Error Correction in Quantum Networks using Shor Code

*Abstract—***As quantum computing comes closer and closer to the forefront of computing, so do quantum networks. All systems suffer from noise, and all networks need error correction. However, an inherent trait of qubits is that they can be perturbed, collapse and change their value when subjected to noise. Quantum networks therefore depend much more on reliable error detection and correction to be usable. In this paper I explore quantum error detection and correction, focusing on the performance of the Shor Code and its impact on network performance. I also tested the error correction performance with an implementation of a quantum repeater, which is necessary for long-range quantum communication.**

Keywords: Quantum Networking, Quantum Communications, Shor Code, Quantum Error Correction

I. Introduction

Classical error correction relies on redundancy, where multiple copies of the data is stored and if some elements of the data don't agree then the most common value is taken as the correct one. However, this is not possible in quantum systems because of the no-cloning theorem. Furthermore, a classical error consists only of a binary bit flip. In quantum computing, there are numerous possible states and both phase (spin angle) and sign (spin direction) changes must be accounted for.

The main method of quantum error correction is to distribute the state of a single entangled qubit onto multiple qubits, called entanglement spreading [1, 2]. This creates a multipartite quantum system that can be transmitted [1], and then decoded at the receiving end. A syndrome measurement can be taken on the multipartite system, allowing errors to be detected.

Noise is everywhere, and errors can occur in all parts of a quantum system, including: generation of qubits, transmission of qubits, collapse of qubits due to expiry, perturbation of qubit due to noise, measurement errors, faulty gates, storage-related errors. Unmentioned thus far is one of the most important types of errors that quantum networks face, and that's errors due to eavesdropping. Qubits tend to collapse during measurement, this means their phase collapses to 0 or 1 and any entanglement is lost—this means that an eavesdropper cannot sniff the data on the network without altering the data in transmission. If we don't detect these errors, then the data has no integrity and can't be trusted because an eavesdropper could be on the network.

II. Background

A checksum transmitted over a classical network can also provide highly reliable error detection for the qubits. However, fully quantum algorithms are more secure, because there's no way to reverse engineer the laws of physics [1].

Fully-quantum error correction algorithms distribute the state of a qubit onto as few as three qubits to determine if the phase has changed or determine if the sign has changed—but not at the same time. Furthermore, no corrections can be made with this information. It takes a minimum of nine qubits to determine sign flips and phase changes and be able to correct them. The most prominent nine-qubit correction code is named Shor Code, after Peter Shor who first proposed it [2].

$$|0_S\rangle = \frac{1}{2\sqrt{2}}(|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle)$$
$$|1_S\rangle = \frac{1}{2\sqrt{2}}(|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle)$$

Figure 1: Shor code transformation

There is extra computing power required to generate nine-qubits, store them, and more bandwidth is needed to transmit them. Therefore, I examined how the extra computing requirements affects the performance of a simulated quantum network and determine whether or not it is viable to use it for real-world error correction.

Shor code is a very well-established method for error correction, however, it has its limitations:

Pros

Certain phase shifts will not require any correction, as the output of the decoding will still result in the correct qubit.

It is architecturally simple to implement because it consists of only Hadamard, CNOT and T gates. The former two of which are fundamental in all quantum systems for entanglement already.

It can correct sign flips, phase shifts and occurrences on both.

Cons

The algorithm is only guaranteed to be successful if only one qubit encountered an error. Links with higher than an 11% error rate may fail to transmit data using Shor code.

It is expensive to generate and store nine qubits.

III. Motivation

If we can determine a method to detect and correct errors, while determining the integrity and ensuring the security of data, it will be a huge step towards making a feasible quantum network.

Furthermore, it will allow software developers who are trying to prepare for the advent of quantum networking by providing them with a specific algorithm for link-level software.

Much work has been done looking at the mathematical aspects of quantum error correction algorithms, and there are as a result many competing algorithms tailored for specific scenarios. However, I want to find one that is suitable for all general-purpose communications and determine the cost and performance impacts it will take to implement it in a real network.

IV. Problem Definition

The problem I am attempting to solve is to find a method of detecting and correcting qubits in quantum networks. The algorithm must be able to detect sign flips and phase shifts in noisy networks.

I propose that Shor code is a suitable error correction algorithm for all general-purpose quantum communications.

V. Experimental Setup

Quantum networks are still in their infancy; therefore, I used the quantum network simulator software SimulaQron [3]. I chose it because it is developed very closely with the IETF’s Quantum Internet Research Group [4] who are actively developing and proposing the standards that will define a quantum internet. It is also open source, which allowed me to make some necessary modifications needed to run my experiments.

I also made use of IBM’s quantum computing platform, IBM Q. These are a set of real quantum computers located at IBM facilities around the world [5], on which quantum logic gates can be configured and executed.

Implementing a Quantum Repeater

The first step to making a suitable quantum network simulation was to implement a quantum repeater—these will be a fundamental part of quantum networks due to the short expiry time of qubits [4]. The no-cloning theorem makes it impossible to implement repeaters in a classical way, which just duplicates bits and then forwards the new bits. Instead, in a quantum repeater two new qubits are generated. The entanglement is then swapped from the incoming qubit, onto one of the qubits that was generated [7]. The other qubit is then transmitted onto the next node in the network. This effectively replaced an aging, less stable qubit, with a new qubit and strengthened the entanglement.

When implementing Shor Code, it is not one single qubit that comes into the repeater. Instead, it is a multipartite system of nine qubits [2]. There were two possible approaches to handling the incoming system of qubits, one would be to do an entanglement swap from one system to another. The other approach is to de-encode the system of qubits into the original qubit. Then, during the de-encoding process errors encountered between the current node and the previous node are corrected. This implements hardware error correction at each hop on the network route, allowing much higher fidelity of data. Furthermore, it requires only one entanglement swap, and then the data is re-encoded using Shor code and then forwarded.

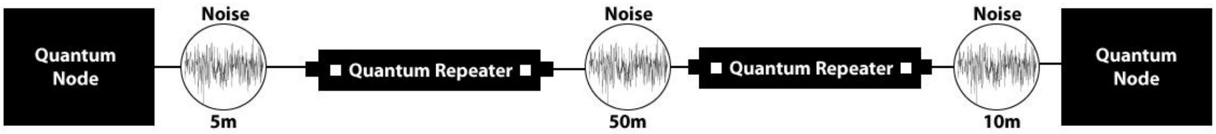
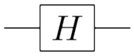


Figure 2: Quantum network components diagram

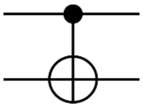
It may seem intuitive that this would allow the repeater to read the data, however, the qubit still can't be measured without risking collapse, therefore the data remains secure.

Shor Decode & Encode Functions

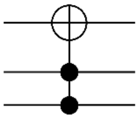
The next step is to implement the Shor decoding. When a system of qubits is received by the repeater, the Shor decode function translates the system into a qubit with the same value and entanglement as the qubit that was encoded. However, nine qubits must be acted upon. If a qubit is lost en route then something must fill the gap. I took the step of assuming that a lost qubit would be the equivalent to a qubit that had collapsed due to error, and therefore I generated a qubit in the basis state of |0> in place of any lost qubits.



is the Hadamard gate;



is the CNOT gate;



is the Toffoli gate.

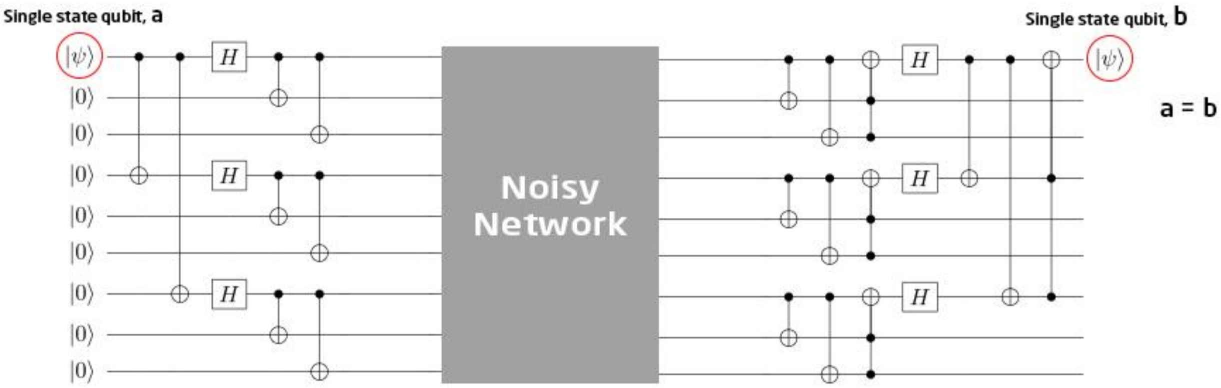


Figure 3: Quantum logic gate implementation of Shor code

The Shor decode function requires Toffoli gates. These are not fundamental gates and are not supported in SimulaQron. Therefore, it would make sense to simply implement a decomposition of the Toffoli gate using fundamental gates.

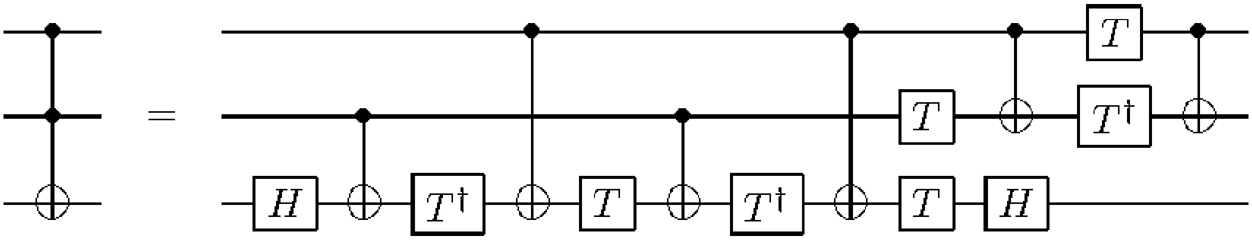


Figure 4: Toffoli gate decomposition

The Toffoli gate decomposition includes a T† gate, which is not included in any of the SimulaQron backends. Therefore, I had to implement it myself in the Qutip backend. I have since made a pull request to the Qutip official repository, so the gate will eventually be included by default, callable as `remote_apply_inverse_T`.

The T† gate: $T^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & exp(\frac{-i\pi}{4}) \end{pmatrix}$

The T gate: $T = \begin{pmatrix} 1 & 0 \\ 0 & exp(\frac{i\pi}{4}) \end{pmatrix}$

Once the T† exists, the full Toffoli gate can be written by implementing the decomposition in Figure 4.

Now that the decode function is written, the much more straightforward Shor encode function needs to be written. This function implements only fundamental quantum gates, which are provided as standard in SimulaQron. The 11 gates shown on the left of Figure 3 are all that is required.

Entanglement Swapping

In between the decode and re-encoding, the entanglement of the incoming qubit must be swapped. The first step in conducting an entanglement involves taking the CNOT value of one of the repeater's new qubit and mapping it into the incoming qubit. Next, put the same repeater-generated qubit through a Hadamard gate, to effectively undo the entanglement between both qubits and their partners [1].

To move the entanglement to the repeater's new qubit, a Bell-state measurement must be taken of both the qubits that we have been working with [7]. Again, this won't reveal the data that we're trying to hide because the previous step undid the entanglement from the qubit at the source. Instead, this combined measurement will make the repeater's qubit entangled with the source qubit.

Some corrections must be made at this point, If the source qubit measures a parity of 1, then the repeater's qubit need to have the Pauli Z gate applied to it to correct the perturbation caused by the measurement. Furthermore, if the measurement of the repeater's qubit's parity is 1, a signal is sent over a classical connection back to the source to tell it to apply the Pauli X gate—to correct the perturbation again. Each router sends these signals to the source if there was a perturbation that needed correcting.

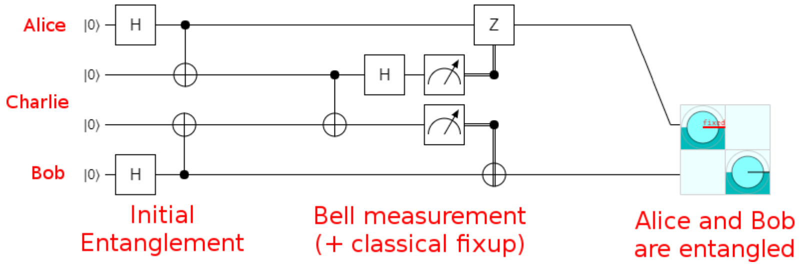

Alice
Charlie
Bob
|0⟩
|0⟩
|0⟩
|0⟩
H
H
H
Z
fixed
Initial Entanglement
Bell measurement (+ classical fixup)
Alice and Bob are entangled

Figure 5: Entanglement swap procedure [6]

Modelling Transmission Noise

Once the repeater system is fully implemented, the next step to developing the simulation is to add a noise model into the system. Numerous papers refer to a common noise model that is close to the type of noise that qubits undergo during transmission [9, 10].

Two types of noise occur in a quantum network:

1) Dephasing noise, which changes the phase of the qubit and causes the probability of measuring a 0 or 1, similar to a classical bit flip.

2) Depolarizing noise, which causes to a sign-flip because it changes the polarity of the spin and therefore the sign of the phase.

Modelling the noise:

The following two equations can be used, they are density matrices which can be applied to mixed qubit states to statistically manipulate their value [7].

1) $Pq\ (\rho) = q\rho + (1 - q)Z\rho Z \qquad (1)$

2) $Dq\ (\rho) = q\rho + (1 - q)½ \qquad (2)$

ρ is a single-qubit state
½ is a maximally mixed single-qubit state
q ∈ [0, 1] is the noise parameter
Z is the Pauli Z gate

Once again, not all of the SimulaQron backends support mixed qubit states. Using Qutip the models can be implementing by applying each of them to the matrices that represent the qubit, with a user-defined noise parameter.

In between each node, the noise is applied to simulate environmental noise that will occur as the qubit travels through a cable. The amount of times the noise is applied is proportional to the simulated length of the wire.

Measuring Repeater Delay

A simulator can't provide realistic values for the delays caused by having the extra gates involves in each repeater transaction. Instead, I used IBM Q's real quantum computers to execute tests to measure the amount of time taken by a repeater. By using Shor decode and encode at each repeater, it adds 65 extra quantum gates.

Using IBM Q's 16-qubit Melbourne-based quantum computer, I ran 8,192 tests which each implemented an example repeater transaction.

Metrics

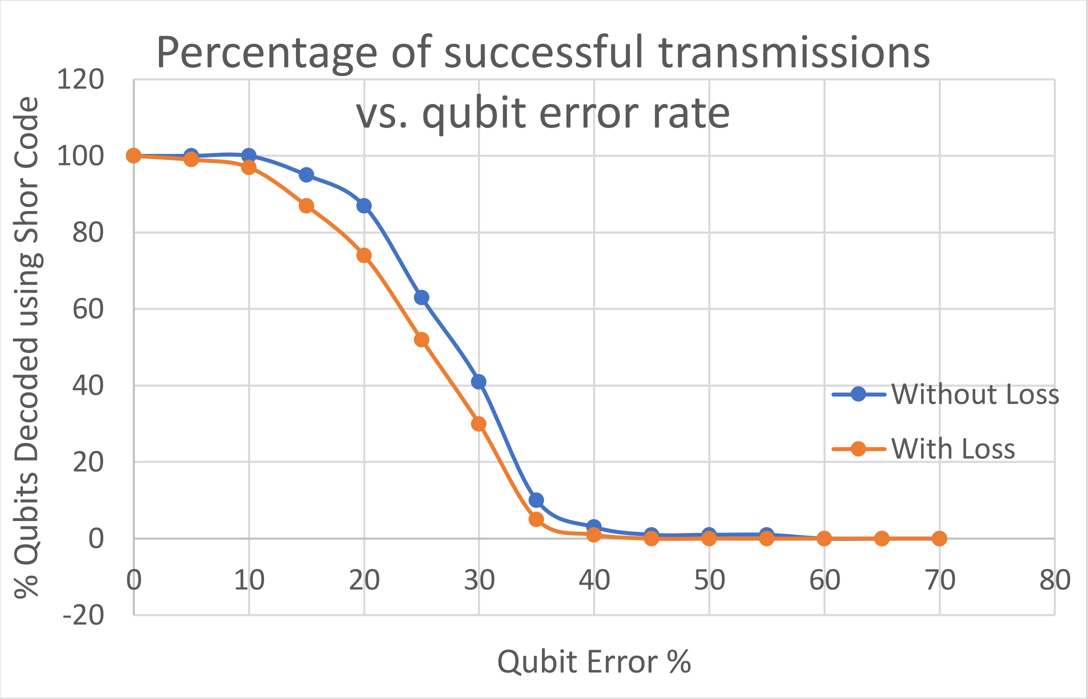
The following metrics will be used to measure the performance that Shor code gives under different amounts of noise, distance and number of intermediate repeaters.

1. Measure the total error rate (perturbed photons + lost photons) vs. the proportion of data that could be decoded successfully—with the common noise model and with different types of noise;
2. Measure and compare the data rate vs. baud rate and the bandwidth cost of using Shor code;
3. Graph the effects of increasing distance vs. the success rate of; transmission;
4. Measure the time delay added by each repeater;
5. Compare the complexity of an implementation that uses Shor code to a non-ECC implementation.

VI. RESULTS

Shor decode success rate

Looking at the success rate of decoding data based on the total proportion of errors in the network allows us to see how effective Shor code is and what its limits are in terms of noise.



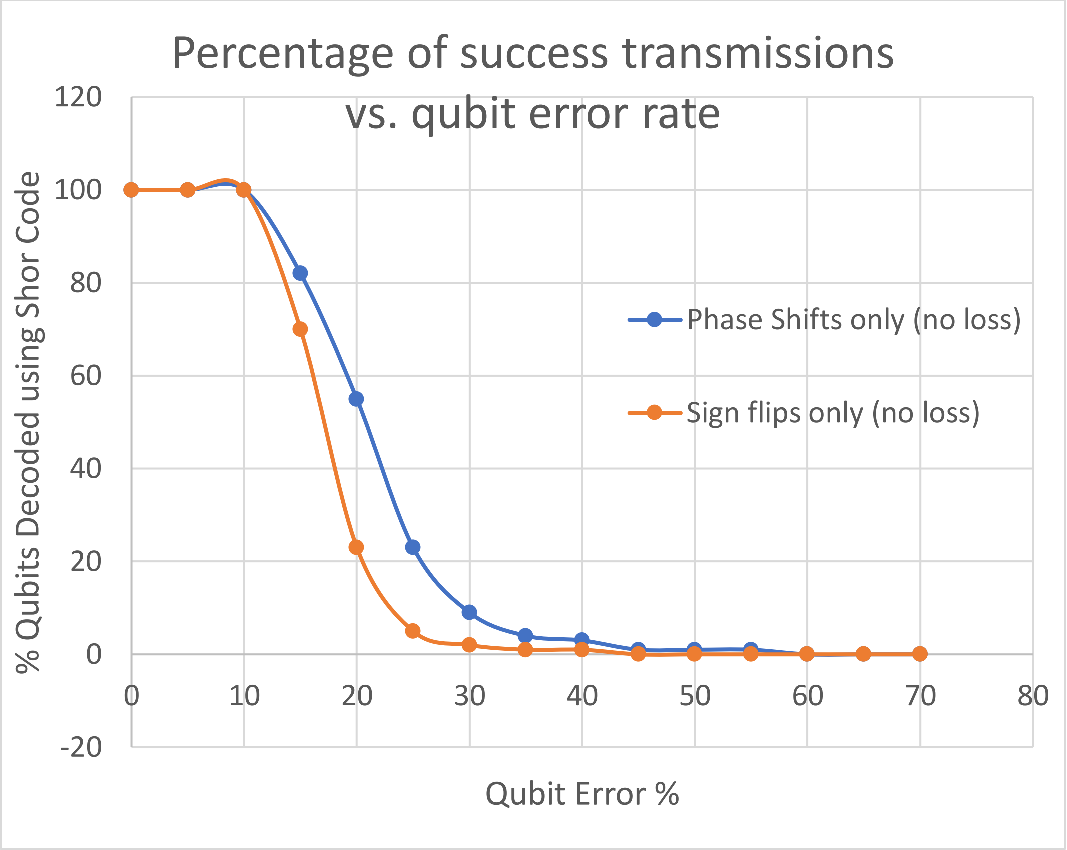
Based on the percentage of qubits that underwent perturbation (phase shifts and sign flips) the algorithm was found to have a 100% success rate for error rates of 11% and below. If loss is introduced to the system, the performance was slightly worse.

The inflection point occurs at 35% error, where the error rate exceeds Shor code's ability to successfully decode the data.

It is expected to occur at around a 30% error rate, because that means 2-3 of the 9-qubits have been perturbed. Shor code can handle a sign-flip, and a bit-flip, and occasionally it can handle a third error—either because a phase shift or sign flip occurred twice to the same qubit resulting in the original state or the new state coincidentally decodes to the same data. When more than 2-3 errors occur, the data cannot be decoded.

Shor decode success rates with different types of noise

By looking at different types of noise, we can see whether one type is more damaging than the other. Different scenarios, such as wireless vs. wired connections, may produce a higher proportion of phase shifts or sign flips depending on the way the noise impacts upon on the system.



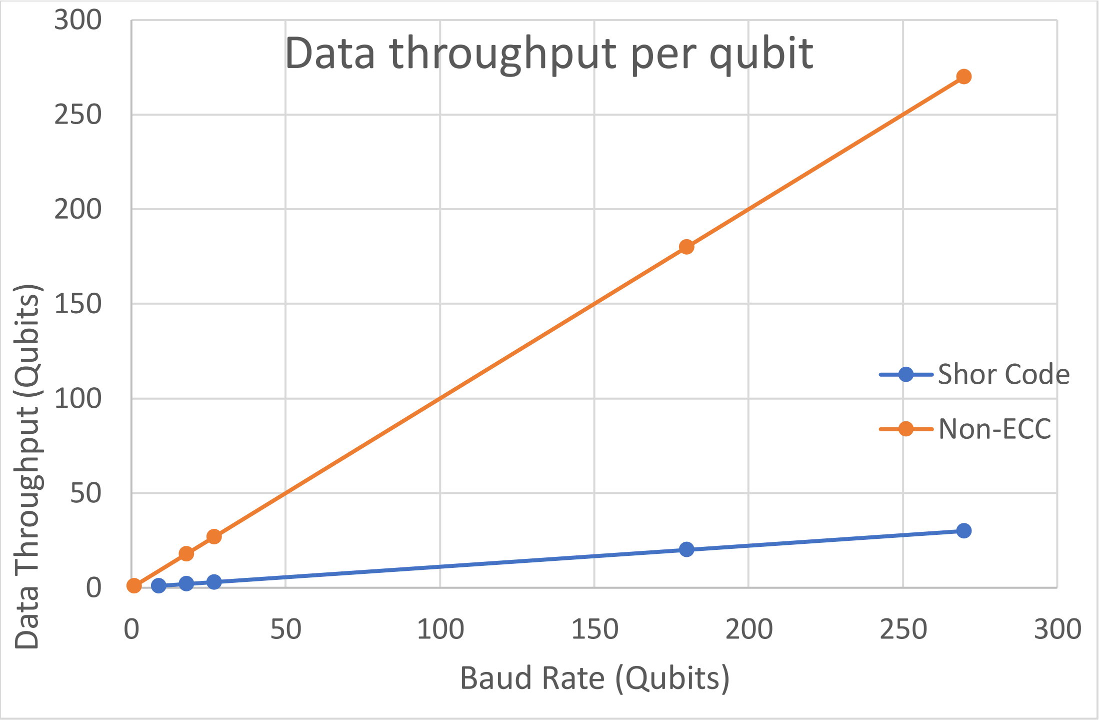
When looking at only one noise vector, i.e. depolarizing noise vs. dephasing noise, it shows that Shor code is much more tolerant to phase shifts and when considering only one type of noise then Shor code performs worse than when faced with both types of noise.

Shor code performs worse when subjected to only one type of noise because it uses 3 bits to track phase shifts and 3 bits to track sign flips, plus 3 qubits to track changes of both. However, if only one type of noise is present in the system it becomes effectively a 3-qubit code which can only handle only one perturbation on the system.

When subjected to dephasing noise, it performed better than when subjected to only depolarizing noise. This is because there are only two states that the polarity can be in, forwards and backwards spin. However, there are more states that can correspond to phase. The relationship with the 9-qubit Shor code and the decoded data is a many-to-one relationship, which means that there are multiple combinations of the 9 qubits that will decode to the same data. Therefore, sometimes no error correction is needed to still get the same data from a perturbed combination, and this is much more likely to occur with a phase shift.

Data throughput of Shor Code vs. Non-ECC in a perfect system

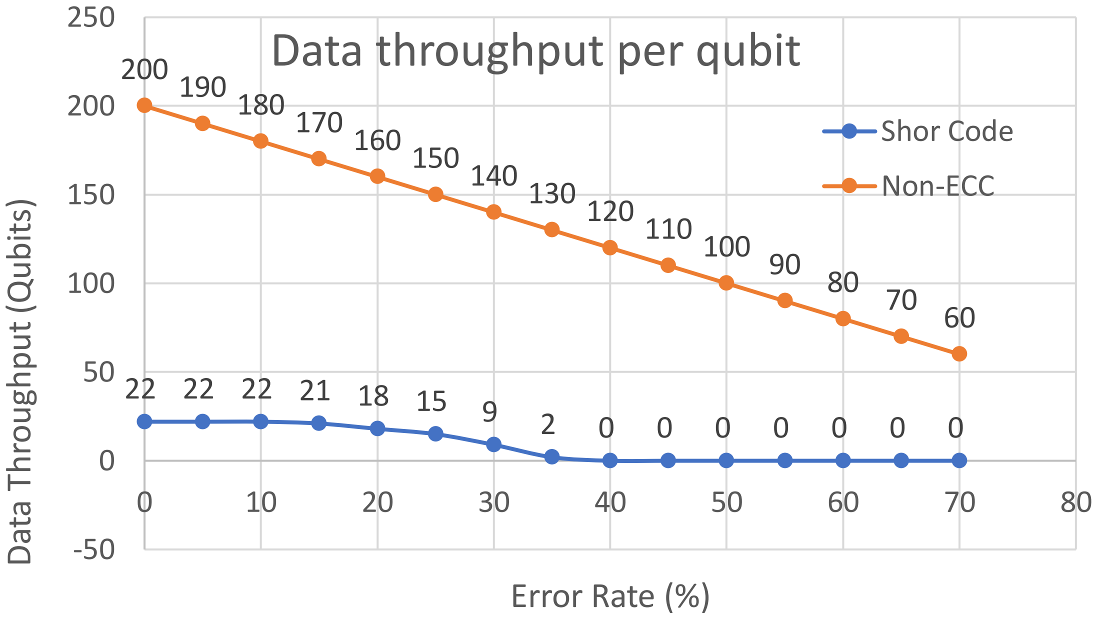
To demonstrate the performance impact that using Shor code has on bandwidth, the graph shows the loss of throughput due to encoding data with Shor code.



The gradient shows the ratio of number of bits transmitted per bit of usable data. Shor code requires much more (9 times) bandwidth to transmit less data. However, in noisy networks the tradeoff is worth it in order to get reliable communications.

Data throughput of Shor Code vs. Non-ECC in a noisy system

The graph shows the usable throughput of data, based on a bandwidth of 200 qubits and the standard noise model.



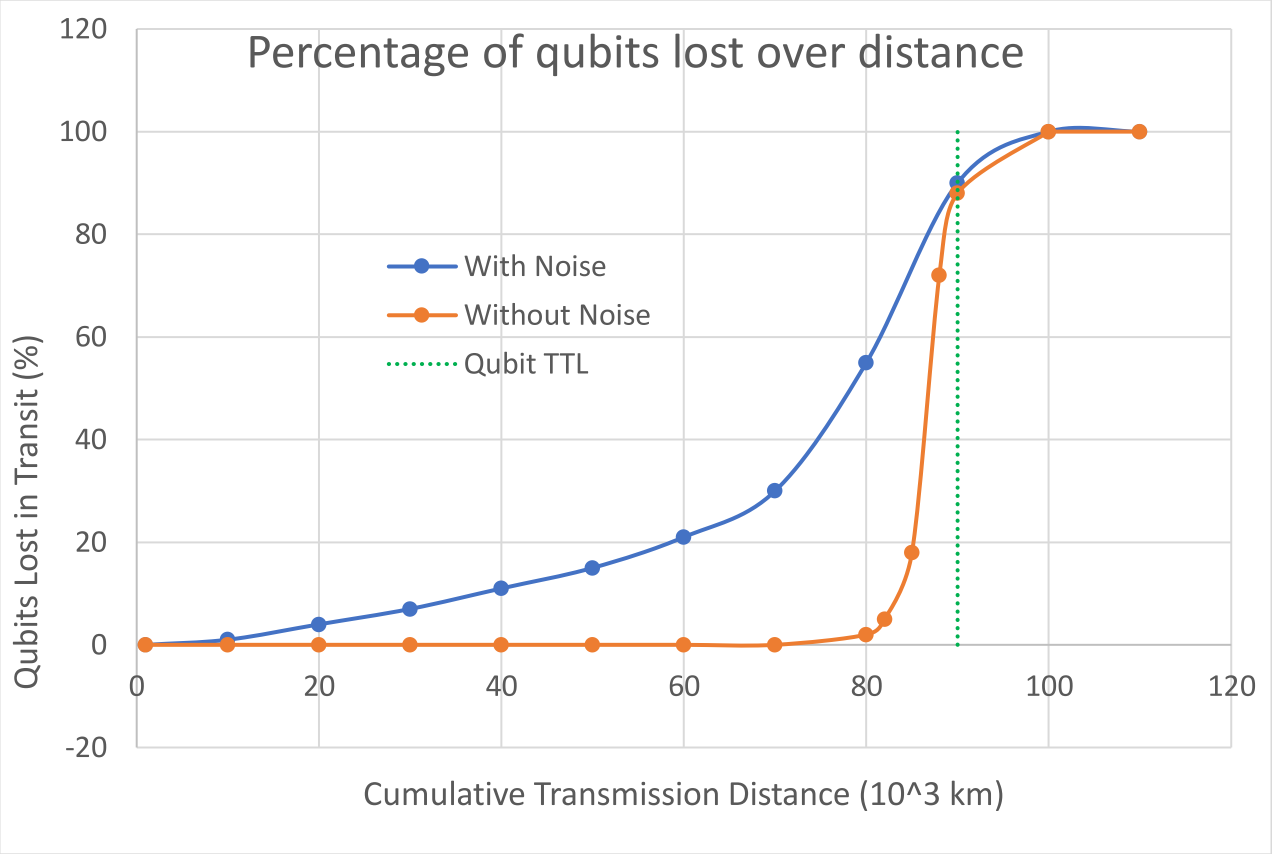
While Shor code has a much lower data rate, it is a tradeoff because it retains its fidelity in the presence of errors. Whereas, sending unencoded qubits through a noisy channel means the integrity of any received data cannot be guaranteed and loss is very high.

However, at a 40% error rate none of the data sent with Shor code could be decoded—but this at least guarantees the receiver knows the data was perturbed and an unusually error rate could be due to eavesdropping, which using Shor code will help to detect.

Transmission distance vs. qubit error rate

The graph demonstrates of the effect of distance (and therefore time) on transmission success rate. Qubits expire after a certain amount of time, and this will dictate how far apart repeaters must be.

The calculations are based on Silicon Quantum Dot technology, because they the qubit that is currently in-use by real quantum computers which most resembles a photon, which therefore makes some of the physical properties are easier to simulate.



Based on Silicon Quantum Dot technology, in a perfect system, with no noise and no variation in delays, every qubit would expire at 90,000 km.

Without noise in the system, the exact expiry time of the qubit varies due to delays and physical variance.

With noise in the system, every perturbation of a qubit reduces its lifespan, resulting in a higher loss rate over shorter distances. Based on a noise parameter of q=0.7 (a medium amount of noise), the noise causes 55% of the qubits to be lost early.

The inflection point comes at 70,000 km, because prior to this value the lost qubits were only as a result of noise, however, at this point both noise and TTL take effect rapidly increasing the loss of qubits.

Shor Code Quantum Repeater Delay

To experimentally test the delay time from decoding and re-encoding Shor code, along with the entanglement swap within a quantum repeater I implemented a repeater and tested it with an IBM-Q quantum computer.

Using 8,192 tests, it executed in a total time of 0.616294607 s. Therefore, the average delay caused by a quantum repeater using Shor code is 75.2312753 microseconds. This is an acceptable delay, and therefore the extra complexity does not have a detrimental effect on processing times.

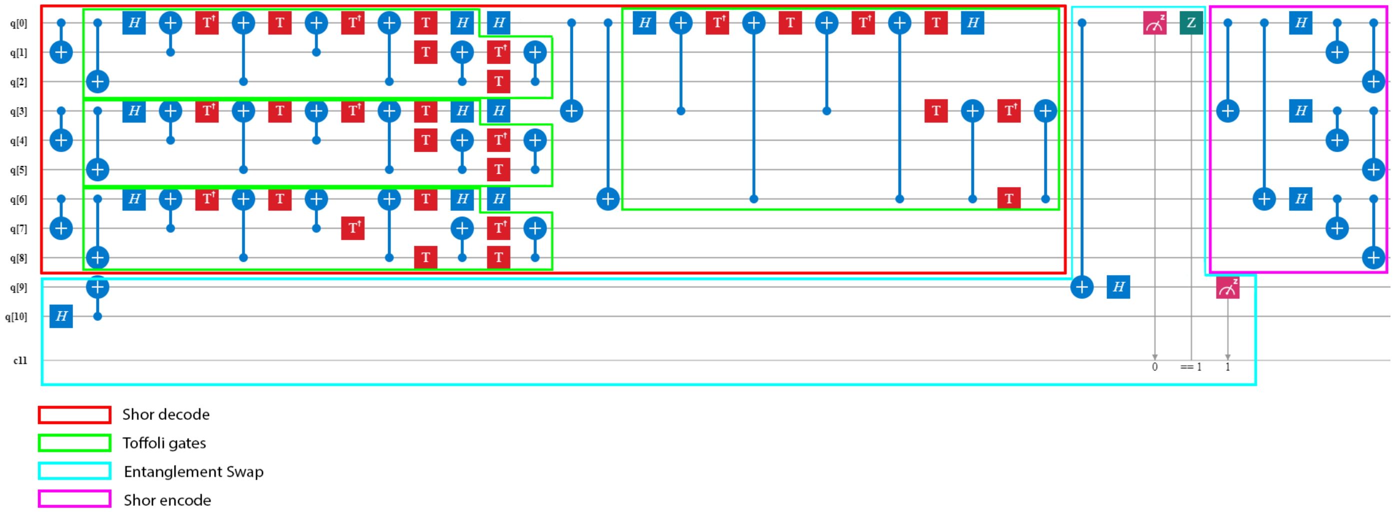
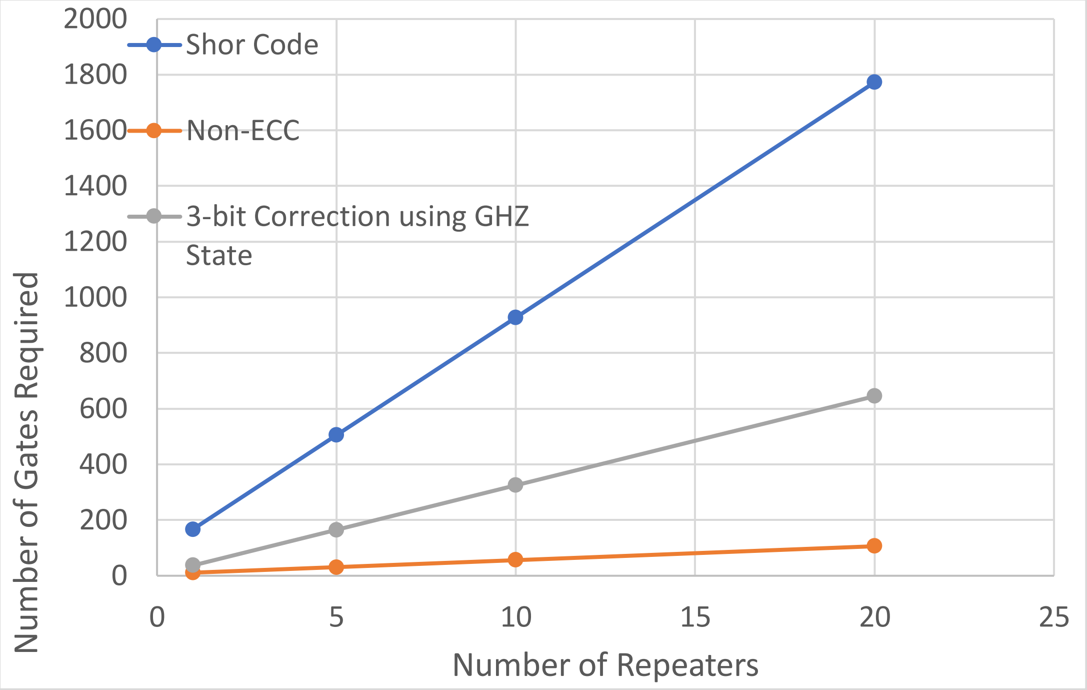


Figure 6: Quantum logic gate implementation of a quantum repeater

Shor code vs. non-ECC implementation complexity

This last graph shows the total gates required to implement the program, with x repeaters. It provides a visualization of the extra complexity required at each step. The sharp rise in gates required suggest that there may be another problem to look at, which is the reliability of the system. If a fault occurs in any of the 1,772 gates needed to implement 20 repeaters, then it could cause significant data loss.

When compared with the complexity of a 3-qubit system, it shows that even though Shor code has triple the number of qubits and can correct and detect multiple errors it packs a lot into a relatively small container.



VII. Conclusions

My results show that the success of Shor code is highly dependent on the type and amount of noise present in a system. It is highly successful at correcting the most common type of noise (phase shifts) and it is able to correct data with a reasonably high (up to 35%) error rate.

The actual amount of noise in a quantum network is hard to predict, as it changes based on the structure used as a qubit and it improves over time as new technology is developed to help quantum network transmission. It is reasonable to assume that the error rates will be below 35% in a production-ready network system, because current production QKD implementations require lower than 20-25% error rates—for which Shor code is very practical.

Furthermore, while Shor code requires many more gates to implement, the processing delay is still negligible. It can also be implemented mostly using gates that already exist in all quantum computers as they are required for producing entanglement. Based on Silicon Dot Technology, the qubits can still travel huge distances even during their short lifespan, and the 75 ms delay will therefore not hinder transmission at all. The distances it can travel even allow for reliable earth-to-space transmission, which usually occurs at distances of about 2,000 km.

Ultimately, I recommend using Shor code as a basis for error correction in quantum networks. It is fast, simple to implement and it can correct most errors within the normal threshold error rate.

VIII. Future Work

Throughout my experiments I looked at a range of noise levels, based on a statistical noise model to get a picture of overall performance of the error correction algorithm. Future work should look closely at the performance of Shor code under conditions closer related to production networks, such as the QKD networks that exist and are in-use already. By studying the networks, timedependence of noise can be modelled and a better proportion of dephasing noise vs. depolarizing noise can be used. Furthermore, the effects of busyness on the noise should be looked at to determine Shor code’s performance under heavy loads.

Loss should be investigated further, instead of using a basis state to fill in for lost qubits there may be a relationship between the successfully received qubits that can be used to create a more reliable replacement for lost qubits that would result in a higher rate of successful decoding.

There are four main technologies being used as the basis of qubits in current quantum computers, these should all be investigated to determine their feasibility in repeater-based networks. The qubits on offer range of a lifespan of 0.3 seconds to 1,000s. However, they also have vastly different levels of fidelity and some of the most stable require extra hardware for stability that makes them averse to any transmission.

References

[1] S. J. Devitt, W. J. Munro and K. Nemoto, "Quantum Error Correction for Beginners," IOP Publishing Ltd., Tokyo, 2013.

[2] E. M. Rains, R. H. Hardin, P. W. Shor and N. J. A. Sloane, "A Nonadditive Quantum Code," *Physical Review Letters,* vol. 79, no. 5, pp. 953-954, 1997.

[3] A. Dahlberg and S. Wehner, "SimulaQron—a simulator for developing quantum internet software," *Quantum Science and Technology,* vol. 4, no. 1, p. 015001, 2018.

[4] IETF, "Quantum Internet Proposed Research Group (qirg) - - IETF Datatracker," IETF, [Online]. Available: https://datatracker.ietf.org/rg/qirg/about/. [Accessed 08 2019].

[5] IBM, "IBM Q - Quantum Computing," IBM, [Online]. Available: https://www.research.ibm.com/ibm-q/. [Accessed 08 2019].

[6] S. Bose, V. Vedral and P. L. Knight, "Multiparticle generalization of entanglement swapping," *Physical Review A,* vol. 57, no. 2, pp. 822-829, 1997.

[7] D. Ghosh, P. Agarwal, P. Pandey, B. K. Behera and P. K. Panigrahi, "Automated error correction in IBM quantum computer and explicit generalization," *Quantum Information Procesing,* vol. 17, no. 6, p. 153, 2018.

[8] A. Zeilinger, M. A. Horne, H. Weinfurter and M. Żukowski, "Three-Particle Entanglements from Two Entangled Pairs," *Physical Review Letters,* vol. 78, no. 16, pp. 3031-3034, 1997.

[9] V. Lipinska, G. Murta and S. Wehner, "Anonymous transmission in a noisy quantum network using the W state," *PHYSICAL REVIEW A,* vol. 98, no. 5, pp. 052320-17, 2018.

[10] N. Kalb, A. A. Reiserer, P. C. Humphreys, J. J. W. Bakermans, S. J. Kamerling, N. H. Nickerson, S. C. Benjamin, D. J. Twitchen and M. H. R. Markham, "Entanglement distillation between solid-state quantum network nodes," *Science,* vol. 356, no. 6341, pp. 928-932, 2017.

[11] H. A. Haus, Electromagnetic Noise and Quantum Optical Measurements, New York: Springer, 2000.