Quantum Communication Complexity on Near-Term Networks

Solving the Equality Problem with Realistic Noise

Tom Jacobs[1]
Supervisor(s): Prof. Stephanie Wehner[1], Dr. Tzula Propp[1]
[1]Delft University of Technology, The Netherlands

January 26, 2025

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
January 26, 2025

Name of the student: Tom Jacobs
Final project course: CSE3000 Research Project
Thesis committee: Prof. Stephanie Wehner, Dr. Tzula Propp, Prof. Neil Yorke-Smith

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Abstract

Quantum computers allow us to solve certain problems that are unsolvable using classical computers. In this study we focus on solving the equality problem by simulating a three quantum computer network and using the communication complexity to determine if our theoretical quantum advantage is still there in practice. We want to know how the noise from realistic quantum networks that already exist affect this communication complexity. We found that we can beat the classical solution when simulating a laboratory setup in which the quantum computers are in close proximity to each other and when using only a small bit strings. However, when moving to setups in which there are kilometres between quantum computers instead of metres or when using larger bit strings as input to our problem we see that the noise becomes too much to simulate.

1 Introduction

In computer science, being able to solve a problem "fast" and being able to solve a problem at all often go hand in hand. Many problems have solutions, but finding these solutions for actual real-world scenarios takes an impractical amount of time (Sometimes longer than the history of the universe given our current computational power). Consider problems in the NP or NP-hard classes, such as the travelling salesman or the knapsack problem. Or consider cryptography: our current encryption is based almost entirely on the fact that classical computers are not capable doing prime factorization in a realistic amount of time. Quantum computers allow us to actually solve some of these problems in a realistic amount of time. These problems are part of the bounded-error quantum polynomial time class or BQP for short, which means we can solve them in polynomial time using quantum computers given we allow for a small error (Figure 1). Shor's algorithm [15] allowing quantum computers to quickly do prime factorization and possibly making our current encryption methods obsolete in the future is probably the most famous example of this.

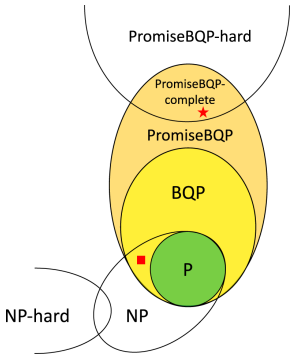


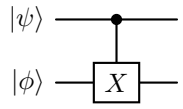
Figure 1: Different Problem Classes [12]

The main difference between quantum computers and classical computers is that quantum computers use qubits. Whereas classical bits can only be in one of two states (0 or 1) at a time, qubits can be in a superposition of different states and are therefore able to carry more information than classical bits. We denote a qubit state as follows: $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$. This means that this qubit is in a superposition of the states $|0\rangle$ and $|1\rangle$ and, when measured will produce either of the results with probability $|a|^2$ or $|b|^2$. Another possible notation used is the use of vectors to denote a certain state. In this case $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$ and $|1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$. Such that $|\psi\rangle = \begin{pmatrix}a\\b\end{pmatrix}$.

Just as classical computers work by applying gates to classical bits (e.g. AND, XOR and NOT gates), quantum computers work by applying gates to qubits. A very common gate is the Hadamard gate, which can be used to put a qubit in a superposition: $H\,|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $H\,|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Just as states can be denoted as vectors, gates can be denoted as matrices. The matrix corresponding to this gate looks as follows: $H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}$. Most gates in a quantum

circuit are depicted as a square, with a letter in it. The Hadamard gate looks like so: $\boxed{H}$.

A very common two-qubit gate is the Controlled-X or Controlled-NOT (or CNOT for short) gate depicted below, which works using two qubits and flips the second bit if and only if the state of the first bit is $|1\rangle$:



Given the states $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$ and $|\phi\rangle = c\,|0\rangle + d\,|1\rangle$. Before the CNOT the combined state looks like this: $|\Psi\rangle = |\psi\rangle \otimes |\phi\rangle = ac(|0\rangle \otimes |0\rangle) + ad(|0\rangle \otimes |1\rangle) + bc(|1\rangle \otimes |0\rangle) + bd(|1\rangle \otimes |1\rangle)$, where $\otimes$ is the tensor product. After having applied the CNOT gate the resulting state will be $|\Psi\rangle = ac(|0\rangle\,|0\rangle) + ad(|0\rangle\,|1\rangle) + bc(|1\rangle\,|1\rangle) + bd(|1\rangle\,|0\rangle)$. There are still tensor products between the individual states, only it is common to leave them out when notating quantum states, so I will also not show them in the rest of the paper. This state is entangled, because it cannot be written as two separate states like $|\psi'\rangle \otimes |\phi'\rangle$ any more (Try it!). There are four special entangled states called the Bell states, they are special because they are maximally entangled, which means that when you measure one of the qubits you are guaranteed to know the value of the other qubit. One of the Bell states is $|\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. We can collapse the state to either $|00\rangle$ or $|11\rangle$ by measuring either one of the qubits, at which point we know what the other one is. When we entangle two qubits and then separate them we can use this feature to transport information from one place to another using a process called teleportation. This nonlocality, meaning that objects are not only influenced by other objects in their direct surroundings was proven to be true in 2015 at the TU Delft [9].

Currently the Quantum Internet Alliance (QIA) is building quantum networks throughout Europe with Delft being one of the frontrunners. In this project we will look at how close these state-of-the-art quantum networks are to classical computer networks in solving the equality problem and find out what is currently possible or will be possible in the short term future. We want to know how what the effect is of noise from the current best quantum computers on our theoretical speed up, because we are currently in the noisy intermediate-scale quantum era (or NISQ-era). In quantum computers, memory has a lifetime, called the coherence time (or decoherence time) (t_c. This is because at any given moment qubits can lose the information they hold (or decohere), meaning they become useless. Error-correction algorithms to correct this decoherence exist, but current quantum computers are too small to run these, hence the name NISQ-era. Knowing the effect of the noise will allow us to get an idea of how close we are to getting an actual quantum advantage.

2 Formal Problem Description

2.1 The Equality Problem

In communication complexity problems, solutions are sought that require minimal communication between two or more parties. In [18] problems are studies in which three parties; Alice, Bob and a Referee, needed to compute a function $f(x, y)$. Alice and Bob own the two bit strings $x \in \{0,1\}^n$ and $y \in \{0,1\}^n$, while the Referee knows the function and needs to compute this while minimizing the amount of bits sent from Alice and Bob to the Referee. For certain communication complexity problems, there exist quantum solutions which have a lower complexity in theory than the best classical solutions. In this paper, we will be looking into the equality problem in which the function outputs 1 if and only if x equals y. That is:

$$f(x,y) = \begin{cases} 1, & \text{if } x = y \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

For this problem, the theoretical speed-up when using quantum computers in a quantum network is $O(\log n)$ [7]. In this study specifically, we will look at how noise in the quantum computers

and in the quantum network will impact this speed-up. We will simulate the problem using noise from current state-of-the-art quantum computers and will try to find the point at which the quantum network will beat the purely classical network.

2.2 Solving the Equality Problem using a Noiseless Quantum Network

2.2.1 Reducing the communication complexity

Trivially, the equality problem can always be done with $O(n)$ communication complexity if both of the parties just send their entire bit string to the Referee. However, more efficient solutions can be found if we allow for a small error ϵ. In this case it has been proven that the problem can be solved classically with $O(\sqrt{n})$ complexity [1]. This is the lowest complexity for this problem, unless both parties have access to a correlated random source [3].

Using qubits instead of classical bits the complexity can be brought down even more to $O(\log n)$, even without prior entanglement between qubits at Alice, Bob and the Referee [7]. Throughout the entire algorithm no communication between Alice and Bob is allowed. In this case, the algorithm goes as follows. First, two bit strings of length n will be encoded using an error-correction code into two longer bit strings of length $m = 2n + 1$, giving us $E(x) \in \{0,1\}^m$ and $E(y) \in \{0,1\}^m$. Using a Justesen code for this gives us a guarantee that if the original bit strings were different, then the resulting bit strings $E(x)$ and $E(y)$ have a Hamming distance of at least $(1 - \delta)m$, with $\delta < 9/10 + 1/15c$, and $c = m/n$ [7]. Justesen codes are also deterministic, so $E(x) = E(y)$ iff $x = y$.

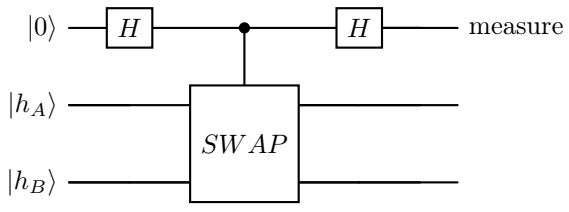
Then the two bit strings of length m need to be encoding into two smaller qubit strings. This is done using the following encoding [7]:

$$|h_x\rangle = \frac{1}{\sqrt{m}} \sum_{i-1}^{m} |i\rangle \, |E_i(x)\rangle \tag{2}$$

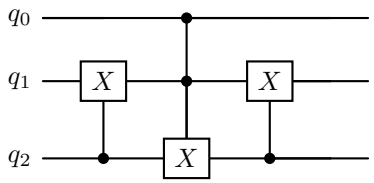
Where $E_i(x)$ is the ith bit of the bit string $E_i(x)$ and $|i\rangle$ is the state corresponding to the binary representation of i. This means that the final qubit string will be of length $log_2(m)+1$ or $log_2(2n+1)+1$. Finally Alice and Bob send their qubit strings to the Referee using teleportation. In teleportation two qubits are entangled. One of which is then send from one computer to the other and used to send over the required state. This resource is then consumed, meaning that for each teleportation attempt two new qubits must be entangled. Depending on the measurement results, some correction may need to be applied on the receiving end. The information on whether or not this correction needs to be applied, and specifically which one can be sent, using 2 classical bits of information. Meaning that each time a single qubit is being sent, we count this as 3. One for the qubit and 2 for the 2 correction bits.

2.2.2 The CSWAP gate

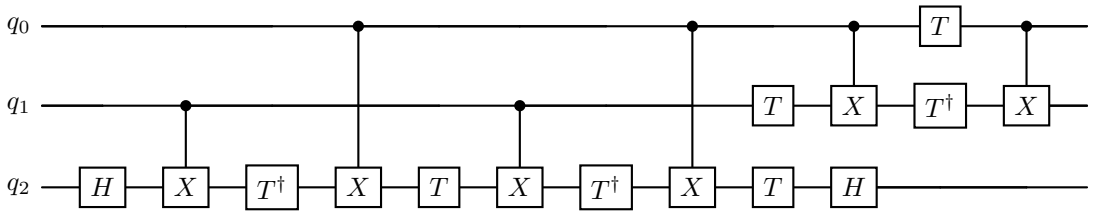
In order to determine whether two qubit strings $|h_A\rangle$ and $|h_B\rangle$ are equal we use the following quantum circuit [7]:



There is however one issue with this circuit, which is the fact there doesn't exist a gate which in itself does a controlled SWAP on 2 qubit strings of arbitrary length. Just like classical computers only implement basic gates like XORs, quantum computers have their own set of basic gates. We need to deconstruct this gate into only single and 2-qubit gates, which are prevalent in the SquidASM library. In Figure 5 in [8] it is shown that a CSWAP gate, which swaps 2 qubits can be deconstructed into three gates, namely two CNOT gates and a Toffoli (or CCNOT gate) which together form the circuit below:



However, even this circuit cannot yet be directly implemented. Since the Toffoli gate is not a single or 2-qubit gate we need to decompose it further. In Figure 1 in [14] it is shown that a Toffoli gate can be broken down into only single and 2-qubit gates like so:



Finally, the CSWAP gate for swapping 2 qubits is deconstructed into only single and 2-qubit gates. There is however one problem still, namely that T-gates (and by extension also $T^\dagger$-gates) aren't implemented in SquidASM. When we look at the matrix representation of these gates, they look as follows:

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{pmatrix}, \quad T^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{-i\pi}{4}} \end{pmatrix}$$

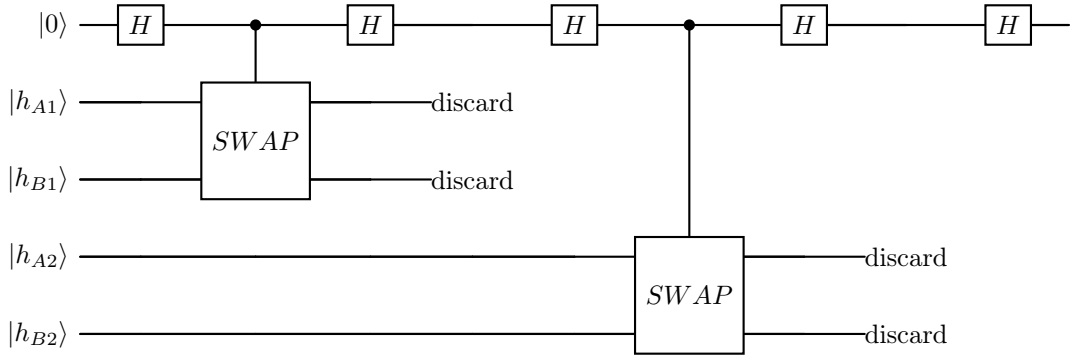
Luckily for us this is equal (up to a global phase) to just a rotation around the Z-axis with a specific angle:

$$R_z(\theta) = \begin{pmatrix} e^{\frac{-i\theta}{2}} & 0 \\ 0 & e^{\frac{i\theta}{2}} \end{pmatrix}$$

$$R_z(\frac{\pi}{4}) = \begin{pmatrix} e^{\frac{-i\frac{\pi}{4}}{2}} & 0 \\ 0 & e^{\frac{i\frac{\pi}{4}}{2}} \end{pmatrix} = \begin{pmatrix} e^{\frac{-i\pi}{8}} & 0 \\ 0 & e^{\frac{i\pi}{8}} \end{pmatrix} = e^{\frac{-i\pi}{8}} \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{pmatrix} \simeq T$$

$$R_z(-\frac{\pi}{4}) = \begin{pmatrix} e^{\frac{i\frac{\pi}{4}}{2}} & 0 \\ 0 & e^{\frac{-i\frac{\pi}{4}}{2}} \end{pmatrix} = \begin{pmatrix} e^{\frac{i\pi}{8}} & 0 \\ 0 & e^{\frac{-i\pi}{8}} \end{pmatrix} = e^{\frac{i\pi}{8}} \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{-i\pi}{4}} \end{pmatrix} \simeq T^\dagger$$

With this final substitution we can now implement a controlled SWAP for 2 qubits in SquidASM. However we don't want to swap just 2 qubits, but 2 strings of qubits. Scaling this up is not too difficult however, because we can simply use a single CSWAP for the first qubits of the strings and then a second CSWAP for the second qubits of each string, all the way until we have done all the qubits. When going through all these CSWAP gates we are entangling the two states that Alice and Bob sent through our measurement qubit. Keeping track of entangled states involving many qubits requires keeping very large matrices in memory. This is very memory-intensive. In order to keep these matrices more manageable, we can discard qubits when we don't need them any more. So, after every CSWAP we can discard the 2 qubits we just swapped, but because these are still entangled with other qubits, discarding these can potentially change the outcome of the circuit, to prevent this, we must apply a Hadamard gate to the measurement qubit before discarding the two qubits. After we discarded the two qubits we apply another Hadamard and continue as if nothing happened. A part of the circuit then looks like this:



2.2.3 Interpreting the Noiseless Measurement Results

If $|h_A\rangle$ and $|h_B\rangle$ are the same then measuring the first qubit after this circuit will always result in a 0, however if $|h_A\rangle$ and $|h_B\rangle$ are not the first qubit will be in a superposition of the two states $|0\rangle$ and $|1\rangle$. And measuring it can now result in a 1 with probability $(1-\delta^2)/2$ [7]. In this project we use randomly generated bit strings for Alice and Bob. Given two random strings of length m the expected Hamming distance is $0.5m$, so we can estimate the probability of measuring a 1, given that the two strings were not the same to be $(1-0.5^2)/2 = 0.375$.

It is clear that with one measurement it is impossible to know whether or not the two original bit strings were the same or not. In fact, if one keeps getting 0s then we can never know for sure if they are the same or we are just getting a very improbable streak. For this project we want to be 99.9999999% sure that we determine whether the two strings were equal or not. Our error rate should be $\epsilon < 10^{-9}$. In order to achieve this Alice and Bob must create their respective states multiple times and send them to the Referee. The Referee then creates a measurement record. If we measure a 1, we can immediately report that the two strings were not the same. If we keep measuring 0s however, we need to repeat this n times to be sure enough that they are the same. Where we can find n using the following formula: $(1-0.375)^n < 10^{-9}$. Which results in $n > 44.09$, so we must repeat the experiment 45 times.

	$P(0)$	$P(1)$
The same	1.0	0.0
Different	0.675	0.375

Table 1: The estimated noiseless measurement probabilities

2.3 Simulating Noise

2.3.1 Realistic Noise

Unfortunately, real quantum computers don't generate perfect results like this. Multiple imperfection parameters have been determined in quantum computer, which together significantly reduce the efficiency of these algorithms [17]. In this study, we will specifically look at coherence time t_c, fidelity of the Bell states F_b, and the fidelity of single and 2-qubit operations $F_{1/2}$. Fidelity simply means how close the actual state is to the theoretical state we expect. In the noiseless case, all fidelities were 1. Now, however they are somewhere between 0 and 1. The Bell states are necessary to teleport qubits from one quantum computer to another. They are states in which 2 qubits are maximally entangled. Single qubit gates refer to gates that only act on single qubits, while 2-qubit gates refer to gates that act on 2 qubits, which in our circuit is only the CNOT gate.

Coherence time might need a bit more explanation. Coherence time is sometimes also referred to as decoherence time. In a quantum computer, there always exists the chance that a qubit loses its prepared state and becomes useless. The coherence time is the time during which we can safely assume that the qubit retains its state. However, because this is probabilistic it is always possible that a qubit has already decohered and is useless while we still need it. In order to prevent this,

there exist quantum error-correction algorithms which are able to reinstate qubits in their required state. We will not use these algorithms during this project.

Each of these parameters influences the probability of getting a wrong measurement [13]:

$$\begin{gathered}\text{Probability of a single 1-qubit gate failing: } P_{1-qubitgate} = \frac{2}{3}(1 - \sqrt{F_{1-qubitgate}}) \\ \text{Probability of a single 2-qubit gate failing: } P_{2-qubitgate} = \frac{4}{5}(1 - \sqrt{F_{2-qubitgate}} \\ \text{Probability of a getting a wrong Bell state: } P_{teleport} = \frac{4}{5}(1 - \sqrt{F_{Bellstate}} \\ \text{Probability of a single qubit decohering: } P_{decohere} = 1 - e^{-\frac{w}{t_c}}\end{gathered} \tag{3}$$

With w begin the window during which the qubit should be active. This includes the time it takes to teleport, as well as the time it takes for the Referee to complete the CSWAP circuit. All these together decide the probability of something going wrong in the entire process. If something goes wrong we cannot say what the outcome will be, so we assume we get a fifty-fifty chance to either get a 0 or a 1.

$$P_{50-50} = 1 - (1 - P_{decohere})^{2N_q+1}(1 - P_{1-qubitgate})^{N_1}(1 - P_{2-qubitgate})^{N_2}(1 - P_{teleport})^{2N_q} \tag{4}$$

With N_q being the amount of qubits Alice and Bob send to the Referee (so together they send $2N_q$ qubits). And N_1 and N_2 being the total amount of 1- and 2-qubit gates respectively. With this probability we can now estimate the new measurement probabilities:

	$P(0)$	$P(1)$
The same	$(1 - P_{50-50}) + 0.5P_{50-50}$	$0.5P_{50-50}$
Different	$0.675(1 - P_{50-50}) + 0.5P_{50-50}$	$0.375(1 - P_{50-50}) + 0.5P_{50-50}$

Table 2: The estimated noisy measurement probabilities

2.3.2 Interpreting the Noisy Measurement Results

When introducing noise, the probabilities to get a 0 or a 1 for our measurement change to make it more difficult to determine whether the bit strings are the same or different. In order to know whether two bit strings were the same or not, we will again create a measurement record $M \in \{0, 1\}^x$, where x is the amount of times we have already repeated the simulation. We then use the probabilities from Table 2 to determine the probabilities that this measurement record is the result of two bits strings that are the same or different:

$$\begin{aligned}P(M \mid \textit{The same}) &= P(0 \mid \textit{The same})^{n_0} \times P(1 \mid \textit{The same})^{n_1} \\ P(M \mid \textit{Different}) &= P(0 \mid \textit{Different})^{n_0} \times P(1 \mid \textit{Different})^{n_1}\end{aligned} \tag{5}$$

Where n_0 is the amount of 0s in M and n_1 is the amount of 1s in M. Then we can use Bayes' theorem to find the probability of the two strings being the same or different given our current measurement record:

$$\begin{aligned}P(\textit{The same} \mid M) &= \frac{P(M \mid \textit{The same})P(\textit{The same})}{P(M)} \\ P(\textit{Different} \mid M) &= \frac{P(M \mid \textit{Different})P(\textit{Different})}{P(M)}\end{aligned} \tag{6}$$

In the study we randomly decide whether we use the same bit string or two different ones with 50% probability each, so $P(\textit{The same}) = P(\textit{Different}) = 0.5$. And $P(M) = P(M \mid \textit{The same}) \times P(\textit{The same}) + P(M \mid \textit{Different}) \times P(\textit{Different})$. Now, if either $P(\textit{The same} \mid M)$ or $P(\textit{Different} \mid M)$ is larger than $\epsilon = 10^{-9}$ we can return our answer. If not, we do another round of the simulation and add that result to our measurement record, after which we will again check if we can now safely say whether the two bit strings were the same or not.

3 Experimental Setup and Results

3.1 Classical Solution

The scaling of the classical solution has been determined to be $\sqrt{3n} + O(\sqrt{n})$ in [1]. We will multiply this by 2, because we receive bits from 2 computers, and get $2\sqrt{3n}$ as our scaling. And the probability of getting the correct result is $\frac{6}{11}$ [1].

3.2 The Simulation

The code for the simulation is written in Python version 3.10.6 and makes heavy use of the library SquidASM, which in turn relies on NetSquid. The code can be found on Github.

3.3 Network with Trapped Ions in a Laboratory

We run the simulation using two different sets of parameters. The first set is to mimic a setup in a laboratory. This means that the distance between the quantum computers in the network is small (meters) and the parameters are generally better. For the quantum computers themselves we will use trapped-ion quantum devices just like in [17]. The state-of-the-art parameters for such a network are:

Parameter	**Value**
$F_{1-qubitgate}$	0.999934 [11]
$F_{2-qubitgate}$	0.9991 [11]
$F_{Bellstate}$	0.990025 [5]
t_c	62 ms [17]
$P_{success}$	0.0087 [17]
$t_{attempt}$	16.95 μs [17]

Table 3: The state-of-the-art parameters for a laboratory scale network using trapped ions.

The final two parameters are the probability of any given entanglement succeeding and the time each attempt takes. These alone don't influence the probabilities, but together with the coherence time can have a great influence on the chances of the circuit succeeding. With these parameters we are now ready to run the simulation and we get the following results:

From this graph it follows that for bit strings of length up to $n = 127$, we can beat the classical solution. However, we do get closer to the classical solution as the amount of qubits increases. Unfortunately, from $n = 128$ onward we require more qubits than the simulation is capable of running. With $n = 127$, we create qubit strings of 9 qubits, meaning our Referee needs to handle $2 \times 9 + 1 = 19$ qubits. More powerful computers may allow us to simulate a couple more qubits, but even then it is unlikely we will be able to simulate bit strings larger than $n = 4095$. If we want to know whether we have an actual advantage, we would need to be able to simulate bit strings of roughly $n = 300000$, because around that point the classical solution overtakes the trivial solution of just sending the entire bit string.

In order to get a better sense of which parameters cause the most noise, we run the simulation again. Only now we set all the parameters to perfect except for one. Then we find value for this parameter where the quantum solution is equal to the classical solution. We run the simulation using $n = 31$ and we keep $P_{success} = 0.0087$ and $t_{attempt} = 16.95$ μs when finding the minimum value for for the decoherence time.

Parameter	**Minimum required value**	**State of the art value**
$F_{1-qubitgate}$	0.978	0.999934 [11]
$F_{2-qubitgate}$	0.979	0.9991 [11]
$F_{Bellstate}$	0.930	0.990025 [5]
t_c	58 ms	62 ms [17]

Table 4: The state-of-the-art values compared with the minimum required values.

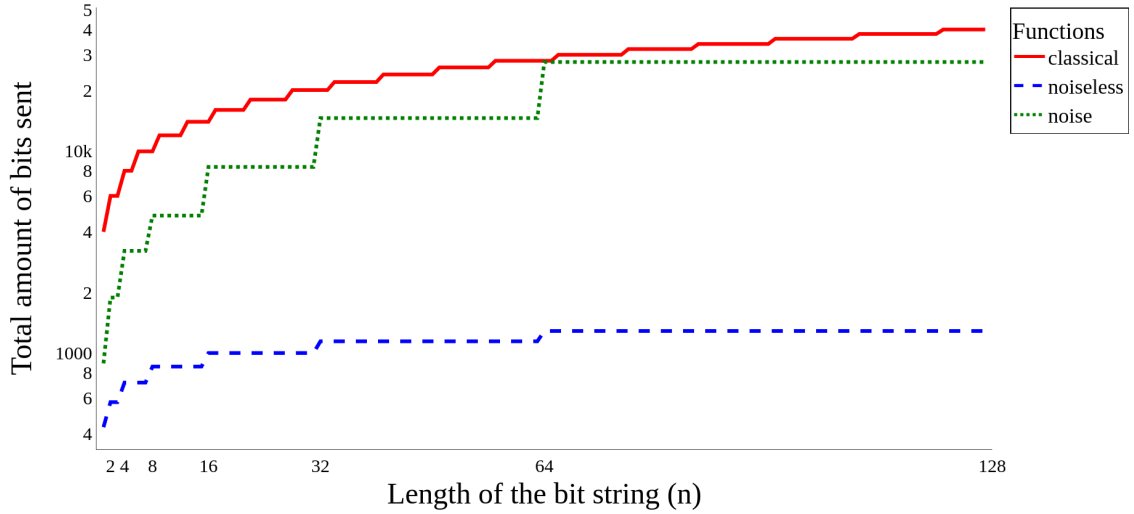


Figure 2: The results from running the simulation using the parameters of a network involving trapped-ion quantum devices in a laboratory. It shows that for small bit strings the noisy simulation (green dotted line) beats the classical solution (red line), despite being a lot worse than the noiseless quantum solution (blue dashed line). The noisy simulation does appear to scaling faster than the classical solution. This is because the noise quickly scales with the amount of qubits and because it gets harder and harder to teleport all the qubits before the first ones have decohered.

From this test we can deduct that the coherence time t_c is the biggest creator of noise in our system. All of the other parameters are already super close to their theoretical maximimum (1). In order to confirm this we ran anther test, wherein we estimated the P_{50-50} from our measurement results. We again run the simulation by setting all parameters to perfect, except for the one we want to find P_{50-50} for, which we set to its state-of-the-art value, again using $n = 31$:

Parameter	**Value**	P_{50-50}
$F_{1-qubitgate}$	0.999934 [11]	0.04
$F_{2-qubitgate}$	0.9991 [11]	0.05
$F_{Bellstate}$	0.990025 [5]	0.07
t_c	62 ms [17]	0.42

Table 5: Estimated P_{50-50} for the state-of-art values for each parameter, given that the others are perfect. It is now clear that most of the noise comes from the coherence time. Which includes both the time it takes for teleportation as well as for the CSWAP circuit.

3.4 Network with NV-centers in between Cities

For our second test we use the parameters from an actual real world experiment from [16]. This is a network between Delft and The Hague with nitrogen-vacancy centers as its quantum devices. The distance between the quantum computers here is about 25 kilometres, while in the previous setup is was just meters. The parameters for this network are:

Parameter	Value
$F_{1-qubitgate}$	0.999 [13]
$F_{2-qubitgate}$	0.97 [2]
$F_{Bellstate}$	0.965 [10]
t_c	687 μs [4]
$P_{success}$	0.0000072 [16]
$t_{attempt}$	200 μs [16]

Table 6: The parameters for a metropolitan scale network using NV-centers.

It is clear that these parameters are a lot worse than those from the previous paragraph. Partially this is because NV-centers are just worse than trapped ions, but mainly because the network is a lot bigger and a lot more can go wrong. It is clear that with these parameters it becomes very hard to get multiple qubits sent to the Referee without the first one decohering. In fact, the minimal amount of qubits in a string is equal to $log_2(2 \times 1 + 1) + 1 = 3$. With 200 μs per attempt, and with a coherence time of 687 μs it is technically possible to get a definitive answer, but the probability of it happening is very low. So low, in fact that all the results from the simulation are basically just random noise.

4 Responsible Research

Quantum computers can be a very powerful tool. It is scary to know that all our current encrypted data can be decrypted using quantum computers, however this does not mean that we should not research quantum computers. As long as we proceed with caution and acknowledge the capabilities of our quantum computers we can make sure to stay aware of their dangers and find solutions to potential risks. For instance, the ability to break our current encryption has already sparked a new area of research called post-quantum cryptography in which researchers try to find encryption methods which cannot be broken by quantum algorithms. And in [6] among others they have found one possible solution.

The fact that quantum computers can be such a powerful tool is only more of an argument to do build them and to research them. We have only scratched the surface of their capabilities and if we are to learn from the development of classical computers their capabilities are probably being underestimated. We, as humanity, stand for a plethora of incredibly difficult problems in which quantum computers can potentially aid us to find solutions.

5 Conclusion

We set out to see how close current real world quantum networks are to the classical computer networks that are everywhere around us. We did this by attempting to solve the equality problem in a setup with three quantum computers in a network. By using a noise model and noise parameters which mimic current state-of-the-art quantum computers, we can see what is currently possible. We can use the communication complexity of the solution to see with how much noise we can still beat the classical solution.

We have seen that in ideal cases we are already capable of beating classical networks on a small scale. However, we can also see that the noise our current best setups have increases quickly as the amount of qubits increases. Though in this study we didn't see if and when the noisy quantum simulation eventually loses to the classical solution, using more powerful computers to simulate this quantum network may allow us to add on a few more qubits, but the fact that even with a relatively small amount of qubits the simulation becomes very hard to run already points towards the quantum solution losing out to the classical solution in the long run. Also, when using quantum networks that cover any meaningful distance we can see that sending over multiple entangled qubits is currently very hard, if not impossible. This is in line with other results from the NISQ-era. On a small scale we can make things work, but if we want to get to a point where we can get a real quantum advantage we still have a long way to go.

From the parameters we used; coherence time, fidelity of the Bell states, single qubit gate fidelity and two qubit gate fidelity, we found that coherence time is by far the most important one. This, together with the difficulty of getting two entangled qubits in different places, accounts for most of the noise. So much even that for networks that are bigger than a single laboratory we are unable to solve the equality problem at all. Again, this is in line with the NISQ-era. Where in the future maybe quantum error-correction algorithms can help us mitigate the effects of decoherence this is not yet possible.

References

[1] Andris Ambainis. "Communication Complexity in a 3-Computer Model". In: *Algorithmica* 16 (Sept. 1996), pp. 298–301. DOI: 10.1007/BF01955678.

[2] Guus Avis et al. "Requirements for a processing-node quantum repeater on a real-world fiber grid". In: *npj Quantum Information* 9.1 (Oct. 2023), p. 100. ISSN: 2056-6387. DOI: 10.1038/s41534-023-00765-x. URL: https://doi.org/10.1038/s41534-023-00765-x.

[3] L. Babai and P.G. Kimmel. "Randomized simultaneous messages: solution of a problem of Yao in communication complexity". In: *Proceedings of Computational Complexity. Twelfth Annual IEEE Conference*. 1997, pp. 239–246. DOI: 10.1109/CCC.1997.612319.

[4] H. Bernien et al. "Heralded entanglement between solid-state qubits separated by three metres". In: *Nature* 497.7447 (May 2013), pp. 86–90. ISSN: 1476-4687. DOI: 10.1038/nature12016. URL: https://doi.org/10.1038/nature12016.

[5] Matthias Bock et al. "High-fidelity entanglement between a trapped ion and a telecom photon via quantum frequency conversion". In: *Nature Communications* 9.1 (May 2018), p. 1998. ISSN: 2041-1723. DOI: 10.1038/s41467-018-04341-2. URL: https://doi.org/10.1038/s41467-018-04341-2.

[6] Joppe Bos et al. *CRYSTALS – Kyber: a CCA-secure module-lattice-based KEM*. Cryptology ePrint Archive, Paper 2017/634. 2017. DOI: 10.1109/EuroSP.2018.00032. URL: https://eprint.iacr.org/2017/634.

[7] Harry Buhrman et al. "Quantum Fingerprinting". In: *Phys. Rev. Lett.* 87 (16 Sept. 2001), p. 167902. DOI: 10.1103/PhysRevLett.87.167902. URL: https://link.aps.org/doi/10.1103/PhysRevLett.87.167902.

[8] Raoul Heese, Patricia Bickert, and Astrid Elisa Niederle. "Representation of binary classification trees with binary features by quantum circuits". In: *Quantum* 6 (Mar. 2022), p. 676. ISSN: 2521-327X. DOI: 10.22331/q-2022-03-30-676. URL: http://dx.doi.org/10.22331/q-2022-03-30-676.

[9] B. Hensen et al. "Loophole-free Bell inequality violation using electron spins separated by 1.3 kilometres". In: *Nature* 526.7575 (Oct. 2015), pp. 682–686. ISSN: 1476-4687. DOI: 10.1038/nature15759. URL: https://doi.org/10.1038/nature15759.

[10] Bas Hensen. "Quantum Nonlocality with Spins in Diamond". Dissertation (TU Delft). PhD thesis. Apr. 2016. ISBN: 978-90-8593-253-6. DOI: 10.4233/uuid:b307dca0-79a4-4cc9-af91-a2927a61088e. URL: https://doi.org/10.4233/uuid:b307dca0-79a4-4cc9-af91-a2927a61088e.

[11] C. W. Hogle et al. "High-fidelity trapped-ion qubit operations with scalable photonic modulators". In: *npj Quantum Information* 9.1 (July 2023), p. 74. ISSN: 2056-6387. DOI: 10.1038/s41534-023-00737-1. URL: https://doi.org/10.1038/s41534-023-00737-1.

[12] Jonas JÃ€ger and Roman Krems. "Universal expressiveness of variational quantum classifiers and quantum kernels for support vector machines". In: *Nature Communications* 14 (Feb. 2023). DOI: 10.1038/s41467-023-36144-5.

[13] Tzula Propp. *Personal conversation*.

[14] Vivek V. Shende and Igor L. Markov. *On the CNOT-cost of TOFFOLI gates*. 2008. arXiv: 0803.2316 [quant-ph]. URL: https://arxiv.org/abs/0803.2316.

[15] Peter W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM Journal on Computing* 26.5 (1997), pp. 1484–1509. DOI: 10.1137/S0097539795293172. eprint: https://doi.org/10.1137/S0097539795293172. URL: https://doi.org/10.1137/S0097539795293172.

[16] Arian J. Stolk et al. "Metropolitan-scale heralded entanglement of solid-state qubits". In: *Science Advances* 10.44 (2024), eadp6442. DOI: 10.1126/sciadv.adp6442. eprint: https://www.science.org/doi/pdf/10.1126/sciadv.adp6442. URL: https://www.science.org/doi/abs/10.1126/sciadv.adp6442.

[17] J. van Dam et al. "Hardware requirements for trapped-ion-based verifiable blind quantum computing with a measurement-only client". English. In: *Quantum Science and Technology* 9.4 (2024). ISSN: 2058-9565. DOI: 10.1088/2058-9565/ad6eb2.

[18] Andrew Chi-Chih Yao. "Some complexity questions related to distributive computing(Preliminary Report)". In: *Proceedings of the eleventh annual ACM symposium on Theory of computing* (1979). URL: https://api.semanticscholar.org/CorpusID:999287.