A test suite for quantum networks

Assessing an application's effectiveness as a benchmark for quantum networks

Casper Dekeling[1]

Supervisor(s): Stephanie Wehner[1]**, Ravisankar Ashok Kumar Vattekkat**[1]

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 25, 2023

Name of the student: Casper Dekeling
Final project course: CSE3000 Research Project
Thesis committee: Stephanie Wehner, Ravisankar Ashok Kumar Vattekkat, Andy Zaidman

Abstract

In the development of any new technology, it is essential to have methods to assess the quality of a system, to compare different systems to one another, and to compare different versions of the same system in order to see if changes to the system can actually be classified as improvements. For quantum networks, this is no different. To further develop quantum networks, we need a benchmarking suite to judge the quality of such systems.

The aim of this research is to determine the effectiveness of blind quantum computation - a quantum network algorithm - as a benchmark for a quantum network. We determine what changes to the system affect the results of executing this application, and we use this to determine whether it would be effective to use this application as part of a larger benchmarking suite. We do this by simulating a quantum network, and manually varying system parameters one by one, to see if they have an effect on the results.

What we observed from these experiments is that blind quantum computation is sensitive to almost all system parameters. This means that introducing an imperfection into almost any part of the system will negatively affect the results of the application. This means the application is useful as a full-system benchmark, because it is affected by almost the entire system. However, it also means that the application is less useful as a benchmark for individual parameters.

This means that to make the most useful benchmarking suite, blind quantum computation would have to be combined with other quantum network applications that are more suitable for benchmarking individual system parameters.

1 Introduction

Quantum networks have numerous potential benefits, such as more secure communication [1], facilitating distributed quantum computation [2], and clock synchronisation [3], which make their development a vital research area. However, this research is still in the early stages, and we are far from realising the full potential of quantum network technology [4]. Nevertheless, numerous such systems are currently being developed, and continuous improvements are being made [5] [6] [7].

As is the case with all novel technologies, the rapid advancement of quantum computers has been accompanied by major progression in the ability to assess their quality, and to compare different quantum computers [8]. Thus, with the current developments in the field of quantum networks, it is essential that similar progress is made in the corresponding problem of benchmarking and comparing quantum network systems. This problem has unique features that cannot be found in the benchmarking methods for quantum computers, such as the testing of quantum communication links, which are not present in quantum computers. Several methods to benchmark quantum networks already exist [9], however as the technology develops, these benchmarking systems will also have to develop.

In this work, we aim to contribute to this need by researching a quantum network application, and performing experiments to see if this application could be used as a benchmarking tool. Specifically, the application we research is blind quantum computation, an application that allows a server to perform a quantum computation on behalf of a client, without knowing exactly what calculations are being performed. We discuss this application's usefulness in benchmarking certain properties of the quantum network, such as entanglement fidelity, quality of quantum operations, and memory lifetimes of the individual qubits. We discuss the extent to which our application can detect errors in these properties, and how sensitive the application is to such errors.

Section 2 will describe the methods used to get our results. This section is followed by a more detailed description of the setup of the experiments in section 3. Section 4 discusses the reproducibility of the research. Section 5 shows the results of all performed experiments, and observations about these results. This is followed by section 6, which provides more in depth discussion about the previously shown results. Section 7 talks about further research that could be done. Lastly, section 8 repeats the research question and provides the final conclusions drawn from the research.

For readers unfamiliar with quantum computing, Appendix A gives some background information on quantum computing to help understand this paper.

2 Methodology

This section explains the application used for benchmarking and how we test its effectiveness as a benchmark.

2.1 The application

To develop this test suite, we start by exploring a quantum network application's effectiveness at detecting imperfections in system parameters, and determining its sensitivity to these errors. Specifically, we assess these qualities for blind quantum computation. Blind quantum computation is a quantum network application with two parties: a client and a server. The server performs an effective computation for the client, without knowing exactly what calculation is being performed.

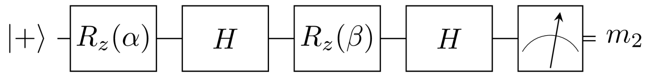

Figure 1: Effective computation performed by the server. Image taken from https://github.com/QuTech-Delft/squidasm/blob/develop/examples/stack/bqc/README.md

Figure 1 shows the effective computation performed by the system. Angles α and β are parameters chosen by the client, and should not be known to the server. The complete circuit that implements this application can be seen in figure 2.

The figure shows that the server does not receive angles α and β from the client, but instead two obfuscated parameters dependent on α, β, θ_1, θ_2 and measurements performed

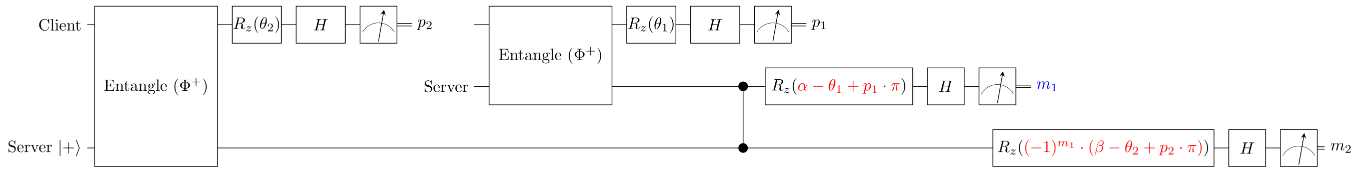

Figure 2: Actual implementation of the effective computation. Blue values are sent from server to client. Red values are computed by the client and sent to the server. Image taken from https://github.com/QuTech-Delft/squidasm/blob/develop/examples/stack/bqc/README.md

earlier in the computation. Here, θ_1 and θ_2 are values in $[0, \frac{\pi}{4}, \frac{2\pi}{4}, ..., \frac{7\pi}{4}]$ chosen randomly by the client, and have no effect on the final result.

2.2 Determining the effectiveness

To assess the applications effectiveness as a benchmark, we execute the application on a simulated quantum network, and manually vary the simulated system parameters to see how imperfections in the system affect the results of the application. For these simulations we use SquidASM [10] [11], which uses NetSquid [12] to simulate quantum network systems and their physical properties.

We vary the simulated system parameters to explore the application's effectiveness at detecting these changes. We vary one parameter at a time, and set the rest of the system to be perfect, i.e. the rest of the system will not cause any errors to occur. This is not realistic when looking at current state-of-the-art systems, but allows us to focus more on the effect of the changed parameter. Appendix B shows what values parameters are set to when an experiment is being executed on another parameter. When the application is run, the calculation is performed, and the server measures the end state in the computational basis, and returns a classical bit to the client. We then calculate the accuracy of the returned results by comparing the amount of measured 0's to the expected amount of 0's. We assess whether these results are affected by the parameter we varied, and if so, we plot these results to explore how sensitive the application is to it.

We repeat this process for multiple system parameters, such as the link entanglement fidelity, the quality of quantum operations and the memory lifetimes of the individual nodes. Per parameter we also examine how different types of nodes, such as generic devices or Nitrogen Vacancy (NV) centres, affect the application's sensitivity to changes in the parameters. When using generic devices, these will always be connected with a depolarising link: a magic state distribution link with depolarising noise. On the other hand, NV-centres will always be connected using a heralded link. Additionally, we repeat this process for different input parameters from the client.

3 Experimental Setup

To setup the experiment, we generate a configuration for two quantum nodes and one quantum link that connects these nodes.

3.1 Nodes

For the individual nodes, we execute experiments with two different types of configurations. Firstly, we use a GenericQDevice configuration, which does not model a specific type of device, but instead simulates a generic quantum device. Secondly, we use an NVQDevice, which models an NV-centre device. Both devices have parameters for qubit initialization times, gate execution times and measuring times, which are all left unchanged from the default values in the SquidASM library throughout all simulations. The number of qubits per node also remains unchanged at 2 qubits per node.

For the GenericQDevice, the first parameters we vary are the T1 and T2 times - the decoherence times. We vary these together, and retain the ratio set in the default configuration, where $T_2 = \frac{T_1}{10}$. Additionally, we vary the probability of a depolarising error occurring for single and two qubit gates. We vary these seperately.

For the NVQDevice, we also vary the T_1 and T_2 times, but separately for the electron qubit and the carbon qubits. Again, we vary the T_1 and T_2 times together, and maintain the ratio that is set in the default configuration, namely $T_2 = \frac{3T_1}{10}$ for the electron qubit and $T_2 = \frac{T_1}{100}$ for the carbon qubit. We also separately vary the depolarising error for the single gates applied to carbon qubits and for those applied to the electron qubit. Additionally, experiments are executed for the depolarising error for the two qubit gates applied to one carbon qubit and the electron qubit. Two parameters we also experiment with that do not have a counterpart in the GenericQDevice are the probability of incorrectly measuring a 1 instead of a 0 when measuring the electron qubit, and conversely the probability of measuring a 0 instead of a 1.

3.2 The link

For the link between the two nodes, we also experiment with two different configurations: a DepolariseLink and a HeraldedLink. When experiments are run with GenericQDevices, we use a depolarise link. For NVQDevices, we use a heralded link.

The DepolariseLink has three parameters. Firstly, we can vary the fidelity of the generated EPR pair - a pair of entangled particles, named after Einstein, Podolsky and Rosen. Secondly, we can change the probability of successfully generating an EPR pair per cycle, and lastly we can vary the time it takes to execute such a cycle. Each of these parameters is varied separately. When experimenting with the fidelity, the success probability and cycle time are both set to be perfect: 1.0 and 0 respectively. These two parameters together determine how long it takes to generate an EPR pair.

The heralded link has six parameters that are varied for the experiments: The physical length of the link, the proba-

α	β	**Expected ratio of measured 0's**
$\frac{\pi}{2}$	$\frac{\pi}{2}$	1.0 (end state: $\|0\rangle$)
$-\frac{\pi}{2}$	$\frac{\pi}{2}$	0.0 (end state: $\|1\rangle$)
0	0	0.5 (end state: $\|+\rangle$)
π	π	0.5 (end state: $\|-\rangle$)
$\frac{\pi}{2}$	0	0.5 (end state: $\|+i\rangle$)
$\frac{\pi}{2}$	π	0.5 (end state: $\|-i\rangle$)
$\frac{\pi}{4}$	$\frac{\pi}{4}$	0.75
$-\frac{\pi}{4}$	$\frac{\pi}{4}$	0.25
$\frac{\pi}{3}$	$\frac{\pi}{3}$	0.875
$-\frac{\pi}{3}$	$\frac{\pi}{3}$	0.125

Table 1: All inputs used in every experiment, with the corresponding expected output ratio of measured 0's. If the inputs result in one of the Pauli basis states as an end state, this is also included.

bility that photons are lost when entering the connection on either side, the attenuation coefficient of fibre on either side, the dark-count probability per detection, the probability that the presence of a photon leads to a detection event, and the Hong-Ou-Mandel visibility of photons that are being interfered. When one parameter is varied, the others are left as the default configuration value set in the SquidASM library, except for the dark-count probability. This value is set to 0.01, because a perfect dark-count probability means that varying the other parameters has no effect. We set it to 0.01 to still keep it low enough that the dark count probability itself does not cause significant errors.

3.3 Inputs

We also run the experiments with different input angles α and β to see if different input angles result in different average accuracies. We manually choose different input combinations such that every experiment executes computations that result in all six of the Pauli basis states $|0\rangle, |1\rangle, |+\rangle, |-\rangle, |+i\rangle$ and $|-i\rangle$. Additionally, we choose two input combinations such that the end states are not any of the Pauli basis states, and for both of these inputs we also find the angles to experiment with their orthogonal states as end states. Table 1 shows all the different inputs that were used in the experiments, as well as the expected output ratio of measured zeroes. The table also includes which inputs result in the Pauli basis states as end states.

4 Responsible Research

The purpose of this research is to serve as a basis for further development of a benchmarking suite by QuTech. Therefore it is vital that our research be responsible and reproducible. We strive to research responsibly by supporting our claims and assumptions with references of academic literature.

Section 3 defines clearly how all experiments were performed, and describes the simulation environments and its parameters. This allows anyone with access to SquidASM and NetSquid to reproduce our results.

Since our experiments involve quantum mechanics, there will always be a certain degree of randomness involved. However, by executing multiple batches for each experiment, where each batch in turn includes numerous simulation results, we can achieve an accurate estimation of the true value of the metrics. As long as the experiments are reproduced with sufficient batches and simulations per batch, the results should be comparable to our results.

The results for every executed experiment are shown in this paper, and none of the results from these experiments have been modified, or have been filtered to include only certain outcomes. For experiments where the varied parameter did not affect the application, the graphs are not shown in the sections, but rather in Appendix C, because the graphs do not provide any useful information. Nevertheless, they are still included to be transparent in all our results.

5 Results

In this section, the results of the executed experiments are shown and discussed. For every experiment, the average accuracy from 10 batches is plotted against the value of the varying parameter. In each experiment, one batch consists of 1000 shots. Additionally, error bars are included that signify the standard error of the results.

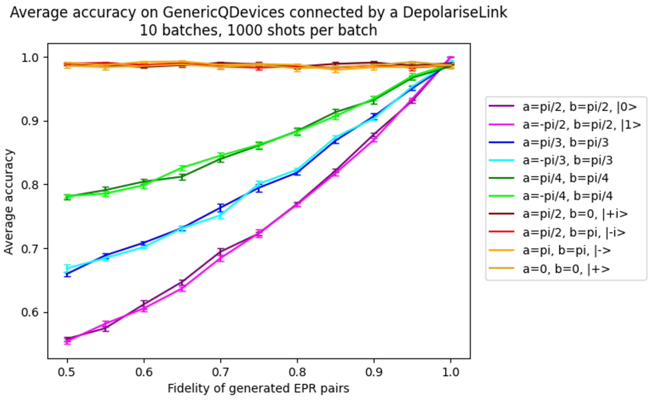

Figure 3: Average accuracy of experiments executed for different parameters of the DepolariseLink. The legend shows what input angles α and β were used. If these angles result in any of the six basis states, this state is included in the legend.

5.1 GenericQDevices connected by a DepolariseLink

For GenericQDevices connected by a DepolariseLink, we executed six experiments: three experiments where we vary parameters of the link, and three where we vary parameters of the nodes.

DepolariseLink

For the link, we only expect the fidelity of the generated EPR pairs to affect the application's accuracy, because this causes direct errors in the intermediate steps of the application, which will carry through to the final result. However, the other two parameters - the time it takes for one attempt at

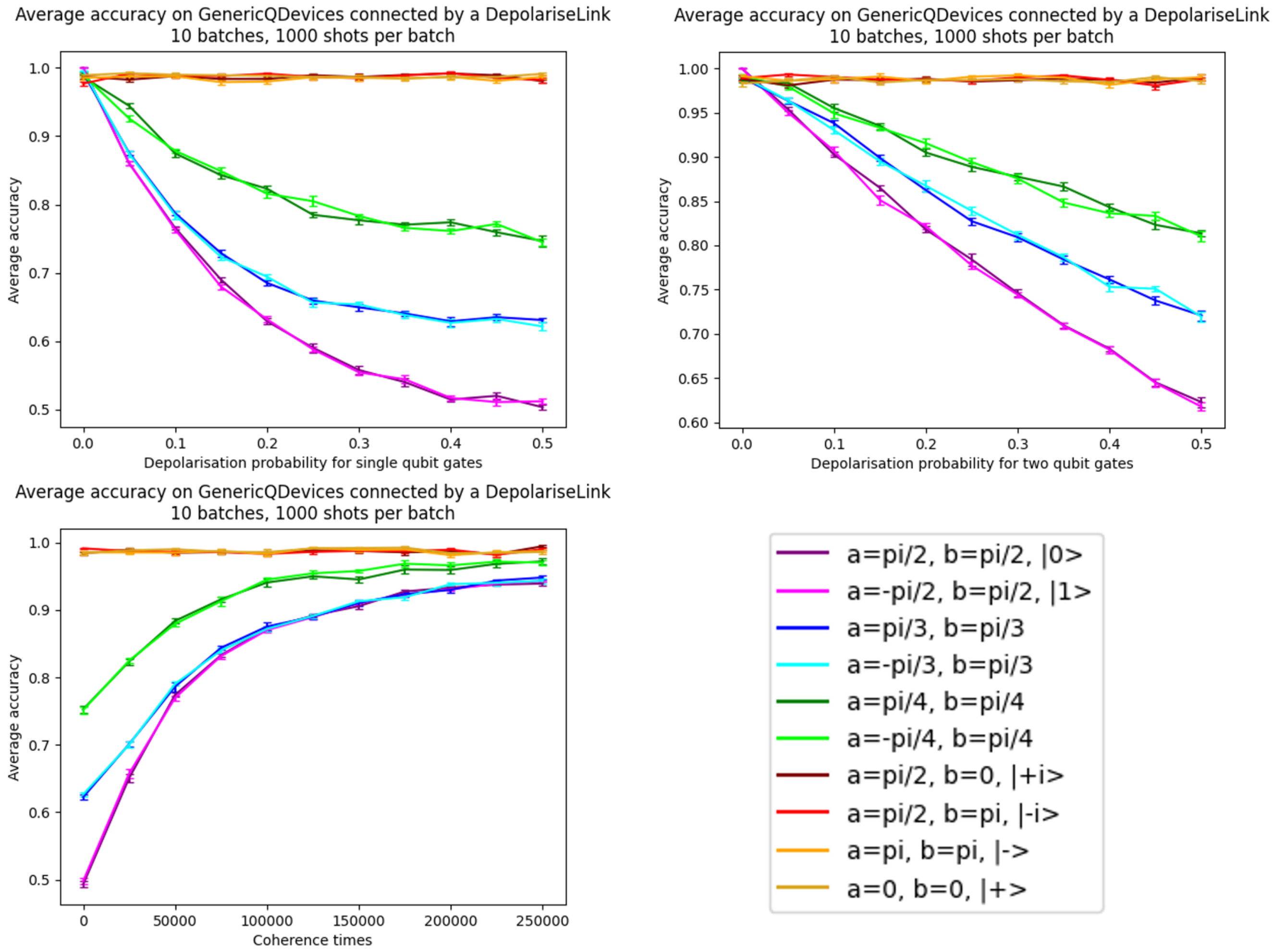

Figure 4: Average accuracy of experiments executed for different parameters of the GenericQDevice. The legend shows what input angles α and β were used. If these angles result in any of the six basis states, this state is included in the legend.

generating an EPR pair, and the success probability of such an attempt - are not expected to affect the application. This is because these parameters only change the amount of time it takes to generate an EPR pair, which will not cause errors if the decoherence times of the nodes are high enough, which is the case in our experiments.

From figure 3 we see that the application's average accuracy is indeed dependent on the link fidelity. A link fidelity of 1.0 results in an accuracy of 1.0. We also see that the accuracy depends on the inputs α and β.

For inputs that result in states with equal probabilities of measuring 0 and 1, such as any of the X and Y basis states, we see that the average accuracy remains around 1.0 independently of the link fidelity. The further the end state moves towards either the $|0\rangle$ or $|1\rangle$ state, the more the accuracy approaches 0.5 for lower link fidelities. This is because we calculate the accuracy with measurements of the end state in the computational basis. The errors that our varying link fidelity introduces will turn approximately as many measured 0's into 1's as it will turn 1's into 0's. This means that when we expect to measure the same amount of 0's as 1's, these errors cancel each other out. The more the ratio of measuring 0's and 1's is skewed towards one of them, the more the errors affect the accuracy. Additionally, we see that orthogonal end states result in approximately the same average accuracies.

As expected, changing the time it takes for one attempt at generating an EPR pair and the success probability of such an attempt does not affect the application's average accuracy. The results for these experiments can be found in Appendix C, along with the graphs of all other parameters that had no effect on the accuracy.

GenericQDevice

For the nodes, we expect all parameters to affect the results. Our application uses both single qubit gates and two qubit gates, so both the single qubit and two qubit depolarisation probability will directly cause errors to occur. Additionally, we expect the circuit from figure 2 to take long enough for lower coherence times to cause errors.

From figure 4 we see that indeed all parameters affect results from the application. Both the depolarisation probability

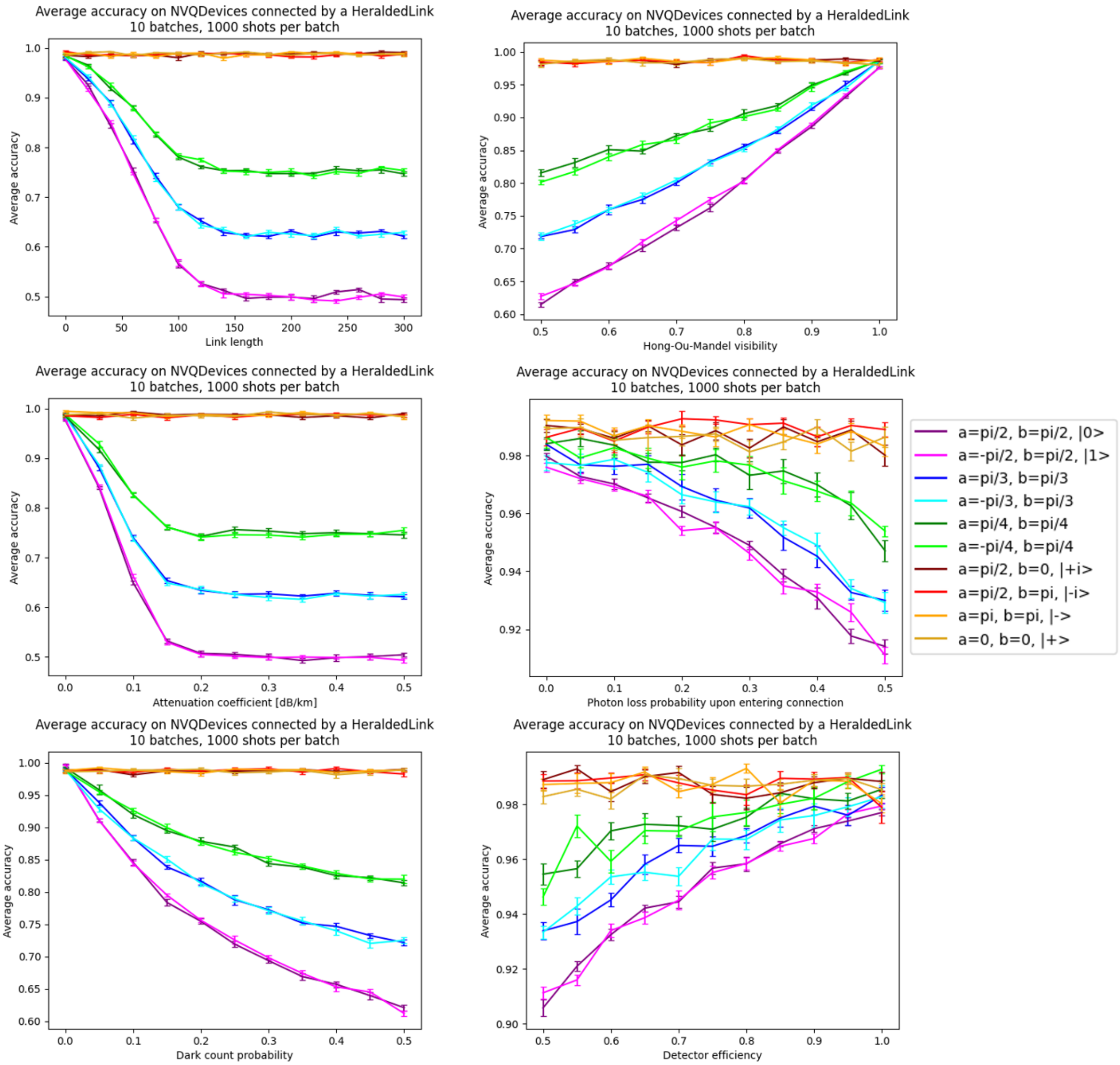

Figure 5: Average accuracy of experiments executed for different parameters of the HeraldedLink. The legend shows what input angles α and β were used. If these angles result in any of the six basis states, this state is included in the legend.

of the single and two qubit gates are negatively related to the average accuracy of the application. We see that the depolarisation probability of single qubit gates has a larger effect on the accuracy. This can be explained by the fact that the application uses significantly more single qubit gates than two qubit gates, as can be seen from figure 2. Also as expected, the coherence times affect the application's results. Lower coherence times result in lower accuracies.

Similar to the DepolariseLink we see that the more the end state moves towards the $|0\rangle$ or the $|1\rangle$ states, the more the accuracy is affected by a change in the parameters. Again, we also see that orthogonal states result in similar average accuracies.

5.2 NVQDevices connected by a HeraldedLink

For the simulations executed with NVQDevices connected by a HeraldedLink, fifteen experiments were performed, as both the link and the nodes have a lot more parameters that can be modified. The link has six parameters that were experimented with, and the nodes have nine. We expect all of these parameters to affect the results of the application, because they all directly cause errors to occur in the application.

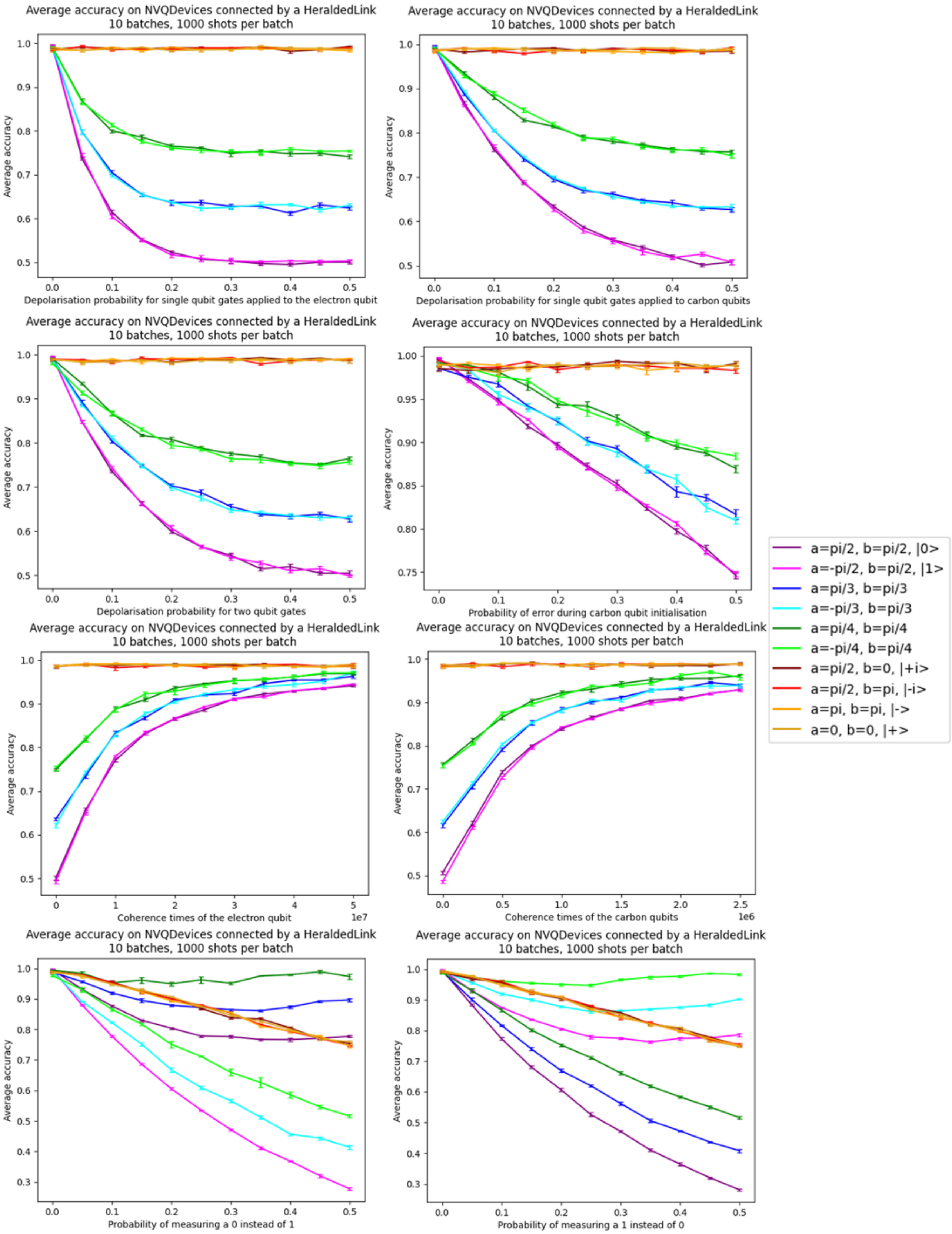

Figure 6: Average accuracy of experiments executed for different parameters of the NVQDevices. The legend shows what input angles α and β were used. If these angles result in any of the six basis states, this state is included in the legend.

HeraldedLink

Figure 5 shows the results of all experiments executed for the parameters of the HeraldedLink. Firstly, from the top-left graph we see that the length of the heralded link is inversely related to the average accuracy. A link with length 0 has an average accuracy of almost 1.

We see a comparable relation in the other two graphs on the left, which display the relation between the average accuracy and the attenuation coefficient and between the dark count probability. However, the dark count probability is not high enough for the accuracy to reach 0.5. Additionally, we see from the middle graph on the right that the probability of photon loss upon entering the connection has a similar relation, but here the average accuracy does not reach values below around 0.92, even when this probability is as high as 0.5.

Both of the remaining graphs on the right show similar relations to the other graphs, although here the relations are not inverse. For both the visibility and the detector efficiency, a perfect value for the parameter also results in an almost perfect accuracy. For the detector efficiency, the accuracy does not drop below 0.9 for a detector efficiency as low as 0.5.

Additionally, we see in all graphs that like the experiments executed for the DepolariseLink, the closer the end states are to the $|0\rangle$ or $|1\rangle$ states, the more accuracies are affected. running the experiment with two orthogonal states again results in approximately equal average accuracies.

NVQDevice

Figure 6 shows the results from the experiments executed for the parameters of NV-centres. We see that the application is affected by all the tested properties, except for the depolarisation probability during the initialisation of the electron qubit.

Additionally, while both the coherence times of the electron and the carbon qubits do affect the average accuracy, we see that it requires much higher coherence times for the application to reach an average accuracy of around 0.9. When we compare this to the GenericQDevice, we see that this already achieved such accuracies for coherence times of around 150,000ns. However, for the NVQDevice the coherence of the carbon qubit needs to be around 2,000,000ns to achieve an average accuracy of around 0.9, while the coherence times of the electron qubit needs to be 30,000,000 to reach that level of accuracy.

Two other interesting parameters are the probability of measuring a 0 instead of a 1, and the probability of measuring a 1 instead of a 0. We see that these parameters are the only two that affect the results when the end state of the application is halfway between the $|0\rangle$ and the $|1\rangle$ states. Additionally, for these parameters, orthogonal end states do not result in the same accuracies. For example, we see that for the inputs $\alpha = \frac{\pi}{4}$ and $\beta = \frac{\pi}{4}$, the average accuracy is not affected when varying the probability of measuring a 0 instead of 1, but when varying the probability of measuring a 1 instead of a 0, it is affected.

6 Discussion

This section will elaborate on the results from section 5, and discuss whether or not these results indicate that blind quantum computation is a good application to use for benchmarking a quantum network.

Table 3 shows all the parameters that were varied for the configuration using two GenericQDevices connected by a DepolariseLink. Similarly, table 2 shows the same thing, but for the experiments performed on two NVQDevices connected by a HeraldedLink. For every parameter it includes whether varying this parameter within the chosen range had an effect on average accuracy of the application. The chosen range and the resulting accuracies are always completely displayed in the graphs that display the experiment results for every parameter. For any parameter that did not affect the application's average accuracy, there is no corresponding graph included with the results. They can instead be found in Appendix C.

6.1 DepolariseLink

For the parameters of the DepolariseLink, the only one that affected the average accuracy of the application was the fidelity of generated EPR pairs, as can be seen in table 3. The success probability of generating entanglement and the time it takes for one attempt at generating entanglement did not affect the results of our experiments. This makes sense, as the only consequence of both these parameters is that it will take longer to successfully generate entanglement. If the decoherence times of the individual nodes are high enough, which was the case in our experiments, this will not affect the accuracy.

An application like this that is only affected by the link fidelity would make a very useful benchmark for this parameter specifically, since it is not influenced by other parameters. You could benchmark the link fidelities of two different links by connecting both links to identical nodes, and use the result of the application as a benchmarking score. However, to benchmark the entire link or system, you would have to use this application in combination with one or more applications that are sensitive to the remaining properties.

Additionally, further research could be done into the relation between the time taken to generate entanglement, and the decoherence times of the nodes, to see if there are situations where blind quantum computation would be sensitive to these parameters.

6.2 GenericQDevice

We can see from table 3 that for the GenericQDevice, the average accuracy of the application is affected by all parameters. This means that blind quantum computation is useful as a full-system benchmark, but less useful as a benchmark for individual parameters.

6.3 HeraldedLink

From table 2 we see that blind quantum computation is sensitive to every parameter of the heralded link. We do see that the application is less sensitive to the detector efficiency and to the probability of photons being lost when entering the connection, because the average accuracy does not drop below 0.9 for both experiments. However, the results of the application are still affected by these parameters.

HeraldedLink parameters	**Detectable**
Length	Yes
Attenuation coefficient	Yes
Probability of photons being lost when entering the connection	Yes
Dark count probability	Yes
Detector efficiency	Yes
Hong-Ou-Mandel visibility	Yes
NVQDevice parameters	**Detectable**
Depolarisation probability during initialisation of the electron qubit	No
Depolarisation probability of single qubit gate applied to the electron qubit	Yes
Depolarisation probability during initialisation of a carbon qubit	Yes
Depolarisation probability of single qubit gate applied to a carbon qubit	Yes
Depolarisation probability of two qubit gate applied to a carbon qubit and the electron qubit	Yes
Probability of measuring a 1 instead of 0 in an electron measurement	Yes
Probability of measuring a 0 instead of 1 in an electron measurement	Yes
Decoherence times of the electron qubit	Yes
Decoherence times of the carbon qubits	Yes

Table 2: A table showing whether the application can detect errors when these parameters are imperfect in a network consisting of NVQDevices connected by a HeraldedLink.

DepolariseLink parameters	**Detectable**
Fidelity of EPR pairs	Yes
Success probability of generating entanglement	No
Time taken for an entanglement generation attempt	No
GenericQDevice parameters	**Detectable**
Decoherence times	Yes
Single qubit gate depolarisation probability	Yes
Two qubit gate depolarisation probability	Yes

Table 3: A table showing whether the application can detect errors when these parameters are imperfect in a network consisting of GenericQDevices connected by a DepolariseLink.

This means it would be possible to use this application as a benchmark for the entire link. However, it would also mean that it cannot be used to benchmark specific parts of the link, so a benchmark suite that combines this application with multiple other applications, each of which is able to benchmark different specific parts of the system would be preferable when comparing two systems on specific properties, or when assessing improvements to specific parts of a system.

6.4 NVQDevice

Table 3 shows that for the NVQDevice, the average accuracy of the application is affected by all parameters, except for the depolarisation probability during electron qubit initialisation. Again, this means that the application could function well as a full-system benchmark, but would be less effective as a benchmark for individual parameters.

As mentioned in section 5, the probability of measuring a 0 instead of a 1, and the probability of measuring a 1 instead of a 0 are the only parameters that affect the inputs that cause the end state to be exactly halfway between the $|0\rangle$ and the $|1\rangle$ states. This means that you could use blind quantum computation with these inputs as a good benchmark for only these parameters. Additionally, we can use different inputs to differentiate between these two parameters. For example, for the inputs $\alpha = \frac{\pi}{4}$ and $\beta = \frac{\pi}{4}$, the average accuracy is affected when varying the probability of measuring a 1 instead of 0, but not when varying the probability of measuring a 0 instead of a 1. This can be used to further increase the applications effectiveness as a benchmark for these two parameters.

Another interesting property of the application is that the increasing accuracy only slows down for very high coherence times of both the carbon and electron qubit. This means the application is more future-proof as a benchmark, because it can benchmark for higher coherence times. On the other hand, this also means that the application is less sensitive to changes in the coherence times when you compare it to the experiments executed for the GenericQDevices, because the same increase in coherence times would result in a bigger change in the average accuracy than for the NVQDevice.

7 Future work

For the experiments executed for this paper, we only vary one parameter at a time. Possible future research includes executing similar experiments, but varying more parameters at a time. For example, for a depolarise link it might be useful to execute simulations where the success probability of an attempt at entanglement generation is varied together with the decoherence times of the nodes.

Another possibility for future research is to try and achieve similar results with only part of the application, the application could for example be reduced to the effective computation being only one rotation around the Z-axis, followed by a Hadamard gate. If we can still achieve similar results with this partial application, it would be beneficial because it would increase the efficiency of our benchmark suite.

8 Conclusion

The aim of this research was to assess the usefulness of blind quantum computation to benchmark a quantum network. We did this by experimenting with two different configurations: two generic quantum devices connected by a depolarising link, and two NV-centres connected by a heralded link. We used SquidASM to simulate these configurations, and manually vary individual parameters of the link or the nodes. We then calculate the average accuracy of the application to see how changes in this parameter affect the results of the application. Using these results we determine whether the application can be used as a benchmark for a quantum network.

Firstly, what we see from the results is that blind quantum computation would be a useful tool in benchmarking the fidelity of EPR pairs in a DepolariseLink. It would not be a good benchmark for the success probability of attempts at generating an EPR pair, nor for the time it takes to execute such an attempt. This means that to benchmark the entire link, the application would have to be combined with an application that is sensitive to these parameters.

Secondly, for the GenericQDevice, the application is useful as a benchmark for the entire system, but less effective as a benchmark for individual parameters. In this case, it would be preferable to combine blind quantum computation with other applications that can be used as a benchmark for individual parameters.

Similarly, we found that for the HeraldedLink, blind quantum computation can be used as a benchmark for the entire link, because it is sensitive to every parameter in the link. However, this also means that again, it is less useful to compare specific parameters between two systems, or to assess improvements to specific parts of the system.

Lastly, for the NVQDevice we also found that the application is good as a full-system benchmark, except for the depolarisation probability during the initialisation of the electron qubit. However, in this case the system would be a good benchmark for two of the individual parameters. Specifically, by using certain inputs, the only parameters that affect the applications average accuracy are the probability of incorrectly measuring 0 instead of 1, and the probability of incorrectly measuring a 1 instead of 0. For the other individual parameters, blind quantum computation would have to be combined with different applications that are better benchmarks for these parameters.

Thus, when looking at the full stack of a quantum network, blind quantum computation would be most effective as a full-system test, comparable to integration testing in software development. It would be beneficial to combine blind quantum computation with applications that can benchmark more individual parameters of the stack, similar to unit testing in software development.

A Background knowledge on quantum computing

This section will provide some background knowledge that will give enough information to understand the concepts used in this paper.

A.1 Qubits

In classical computing, a classical bit will always have one of two values, 0 or 1. In quantum computing, a qubit can be in a superposition of both 0 and 1. This is a state where the qubit is in between 0 and 1. To extract the information stored in a qubit, it needs to be measured. Upon measurement, the result will always be either 0 or 1. This is a probabilistic measurement, where the chance of measuring 0 or 1 is determined by this superposition. We can visualise this superposition using the Bloch Sphere, which can be seen in figure 7.

Any qubit state can be seen as point on this sphere. When this point is on top of the sphere, where it says $|0\rangle$, it means that upon measurement there is a 100% chance of measuring a 0. Similarly, when the point is at the bottom of the sphere, where it says $|1\rangle$, it means that upon measurement there is a 100% chance of measuring a 1. If the point is along the dotted line of the sphere, for example where it says $|+\rangle$, it means that upon measurement there is a 50% chance of measuring a 0, and a 50% chance of measuring 1.

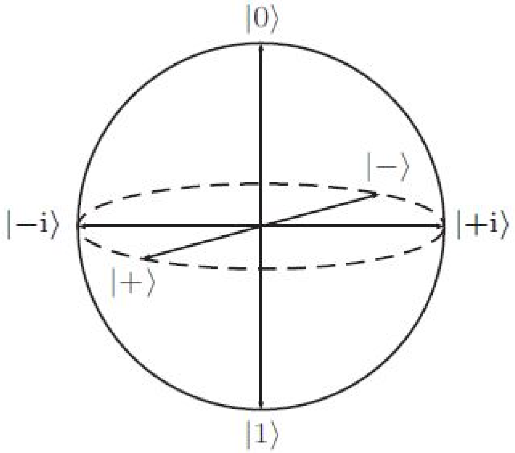

Figure 7: The Bloch Sphere. Image taken from Wikipedia: https://en.wikipedia.org/wiki/Bloch_sphere

A.2 Quantum circuits

Any operation that can be performed on a qubit, can be seen as a rotation on this sphere, or a combination of rotations. For instance, one of the gates used in this paper is the $R_z(x)$ gate, which rotates a point around the Z axis (the line going from $|0\rangle$ to $|1\rangle$), with angle x. For example, when a qubit is in the $|+\rangle$ state, and we apply a R_z gate with an angle of $\frac{\pi}{2}$, the qubit will end up in the $|+i\rangle$ state.

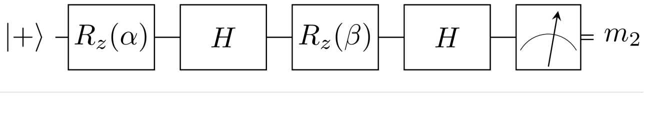

Figure 8: An example of a quantum circuit. Image taken from https://github.com/QuTech-Delft/squidasm/blob/develop/examples/stack/bqc/README.md

Figure 8 is an example of a quantum circuit. What we see from left to right is: The qubit starts in the $|+\rangle$ state. A rotation is then applied to the qubit around the Z-axis, with angle

α. After this, a Hadamard gate is applied to the qubit. A Hadamard gate is equivalent to a rotation around the Y-axis with angle $\frac{\pi}{2}$, followed by a rotation around the X-axis with angle π. This is followed by another rotation around the Z-axis with angle β, and then another Hadamard gate. After the last Hadamard gate, a measurement is performed on the qubit. This measurement will either return 0 or 1, which is based on the state the qubit is in after the four gates are applied to it. That means that angles α and β directly determine the probability of measuring a 0 or 1.

Quantum systems are still a very novel technology, meaning that they are still by no means perfect. For instance, applying quantum gates causes errors or noise to the qubits. Another example is that any qubit will naturally decay back to the $|0\rangle$ state over time. How many errors gates cause, or how long it takes for a qubit to decay is determined by the quality of the system. These parameters will be simulated and manually varied in this research to determine their effect on the results of the application.

B Default parameter values

Table 4 shows the default parameters values for the nodes and links we use in our simulations. These are the values that the parameters are set to when experiments are performed on other parameters.

C Parameters that do not affect the application results

Figures 9, 10 and 11 show the results of the experiments run on the configuration with GenericQDevices connected by a DepolariseLink that did not give any meaningful results. This means that the parameters that were varied did not affect the accuracy of the application, and therefore are not informative.

Due to the probabilistic nature of quantum mechanics, only the inputs where the end state is either the $|0\rangle$ or $|1\rangle$ state result in an average accuracy of 1.0. The other end states result in an average accuracy of around 0.99.

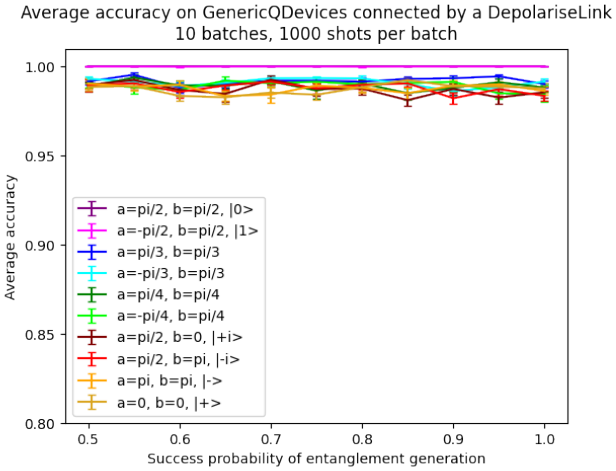

Figure 9: Results for experiments with a varying success probability of entanglement generation.

DepolariseLink parameter	**Value**
Fidelity of EPR pairs	1.0
Success probability of generating entanglement	1.0
Time taken for an entanglement generation attempt	0.0s
GenericQDevice parameter	**Value**
Decoherence times:	
T1 time	1e10 ns
T2 time	1e9 ns
Single qubit gate depolarisation probability	0.0
Two qubit gate depolarisation probability	0.0
HeraldedLink parameter	**Value**
Length	0.0km
Attenuation coefficient	0.25dB/km
Probability of photons being lost when entering the connection	0.0
Dark count probability	0.0
Detector efficiency	1.0
Hong-Ou-Mandel visibility	1.0
NVQDevice parameter	**Value**
Depolarisation probability during initialisation of the electron qubit	0.0
Depolarisation probability of single qubit gate applied to the electron qubit	0.0
Depolarisation probability during initialisation of a carbon qubit	0.0
Depolarisation probability of single qubit gate applied to a carbon qubit	0.0
Depolarisation probability of two qubit gate applied to a carbon qubit and the electron qubit	0.0
Probability of measuring a 1 instead of 0 in an electron measurement	0.0
Probability of measuring a 0 instead of 1 in an electron measurement	0.0
Decoherence times of the electron qubit:	
T1 time	1e9 ns
T2 time	$3 \cdot$ 1e8 ns
Decoherence times of the carbon qubits:	
T1 time	$1.5 \cdot$ 1e11 ns
T2 time	$1.5 \cdot$ 1e9 ns

Table 4: A table showing the default values for all parameters of the GenericQDevice, DepolariseLink, NVQDevice and HeraldedLink.

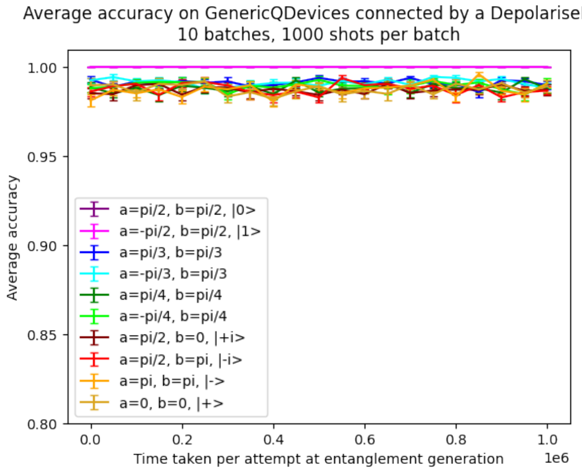

Figure 10: Results for experiments with a varying time taken per attempt at entanglement generation.

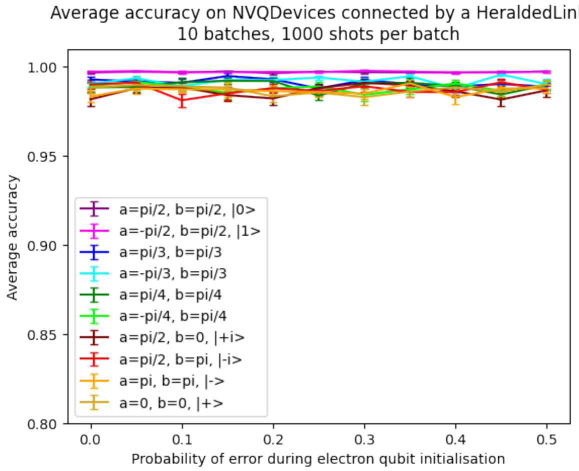

Figure 11: Results for experiments with a varying probability of depolarisation error during initialisation of the electron qubit

References

[1] C. Simon, "Towards a global quantum network," *Nature Photon*, vol. 11, pp. 678–680, Oct. 2017. DOI: 10.1038/s41566-017-0032-0. [Online]. Available: https://doi.org/10.1038/s41566-017-0032-0.

[2] R. Beals, S. Brierley, O. Gray, *et al.*, "Efficient distributed quantum computing," *Royal Society*, vol. 469, May 2013. DOI: 10.1098/rspa.2012.0686. [Online]. Available: https://doi.org/10.1098/rspa.2012.0686.

[3] P. Kómár, E. M. Kessler, M. Bishof, *et al.*, "A quantum network of clocks," *Nature Phys*, vol. 10, pp. 582–587, Jun. 2014. DOI: 10.1038/nphys3000. [Online]. Available: https://doi.org/10.1038/nphys3000.

[4] W. Kozlowski and S. Wehner, "Towards large-scale quantum networks," in *Proceedings of the Sixth Annual ACM International Conference on Nanoscale Computing and Communication*, ser. NANOCOM '19, Dublin, Ireland: Association for Computing Machinery, 2019, ISBN: 9781450368971. DOI: 10.1145/3345312.3345497. [Online]. Available: https://doi.org/10.1145/3345312.3345497.

[5] D. Sukachev and M. Bhaskar. "Announcing the AWS Center for Quantum Networking." (Jun. 2022), [Online]. Available: https://aws.amazon.com/blogs/quantum-computing/announcing-the-aws-center-for-quantum-networking/ (visited on 06/12/2023).

[6] P. Cage. "NRL Announces the Washington Metropolitan Quantum Network Research Consortium (DC-QNet)." (Jun. 2022), [Online]. Available: https://www.navy.mil/Press-Office/News-Stories/Article/3074856/nrl-announces-the-washington-metropolitan-quantum-network-research-consortium-d/ (visited on 06/12/2023).

[7] "Toshiba, Chicago Quantum Exchange Partner on New Project." (Apr. 2022), [Online]. Available: https://chicagoquantum.org/news/toshiba-chicago-quantum-exchange-partner-new-project-economy (visited on 06/12/2023).

[8] J. Eisert, D. Hangleiter, N. Walk, *et al.*, "Quantum certification and benchmarking," *Nature Reviews Physics*, vol. 2, pp. 382–390, Jun. 2020. DOI: 10.1038/s42254-020-0186-4. [Online]. Available: https://doi.org/10.1038/s42254-020-0186-4.

[9] J. Helsen and S. Wehner, "A benchmarking procedure for quantum networks," *npj Quantum Information*, vol. 9, Feb. 2023. DOI: 10.1038/s41534-022-00628-x. [Online]. Available: https://doi.org/10.1038/s41534-022-00628-x.

[10] B. van der Vecht, S. Wehner, and A. Dahlberg, *SquidASM GitHub repository*. [Online]. Available: https://github.com/QuTech-Delft/squidasm (visited on 05/31/2023).

[11] B. van der Vecht, S. Wehner, and A. Dahlberg, *SquidASM documentation*. [Online]. Available: https://squidasm.readthedocs.io/en/latest/index.html (visited on 05/31/2023).

[12] T. Coopmans, R. Knegjens, and A. e. a. Dahlberg, "Netsquid, a network simulator for quantum information using discrete events.," *Commun Phys*, vol. 4, p. 164, Jul. 2021. DOI: 10.1038/s42005-021-00647-8. [Online]. Available: https://doi.org/10.1038/s42005-021-00647-8.