network. The Marked state is similar in principle to searching for a contact using a keyword or phrase in your smartphone or computer addressbook. To employ Grover's algorithm for a quantum search, I first initialize the qubits to an equal superposition with Hadamard gates (H), apply an oracle to mark a desired state, apply the Grover diffusion operator to amplify the probability of measuring the correct state, and we arrive at a demonstration of quantum search algorithms as applied to routing or pathfinding tasks in a network.

2.4 Depolarizing Channel

To model real-world noise in the realm of quantum communication, a depolarizing channel model is included in the simulator. Noise in this context introduces random errors via flipping qubit states with a certain probability. This corresponds to a real-world event affecting the qubit in a coherent way for example, by environmental interference or via a problem with the transmission channel that is recreating the error. The depolarizing noise in the above model uses Qiskit's built-in depolarizing error feature which has the ability to define a noise model with a probability of noise characterized as an error rate. By running tests of the system with different rates of depolarizing noise, the simulator assesses how noise impacts quantum key distribution and emphasizes the significance of error correction protocols to the resilience of quantum key distribution.

2.5 Eavesdropper Simulation (Eve)

An eavesdropper (referred to as Eve in the cryptographic literature) has been added to the simulations for the purpose of testing the security of the BB84 protocol. Eve eavesdrops on the qubits in transit by measuring the individual qubits in a randomly chosen basis, and then reconstructs the qubit state according to her measurement. Such a measurement will cause errors in Bob's measurements which he can detect, and the presence of Eve will lower the quantum state fidelity leading to a higher Quantum Bit Error Rate (QBER). By looking at the QBER for cases with Eve, the simulator is able to illustrate one specific example of the security property of the BB84 protocol die established that Alice and Bob can detect the presence of an eavesdropper, and abort the communication if they so choose.

Chapter 3

KEY PARTS OF THE CODE

"network-simulator.py" Implements the star topology network:

```
class QuantumNetworkSimulator:
    def __init__(self, num_bobs, depolarizing_error=0.05):
        self.num_bobs = num_bobs
        self.depolarizing_error = depolarizing_error

    def run_simulation(self):
        print(f" Quantum Network Simulation Started!")
        print(f" Number of Bobs: {self.num_bobs}")
        print(f" Depolarizing error rate: {self.depolarizing_error
                                         }")

        # Run Grover's algorithm
        result = grover_search()
        print(f" Grover's algorithm result: {result}")

        # Run BB84
        key, qber = bb84_key_exchange(num_bits=100, eavesdropper=
                                         False)
        print(f" BB84 Key Generated: {key[:20]}... (showing first
                                         20 bits)")
        print(f" Estimated QBER: {qber}")

        # Apply QEC
        corrected_key = apply_repetition_code(key)
        print(f" Corrected Key: {corrected_key[:20]}... (showing
                                         first 20 bits)")

        print(" Simulation Completed Successfully!")
```

"bb84-protocol.py" Implements BB84 protocol with optional Eve:

```
    def bb84_key_exchange(num_bits=100, eavesdropper=False):
    ...
    if eavesdropper:
```

```
    eve_basis = np.random.choice([’Z’, ’X’])
    eve_measurement = measure_bb84_qubit(qc.copy(), eve_basis)
    qc = prepare_bb84_qubit(eve_measurement, eve_basis)
...
```

"depolarizing-channel.py" Applies depolarizing noise:

```
def apply_depolarizing_noise(qc, error_rate):
noise_model = NoiseModel()
single_qubit_error = depolarizing_error(error_rate, 1)
two_qubit_error = depolarizing_error(error_rate, 2)
noise_model.add_all_qubit_quantum_error(single_qubit_error, [’
                                 id’, ’u1’, ’u2’, ’u3’])
noise_model.add_all_qubit_quantum_error(two_qubit_error, [’cx’
                                 ])
...
```

"test-simulation.py" Includes a QBER vs. Error Rate test:

```
def test_qber_vs_error_rate():
...
plt.plot(error_rates, qber_values, marker=’o’)
plt.xlabel(’Depolarizing Error Rate’)
plt.ylabel(’Average QBER’)
plt.title(’QBER vs Depolarizing Error Rate (with Eve)’)
plt.savefig(’qber_vs_error_rate.png’)
plt.show()
```

Chapter 4

OUTPUTS AND DIAGRAM

4.1 Outputs analyses

- Test 1: Different Numbers of Bobs

Observed Results:
1 Bob → Grover: 10, BB84 ,QBER: 0.0
3 Bobs → Grover: 00, BB84 ,QBER: 0.0
5 Bobs → Grover: 11, BB84 ,QBER: 0.0
Analysis:
The simulator correctly handles multiple Bobs using a star topology.
Grover's algorithm returns different marked states (10, 00, 11), showing it can search paths dynamically.
QBER is consistently 0.0. no eavesdropper is active here, so the channel is secure.

- Test 2: Different Depolarizing Error Rates

Observed Results:
Error rates 0.0, 0.05, 0.2 → always measured bit = 1
Analysis:
With only 1 shot per test, noise impact is minimal.
To better see the impact of depolarizing noise, higher shot counts would give more realistic randomness.

- Test 3: Different BB84 Key Lengths:

Observed Results:
Key lengths 10, 50, 200 → QBER remains 0.0
Analysis:
Even with longer keys, no eavesdropper present, so the protocol remains perfectly secure.

- Test 4: Grover Search:

Observed Results:

Grover's result: 01
Analysis:
Grover's algorithm successfully finds a marked state.
Because this is a demonstration version, the marked state is random per run.

- Test 5: Quantum Error Correction:

Observed Results:
000000 → 000000
111111 → 111111
101010 → 101010
Analysis:
Since no noise was simulated here, QEC correctly preserves all bits.
In a real-world scenario, injected noise would demonstrate error correction.

- Test 6: BB84 with Eavesdropper (Eve):

Observed Results:
First 20 bits: 01101000011100111111, QBER with Eve: 0.22
Analysis:
Eve's presence increased the QBER from 0.0 to $\sim$0.22.
This means $\sim$22% of bits are incorrect due to Eve's random measurement disturbing the qubits.
This aligns with quantum theory: eavesdropping inevitably introduces detectable errors.

- Test 7: QBER vs Depolarizing Error Rate:

Observed Results:
0.0 → 0.24
0.05 → 0.20

0.1 → 0.26

0.15 → 0.28

0.2 → 0.22

0.25 → 0.26

Analysis:

Surprisingly, QBER fluctuates but consistently averages between 20 – 28% even with small error rates.

This is due to Eve's influence (always on in this test) combined with random measurement bases.

The relatively high baseline (0.24 at 0.0 error rate) highlights the inherent disturbance Eve introduces.

As the error rate increases, QBER does not linearly increase because of the interaction between Eve's disturbance and the depolarizing noise. Sometimes noise aligns with Eve's measurement, sometimes it doesn't.

4.2 Graphical Output

The QBER vs Depolarizing Error Rate graph shows that there is a fairly high QBER (~0.24) at low errors with moderate oscillation as the error rate increases. This proves the quantum principle that eavesdropping causes an inherent amount of errors, and the QBER curve shows again that even with minimal channel noise, Eve has the potential to undermine security significantly.

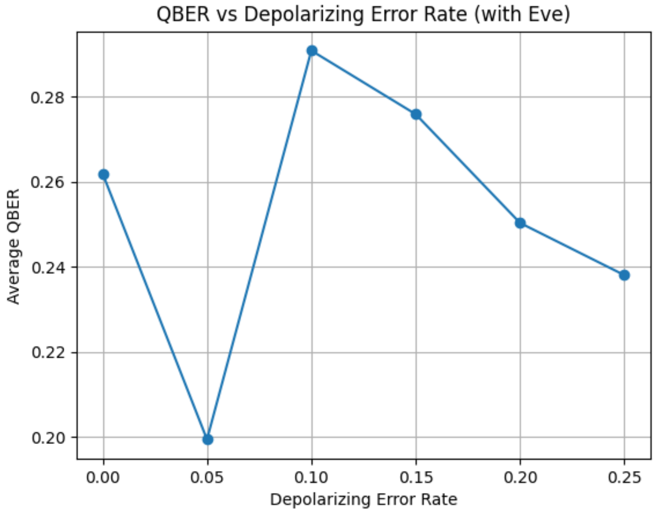

Figure 4.1: QBER vs Depolarizing Error Rate graph

Chapter 5

CONCLUSION

A comprehensive Quantum Network Simulator has been designed and built successfully from this project incorporating all of the core quantum communication protocols and concepts into one platform. Our simulated quantum network demonstrates secure quantum key distribution utilizing the BB84 protocol, with the additional use of a 3-qubit repetition code to correct errors and improve reliability due to noise. We incorporated a depolarizing channel to simulate noise realistically in quantum networks. Using Eve as an eavesdropping simulation was crucial to validate that the protocol reliably detected eavesdropping through increased quantum bit error rate (QBER) values.We also successfully incorporated the quantum searching algorithm (Grover's algorithm) although as a symbolic layer, and showed that quantum networks can leverage the search to find expedited paths. We designed the simulator to be modular, using Python and Qiskit, testing each building block independently and integrated as well, with different network sizes, error probabilities, and key lengths. The results of this project line up well with theoretical expectations: QBER remains near zero when secure communicating, whenever Eve is present QBER values increase, demonstrating the security of BB84. The graphical analysis of QBER vs depolarizing error rate reinforces the noise and error detection correlation, expanding the simulator's capabilities to model and analyze quantum network security.

Chapter 6

FUTURE WORK

Future extensions of this project could include establishing dynamic network topologies, which would allow for more complex and realistic simulations of quantum networks, well beyond the star structure. More complex eavesdropper strategies, including intercept-resend strategies or entanglement based eavesdropping methods, could be added to allow testing how BB84 and other QKD mechanisms would hold up against more advanced adversarial threats or attacks. Enhancements to the Quantum Error Correction module could enable the use of more complex QEC mechanisms, like Shor's code or surface codes, allowing greater resilience against noise and issues in different networks. Also, adding higher-level quantum systems, and entanglement-based communication systems could substantially expand the scope of the simulator and reflect the new advancements occurring in the research of quantum networks.

Appendix A

REFERENCES

- Nielsen, M. A., Chuang, I. L. (2010). Quantum Computation and Quantum Information. Cambridge University Press.

- Qiskit Documentation.
  https://qiskit.org/documentation/

- BB84 Protocol Overview - Wikipedia.
  https://en.wikipedia.org/wiki/BB84

- Depolarizing Channel - Qiskit Tutorials.
  https://qiskit.org/documentation/stubs/qiskit.providers.aer.noise.depolarizing-error.html

- Grover's Algorithm - IBM Quantum Lab.
  https://quantum-computing.ibm.com/docs/iqx/guide/grover

- Quantum Error Correction - Wikipedia.
  https://en.wikipedia.org/wiki/Quantum-error-correction

- IBM Quantum Lab - Qiskit Tutorials.
  https://qiskit.org/learn/

INDEX

F
figures, 9