



**A Test Suite for Quantum Network Applications**  
**Evaluation of the Distributed CNOT Application in its Ability to Benchmark Quantum Networks**

**W. van Loevezijn<sup>1</sup>**  
**Supervisors: S. Wehner<sup>1</sup>, R. Ashok Kumar Vattekkat<sup>1</sup>**

<sup>1</sup>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: Wout van Loevezijn  
Final project course: CSE3000 Research Project  
Thesis committee: S. Wehner, R. Ashok Kumar Vattekkat, A. Zaidman

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

**Abstract**—Benchmarks for quantum network systems currently are an underrepresented subject in the research field. This paper evaluates how informative a distributed CNOT gate application is, in recognizing errors in the total quantum system. The goal is to decide whether it should be included as a benchmark in a newly developed benchmark suite for quantum network systems. The application is simulated and its error rate on certain inputs is plotted against properties of the quantum network system. This way it is evaluated that the application is sensitive to certain parameters of the quantum link and quantum devices in the nodes. Additionally, a quantitative analysis on the results shows which inputs are most effective in reacting to errors in the total quantum system. That way it is possible to conclude that inputs in the x-basis are sensitive to the most parameters and that combining these results with the inputs  $|10\rangle$  or  $|11\rangle$  yields an optimal result. It can be included in a benchmark suite if it is sensitive to parameters to which non of the other applications in the suite are sensitive.

## I. INTRODUCTION

With the emergence of quantum computing to the repertoire of computer science, certain applications could be implemented which are proven to be impossible in the classical domain. One example is the unconditionally secure quantum key distribution algorithm [1]. These quantum applications also play a role in the domain computer networking, which creates the need for quantum networks.

A classical network of computers can be extended with a quantum link which shares entangled particles between network nodes. The quantum teleportation protocol [2] is an example where these entangled particles are used to share quantum states between nodes. A combination of these quantum nodes and quantum links is called a quantum network. However, such a full quantum network stack is still in its early stages of development and there are multiple challenges laying ahead before such a system could be deployed as a service which runs in parallel to the current day internet [3].

One of these challenges is to construct a proper quantum benchmark suite to objectively compare the performance of a range of different quantum hard- and software systems. By identifying a set of relevant performance metrics, developers of the stack can have a better hold on the overall workings of their system, which yields a better sense of direction in which to improve them on.

A test suite for analysis on quantum devices already exist. One of these examples presented in the SupermarQ paper [4]. This paper proposed multiple metrics which quantified the ability of an application to identify errors different properties of the total system. It then composed a benchmark suite of multiple applications which all scored well on at least one of these metrics. This way the benchmark sensitivity to the total quantum system is maximized. Other attempts to develop a quantum benchmark suite are more focused on individual metrics. IBM [5] has developed a widely accepted metric to quantify the largest circuit a quantum device could implement, which is called the quantum volume. Another paper: QPack [6] focuses more on the runtime, accuracy and scalability in their benchmark.

Inspired by these approaches, the aim of this research is to develop a benchmark suite fit the needs of quantum network systems. Moreover, the scope of this research is to evaluate one specific quantum network application in its ability to be

useful in such a benchmark suite. The application of interest is a distributed CNOT gate application [7], which operates on two separate network nodes.

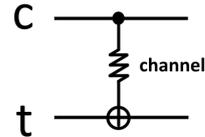


Figure 1. Distributed CNOT schematic. The control (c) and target (t) qubits on the control and target nodes are connected via a classical and quantum channel. The channel is capable of generating EPR-pairs and sending classical bits during execution.

Similar to a CNOT circuit gate on a single quantum device, the distributed CNOT application performs a gate operation on a control and a target qubit. However, these control and target qubits are located on two separate quantum devices. These nodes interact with each other through a quantum and classical link (see Fig. 1). The functionality of the distributed CNOT gate is comparable to a single CNOT gate and will generate the same output in a theoretically perfect environment.

Even though this is the desired case, for noisy intermediate-scale quantum technology noise is still one of the key limiting factors [8]. The different way the properties of the quantum link and device nodes have an effect on the distributed CNOT gate application are of interest in this paper. The research question is thus as follows:

*RQ: How informative is a distributed CNOT gate application as a benchmark for properties of the total quantum network system?*

Answering this research question will help achieve the broader goal to develop an informative quantum network benchmark suite, which may or may not include the distributed CNOT gate application as one of its benchmarks. To evaluate to which properties the application is sensitive and what not will give information in what benchmark suite this application as a performance metric could be useful.

To simulate the effect of these properties on the application, the SDK called SquidASM [9] is used, which helps users to write quantum network applications and tweak hardware parameters on which these applications are simulated. Performance metrics are constructed to which hardware parameters are tested in order to simulate errors in the quantum system. The sensitivity of these performance metrics then give an indication in the amount the distributed CNOT gate application is informative of properties of the total quantum system.

The remainder of the paper is organised as follows: an overview of the benchmark requirements, the quantum network application, simulation and metrics used to answer the research question is given in section II. The parameter settings for the experiments and results are described in section III. A critical analysis on the validity of the scientific method used in this paper is given in section IV. Lastly, the discussion, conclusions drawn from the data and future recommendations are stated in sections V & VI.

## II. METHODOLOGY

In this section some of the concepts related to the research goal are explored. Moreover, the reasoning behind the proposed method is explained to answer the research question. This section is divided in the following subsections: subsection II-A explains what is expected of an informative benchmark. Subsection II-B gives a quick introduction to quantum computing and explains the distributed CNOT application. Lastly, subsections II-C & II-D explain what system will be simulated to what performance metrics in the application.

### A. Benchmarks

A benchmark is defined as a “standard tool for the competitive evaluation and comparison of competing systems or components according to specific characteristics, such as performance, dependability, or security” [10]. In the case of this paper, we are evaluating whether the distributed CNOT gate application can properly function as a tool to compare properties of the quantum network system.

More specifically, we evaluate whether it has the ability to compare different properties in such a way that this application can be part of a bigger benchmark suite which test properties of the total quantum network system. Such a suite is the most informative on the errors in the system if every property of the system has an effect on at least one of the benchmarks in the suite. Therefore, for as many properties of the total quantum network its needs to be evaluated whether they have an affect on the performance of the distributed CNOT gate application as a benchmark to quantify as accurately as possible how informative it is in a benchmark suite.

The distributed CNOT gate application on its own is not a benchmark, since running it does not necessarily give a quantitative assessment of the quantum network system. However, the application can yield different performance metrics. These could be sensitive to different properties and thus need to be carefully constructed in order to give the desired result. There are different quality criteria [11] for choosing what benchmark is the correct one to use:

- 1) Relevance: Benchmarks should measure important features.
- 2) Representativeness: Benchmark performance metrics should be broadly accepted by industry and academia.
- 3) Equity: All systems should be fairly compared.
- 4) Repeatability: The benchmark results should be verifiable.
- 5) Cost-effectiveness: Benchmark tests should be economical.
- 6) Scalability: Benchmark tests should measure from a single server to multiple servers.
- 7) Transparency: Benchmark metrics should be readily understandable.

Since the goal of this research is to choose performance metrics and measure their sensitivity to errors in the total quantum system, the first criterion is the most relevant for this paper. However the other criteria nonetheless give a proper

sense of direction as it comes to evaluation of the benchmark as a whole.

### B. Application

The distributed CNOT gate application is a quantum application composed of quantum bits, quantum gates and entangled EPR-pairs. Firstly, a short explanation of these concepts is given. This is by no means a full explanation for all quantum concepts presented in this paper, but it is meant to give a quick intuition of the application and performance metrics used in the experiments. A more in depth introduction to quantum computing can be found at [2].

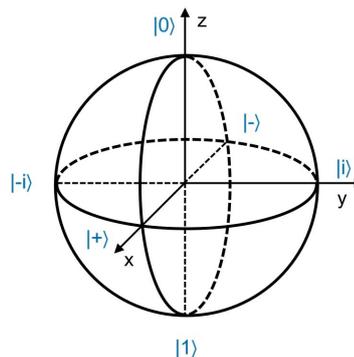


Figure 2. The Bloch Sphere which is used to represent qubit states [2]. The two pole states at the z-axis represent qubit state  $|0\rangle$  and  $|1\rangle$ . The pole states on the x-axis are  $|+\rangle$  and  $|-\rangle$ .

Just like classical bits have two states 0 and 1 which can be manipulated by applying gates in circuits, quantum bits (or qubits) can have state  $|0\rangle$  and  $|1\rangle$  which can be manipulated by quantum gates in quantum circuits. The difference being that classical bits are limited by being either 0 or 1, as qubits have a continuous space of quantum states it can be in. This space is called the Bloch Sphere (Figure 2). The state of a qubit can be represented as a point on this sphere and a single qubit gate can be interpreted as rotating this point on the sphere around a certain axis. For the most part the pole states of this sphere are relevant in this paper (shaded in blue).

One qubit gate of interest is the CNOT gate. This is a two qubit gate which performs an operation simultaneously on a control ( $c$ ) and target ( $t$ ) qubit. With inputs similar to classical bits ( $|0\rangle$  and  $|1\rangle$ ), the output on the target qubit acts similar to a classical XOR gate by flipping the  $|0\rangle$  to a  $|1\rangle$  on the target qubit if the control qubit is  $|1\rangle$ . However, due to the inputs being qubits, they are not limited to just  $|0\rangle$  and  $|1\rangle$ . Most interesting is the case of  $|+ -\rangle$ , where the control qubit is  $|+\rangle$  and the target qubit  $|-\rangle$ . The phase of  $t$  is then said to be transferred to  $c$  and the output is thus  $|--\rangle$ . This phenomenon is called ‘phase kickback’.

By cleverly manipulating qubit states it can be that two qubits become entangled. This means that measuring the state of one qubit gives information about the state of the other qubit. For example the state  $\frac{|00\rangle+|11\rangle}{\sqrt{2}}$  tells us that the two qubits have a 50% chance of being state  $|00\rangle$  and a 50% chance of being state  $|11\rangle$ . Measuring one qubit therefore immediately gives information about the state the other qubit

is in. This particular state is called a Bell state and the qubit pair is called an EPR-pair.

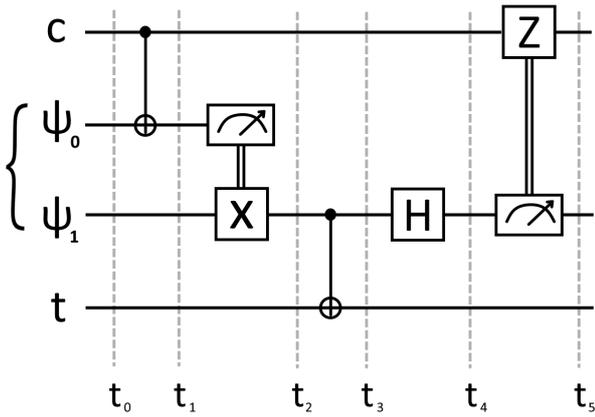


Figure 3. Implementation of the distributed CNOT circuit [7]. The control node holds qubits  $c$  and  $\psi_0$ , while the target node holds qubits  $\psi_1$  and  $t$ . Qubit  $c$  and  $t$  have the same control and target qubit functionality as the typical CNOT gate. Qubit  $\psi_0$  and  $\psi_1$  are an entangled EPR-pair, generated by the quantum link between the control and target node. Moreover, the classical measurement results of  $\psi_0$  and  $\psi_1$  at  $t_1$  and  $t_4$  respectively are sent over the classical channel to perform controlled  $X$  and  $Z$  gates.

The distributed CNOT gate application runs in two parts on two different quantum devices connected via a quantum link. Even though it behaves like the quantum CNOT gate, it is by no means a component of the system. It rather makes use of different parts of the quantum network system such as gates, qubits and the quantum link, which makes it a quantum network application.

The network nodes are either the 'control' or 'target' node, which represent the control and target qubits of a typical CNOT gate. One possible implementation [7] of the shared circuit diagram is displayed in Fig. 3. Assuming there exists no noise in the circuit gates, qubit memory lifetime and EPR-generation, the following equation holds:

$$Dist\_CNOT|ct\rangle = CNOT|ct\rangle$$

The proof for this claim can be found in Appendix A. In short,  $c$  becomes entangled with  $\psi_1$  at  $t_2$  by having similar outcomes, after which  $\psi_1$  acts as the control bit on the target node with  $t$  as the target qubit.  $t_3$  to  $t_5$  then handles the phase kickback on the control qubit  $c$ .

### C. Simulation

The distributed CNOT application is simulated using the software stack of QuTech called SquidASM [9]. QuTech is a company who is developing a full stack quantum network system. They released SquidASM as a high level SDK for the public to work with. It translates the code into assembly like language called NetQASM [12], which can be run on a simulation back-end called NetSquid [13]. Some applications are already written by QuTech [14], but they are not assessed in their ability to benchmark a quantum network system.

SquidASM allows for simulation of different quantum stack and link types. There are two stack types: a generic and NV device [15]. The NV device describes nitrogen-vacancy center qubits [15] which are proven to be able to generate entanglement over long distances [16]. There is also

the generic stack device. As the name implies, it has a more idealized simulation model for quantum circuits [15]. An overview of the hardware parameters of the different devices which are tested can be found in Table I.

Similar to the stack types are two link types available: a magic state distributor and heralded link [15]. The magic state distributor link 'magically' models the generation of an EPR-pair at both nodes with a certain fidelity and probability of success. However, the heralded link describes EPR generation by the double-click model mentioned by the following paper [17]. For the experiments all simulations with the magic state distributor link are conducted with generic device types as the nodes. Similarly, simulations with the heralded link have the NV device as nodes. An overview of all the hardware parameters of the different links can be found in Table II.

The amount of parameters which are tested on the stack devices is less extensive than the testing on the link parameters. The timing parameters of both the generic and NV devices are omitted from the experiments. The reason for this being, that the experiments of this research only focus on the fidelity of the application (see subsection II-D: Performance metrics) and increasing gate time reduces fidelity, which is expected to be similar to the effects on the depolarise probabilities. The results of the experiments conducted will therefore not include the ability to be able to detect errors in timing parameters.

Generic device	NV device
single qubit gate	electron init
two qubit gate	electron single qubit
	carbon init
	carbon z rotation
	ec gate
	0 error
	1 error

Table I  
DEPOLARISE PROBABILITY PARAMETERS OF THE SIMULATED QUANTUM NETWORK STACK TYPES.

Magic state distributor link	Heralded link
fidelity	length
t cycle	init loss probability
probability of success	length loss probability
	dark count probability
	detector efficiency
	visibility
	number resolving

Table II  
HARDWARE PARAMETERS OF THE SIMULATED QUANTUM NETWORK LINK TYPES.

### D. Performance metrics

Due to the exploratory nature of the research question, it is only possible to first evaluate theoretically in what ways the distributed CNOT application interacts with the quantum hardware and based on that: make an educated guess which measurable workings of the application could be particularly sensitive to certain hardware parameters, before they are

tested. Based on these results one could even tweak these performance metrics again to make an even better educated guess.

Regarding the quantum circuit presented in Fig. 3, it is possible to extract at least two functionalities. Firstly, the target qubit needs to be flipped in the classical basis, based on the input of the control qubit. Secondly, the control bit is flipped in the x-basis, based on the input of the target qubit. Both of these functionalities are the two nodes interacting with each other, which implies that the quantum link is being used. Therefore, the first performance metrics is giving different inputs in both the z and x-basis, and comparing the output with the expected case. These in and output sets are particularly useful, since none of these in- and output pairs are expected to be entangled:

$$CNOT|00\rangle = |00\rangle$$

$$CNOT|01\rangle = |01\rangle$$

$$CNOT|10\rangle = |11\rangle$$

$$CNOT|11\rangle = |10\rangle$$

$$CNOT|++\rangle = |++\rangle$$

$$CNOT|+-\rangle = |--\rangle$$

$$CNOT|-+\rangle = |-+\rangle$$

$$CNOT|--\rangle = |--\rangle$$

The performance metrics which are constructed are thus the fidelities of these sets of in- and outputs. This can be calculated by applying  $Y_{rot}$  and  $Z_{rot}$  gates to rotate the  $c$  and  $t$  qubits to their desired input state, applying  $Dist\_CNOT|ct\rangle$  and lastly rotate the expected qubits back to the  $|00\rangle$  state and measure them in the computational basis. This way we can construct an oracle. If  $|ct\rangle = |00\rangle$  we have a hit, so the oracle returns 0. Any other state represents a miss, which means some form of error has occurred and thus it returns 1. By running the application  $N$  amount of times and averaging the errors, we can extract the error rate of the application, which converges to the actual error rate when  $N$  grows to infinity. Moreover, we also calculate the standard error per data set and plot the results in a graph. In total we plot 10 different values per hardware parameter.

The approach used to generate results is having a perfect configuration for the node's stack and link except for one parameter we are tweaking. The result of interest is if there is any noticeable effect in the error rate of the application when adjusting this particular parameter. This means that any of the performance metrics change depending on what value the hardware parameter has. We can say that the performance metric is dependent on the hardware parameter if non-ideal inputs vary significantly compared to the ideal input.

After identifying which plots do have an effect, some statistics can be calculated on the data of the plots itself. Most importantly some quantitative analysis of the sensitivity could be of use, since it makes it possible to compare different performance metrics. First the results need to be interpolated to model the trend of the effect as a polynomial by minimizing the square error. The maximum of the derivative in

the interval of values tested could be calculated to see which effect is the steepest at any point in the model. There is no difference in use between a negative and a positive slope. Therefore the maximum value of the absolute value of the derivative is calculated for the sensitivity.

To speed up the simulation, the results are simulated in parallel on a pool of different processes. The number of processes which are spawned depends on the number of available processing threads on the host computer. To have good separation of concerns, the parameter configuration for a single data point is only set up once and all experiments are conducted and evaluated in the same process. Once the mean and standard error of the data point are returned the process can pick up a new data point to simulate if there are any left.

### III. EXPERIMENTAL SETUP AND RESULTS

The previous section explained some key concepts of the experiments necessary to answer the research question. This section will elaborate on these experiments by giving detailed instructions to generate results and showing these results. Since the experiments were conducted via simulation, many parameters used in the simulation are explained.

Firstly, there are system specific parameters. All experiments are processed on a 6-Core Intel Core i7-8750H on a 64-bit Ubuntu operating system version 22.04.2 LTS. The application is written in Python version 2.7 executed in the environment of the SquidASM SDK [9]. During execution, the application is forked into a pool of 12 processes, due to the processor of the system having 12 threads. Before running the distributed CNOT gate, a call is made to the NetSquid simulator [13] to renew its RNG seed. This part was necessary to make sure no two processes using the same seed due to it being copied while forking.

In total 26 different hardware parameters are plotted against the error rate of 8 different application inputs. Of the 26 hardware parameters, 3 are from the magic state distributor link configuration, 5 from the heralded link, 4 from the generic device and 14 from the NV device. Both the parameters of the generic and NV device are doubled due to it being tested on either the control or target node. An overview of all parameter values which are used in these plots can be found in table II & I. Every performance metric tests 10 different parameter values, which are the data points for the plot. One of the data points always represents a perfect parameter value. Each of the data points simulates the distributed CNOT gate application 250 times and calculates the mean, which represents the error rate, and the standard error of the mean. In total  $26 * 8 = 208$  plots are generated.

All these plots are evaluated in whether they show any effect in the error rate. That is, if the error rate of non-perfect parameters varies significantly compared to the error rate of the perfect input parameter. The error rate of the perfect parameter value is expected to be 0. Therefore, to decide whether any effect is present caused by adjusting the parameter value of its perfect configuration, the plots are visually assessed if they have at least one error rate significantly higher than 0. To make sure this hypothetical effect can only be caused by the parameter value being adjusted,

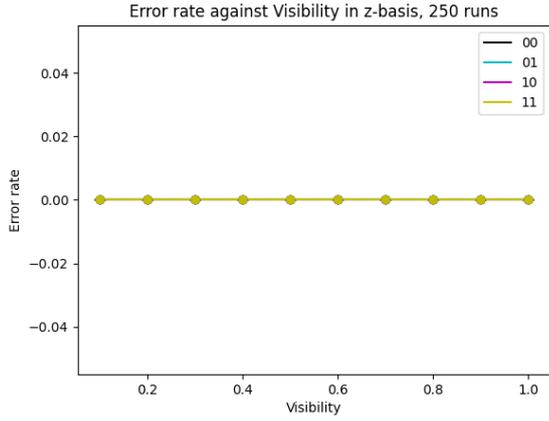


Figure 4. Error rate of the four input values in the z-basis plotted against the visibility hardware parameter of a heralded link configuration. The plot used the performance metrics on 250 runs of the distributed CNOT gate application to calculate one data point. At any point of the four different plots the error rate equals 0. These application inputs thus are not sensitive to the changes in the visibility parameter.

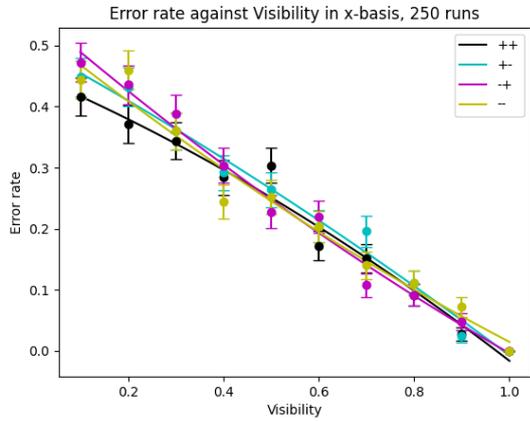


Figure 5. Error rate of the four input values in the x-basis plotted against the visibility hardware parameter of a heralded link configuration. The plot used the performance metrics on 250 runs of the distributed CNOT gate application to calculate one data point. Non perfect parameters show change in the error rate since they are not equal to 0. These application inputs thus are sensitive to the changes in the visibility parameter.

all other parameters adhere to a perfect configuration while performing the experiment. A list of all perfect parameter configurations is given in Appendix B. In short, all fidelity, efficiency and success values are set to 1, the gate execution times are 0s and all gate depolarise and loss probabilities are set to 0.

Not only the perfect parameter values, but also the range in which values are plotted vary per parameter. The parameters fidelity, probability of success, detector efficiency and visibility have input values from 0.1 to 1 with a step size of 0.1. This is because of the fact that these input values cannot have the value 0. For the values, initial loss probability, length loss probability, dark count probability and all depolarise probabilities it is the complete opposite. These values cannot have the value 1. That is why they vary from 0 to 0.9, with a step size of 0.1. T cycle is varied from the ideal value of 0ns to 1000ns. The length parameter is varied from 0km to 500km, since any value higher than 500km leads to a

crash. For the length loss probability the length is set as high as possible, such that it does not get errors when executing (100km). Lastly, all parameters of the heralded link are tested with number resolving on set to both *false* and *true*.

To illustrate the difference between a sensitive and non-sensitive performance metric the plots of the visibility parameter are shown in Figure 4 & 5. This is a parameter, which in the value range of 0.1 to 1.0 has an effect on the error rate of inputs in the x-basis, but not in the z-basis. Notice that the perfect value for visibility of 1.0 always has an error rate of 0 with a standard error of 0. All plots in the x-basis (Figure 5) show a non-zero error rate for non-perfect parameter values. The performance metrics with inputs in the x-basis are all sensitive to the visibility parameter. To the contrary, Figure 4 shows a perfect error rate for all non-perfect parameter values. The plots display a flat line at value 0 with an error rate of 0. The reason that only one line is visible is due to the fact the four plots lay exactly on top of each other. The performance metrics with inputs in the z-basis are thus not sensitive to the visibility parameter.

	fidelity	t cycle	probability of success	length	length loss probability	initial loss probability	dark count probability	detector efficiency	visibility
00	v					v*			
01	v					v*			
10	v					v*			
11	v					v*			
++	v					v*		v	
+-	v					v*		v	
-+	v					v*		v	
--	v					v*		v	
avg	1	0	0	0	0	1	0	0.5	

Table III  
APPLICATION INPUT VALUES WHICH ARE SENSITIVE TO CHANGES IN THE LINK PARAMETERS OF THE MAGIC STATE DISTRIBUTOR AND HERALDED LINK. ANY 'v' REPRESENTS THE APPLICATION INPUT BEING SENSITIVE TO THE HARDWARE PARAMETER. AN ABSENCE OF THE 'v' CHARACTER SHOWS THAT THE INPUT IS NOT SENSITIVE. THE LAST ROW AVERAGES THE AMOUNT OF SENSITIVE INPUTS PER PARAMETER. FOR THE PARAMETERS OF THE HERALDED LINK THE \* SIGNIFIES THAT THE PARAMETER ONLY HAS AN EFFECT WHEN NUMBER RESOLVING IS SET TO *false*.

The which performance metrics are sensitive is presented in Tables III, IV & V. Table III shows the performance metrics which are sensitive to the link parameters of the magic state distributor and heralded link. The same is done in Table IV & V. These tables show the whether a performance metric is sensitive to the depolarise probabilities of either the control or target node. Lastly, the last row of these tables shows the averaged amount of inputs the parameter has an effect on. This is calculated by summing the amount of sensitive inputs and dividing this number by the total amount of inputs tested, which is 8.

Also the sensitivity of every sensitive plot is presented in Tables VI, VII & VIII. This sensitivity is calculated by minimizing the squared error of a polynomial with degree 2. This degree is chosen due to the fact that by experimenting with different plots, all plots fitted at least a quadratic of linear relation. After that the maximum point of the absolute

	single qubit gate	two qubit gate	electron init	electron single qubit	carbon init	carbon z rotation	EC gate	0 error	1 error
00	v	v	v	v	v	v	v	v	v
01	v	v	v	v	v	v	v	v	v
10	v	v	v	v	v	v	v	v	v
11	v	v	v	v	v	v	v	v	v
++	v	v	v	v	v	v	v	v	v
+-	v	v	v	v	v	v	v	v	v
+ -	v	v	v	v	v	v	v	v	v
- -	v	v	v	v	v	v	v	v	v
avg	1	1	0.75	1	0.75	1	1	0	0

Table IV

APPLICATION INPUT VALUES WHICH ARE SENSITIVE TO CHANGES IN THE DEPOLARISE PROBABILITY OF DIFFERENT DEVICE PARAMETERS ON THE CONTROL NODE OF THE GENERIC AND NV DEVICE. ANY 'v' REPRESENTS THE APPLICATION INPUT BEING SENSITIVE TO THE HARDWARE PARAMETER. AN ABSENCE OF THE 'v' CHARACTER SHOWS THAT THE INPUT IS NOT SENSITIVE. THE LAST ROW AVERAGES THE AMOUNT OF SENSITIVE INPUTS PER PARAMETER.

	single qubit gate	two qubit gate	electron init	electron single qubit	carbon init	carbon z rotation	EC gate	0 error	1 error
00	v	v	v	v	v	v	v	v	v
01	v	v	v	v	v	v	v	v	v
10	v	v	v	v	v	v	v	v	v
11	v	v	v	v	v	v	v	v	v
++	v	v	v	v	v	v	v	v	v
+-	v	v	v	v	v	v	v	v	v
+ -	v	v	v	v	v	v	v	v	v
- -	v	v	v	v	v	v	v	v	v
avg	1	1	0	1	1	1	1	1	1

Table V

APPLICATION INPUT VALUES WHICH ARE SENSITIVE TO CHANGES IN THE DEPOLARISE PROBABILITY OF DIFFERENT DEVICE PARAMETERS ON THE TARGET NODE OF THE GENERIC AND NV DEVICE. ANY 'v' REPRESENTS THE APPLICATION INPUT BEING SENSITIVE TO THE HARDWARE PARAMETER. AN ABSENCE OF THE 'v' CHARACTER SHOWS THAT THE INPUT IS NOT SENSITIVE. THE LAST ROW AVERAGES THE AMOUNT OF SENSITIVE INPUTS PER PARAMETER.

value of the derivative of the polynomial  $ax^2 + bx + c$  is calculated as follows:

$$\max(|2a * \min\_value + b|, |2a * \max\_value + b|)$$

Lastly the average sensitivity is calculated per parameter by summing all values and dividing it by the amount of performance metrics, which is 8. The non-sensitive parameters are assumed to have a sensitivity of 0.0. The sensitivity is not normalized across parameters, since all parameters which are sensitive already have the same value range of 1.0.

#### IV. RESPONSIBLE RESEARCH

Since the research conducted in this paper is an attempt to broaden the knowledge in the public domain, it is important to reflect about the integrity and reproducibility of the claims made. This section explains how one could verify and reproduce the plots and data shown in Section III. Moreover, some reflection on the validity of the approach is explained.

In order to verify the results presented in Section III, all plots and results are zipped and saved on SURFDrive, which is accessible by QuTech. If the results of this research were

to be peer reviewed and published, these results could also be made public on a public data repository such as 4TU [18].

All plots are in a .png file format and the plotted data is stored in a .txt file. The title of these two files are the same for every experiment, except for the file extension. The data file has its own syntax. The first line is the title of the data. After that, every line represents one data point with the following syntax:

$$" < experiment\_type >, < parameter\_value >, \\ < mean >, < standard\_error > "$$

The  $experiment\_type$  represents with which performance metric the data point is calculated. It can be '00', '01', '10', '11', '+ +', '+ -', '- +' or '- -'.

Additionally, the code repository used in the experimental setup is publicly available [19]. The repository includes a README.md file which explains the installation procedure and how to use the files in the repository in order to get the same plots presented in Section III. Moreover, the code itself is thoroughly commented and has variable names similar to the concepts used in the paper. The latter is true except for the 'magic state distributor link', which is called the 'depolarise link' by the SquidASM repository. Lastly, there is also a tool in this repository which looks at all chart data in a given directory and calculates the sensitivity metric, which method is similar to the one used in the experiments.

Next to the data being verifiable and reproducible, it is also good practice to reflect on the validity of the data. Even though the plots do not show any anomalies, it is possible that there are bugs present in the code used for the experimental setup or code used for the SquidASM simulation. These bugs could cause the data to be misrepresented. The code in the experimental setup is especially vulnerable since its functionality is only tested by plotting data. However, the fact that both repositories are publicly available makes it possible to verify whether the code used is valid or not during the peer reviewing process. Any mistakes could be filtered out before any claims were to be published.

#### V. DISCUSSION

The goal of evaluating the sensitivity of different performance metrics on properties of the network system is to substantiate whether the distributed CNOT application is informative enough to be put into a benchmark suite for quantum network systems. Previous section summarized the results of which parameters have an effect on the eight performance metrics proposed by this paper. Of the sensitive cases, a quantitative estimate is given on their sensitivity. This section will highlight the interesting trends in this data and discuss their validity.

First of all, the check marks in Tables III, IV & V show that the performance metrics are sensitive to exactly half the parameters tested, when counting the parameters for which at least one of the eight performance metrics have an effect. However, most of these parameters are hardware parameters which show errors in one of the stack devices and not in the link. Only errors in the fidelity of a magic state distributor link and dark count probability and visibility of a heralded

link are noticeable. The dark count probability only has an effect when no photon-number-resolving detectors are used for the Bell-state measurement.

Furthermore, there is a different in device parameters on the control and target node. The initialization error of the electron spin is only noticeable in the target node, and not in the target node. Similarly 0 and 1 measurement errors are only noticeable if they are present in the control node and not in the target node.

Differences per performance metric are also present. The effect of the visibility of a heralded link is only present when inputs are used in the x-basis and not in the z-basis. Moreover, the electron initialization errors in the target node are not noticeable only when using the inputs  $|00\rangle$  and  $|01\rangle$ . Equivalently, carbon initialization errors in the target node only do not have an effect on the metrics with input  $|10\rangle$  and  $|11\rangle$ . So to summarize, the metrics with inputs in the x-basis have the most effect on the tested parameters and there exists no parameter to which only an input in the z-basis is sensitive.

	fideliity	dark count probability	visibility
00	0.68	1.21	-
01	0.71	1.37	-
10	0.75	1.33	-
11	0.75	1.38	-
++	0.79	0.60	0.60
+-	0.97	0.60	0.58
-+	0.67	0.63	0.65
--	0.71	0.84	0.60
avg	0.75	1.00	0.30

Table VI

SENSITIVITY OF DIFFERENT PERFORMANCE METRICS AGAINST ALL LINK PARAMETERS. ONLY THE SENSITIVITY IS DISPLAYED OF THE EXPERIMENTS WHICH ARE LABELLED AS SENSITIVE IN TABLE III.

Additionally, the amount to which a particular performance metric is sensitive to a parameter is presented in Tables VI, VII & VIII. These results show that metrics with inputs in the z-basis are significantly more sensitive to the dark count probability of a heralded link than with inputs in the x-basis. The sensitivities of the metrics to fidelity of a magic state distributor link do not vary significantly. For the metrics with inputs in the z-basis their highest sensitivity per parameter of the control stack is never significantly lower than the highest sensitivity of the metrics with an input in the x-basis. However, in reverse it is the case for the depolarise probability of the two qubit gate, electron single qubit and EC gate. Conversely, in the target node the highest sensitivity of metrics with an input in the x-basis are always higher or not significantly lower than the of the z-basis. In short, the z-basis is more sensitive to the depolarise probabilities in the control node and the x-basis to the depolarise probabilities in the target node.

Lastly, there are a few parameters for which the sensitivity of metrics differs significantly between inputs in the same basis. For the single qubit gate depolarise probability in the control node, the inputs  $|00\rangle$  and  $|01\rangle$  are way less sensitive than  $|10\rangle$  and  $|11\rangle$ . Similarly, in the target node  $|00\rangle$  is

	single qubit gate	two qubit gate	electron init	electron single qubit	carbon init	carbon z rotation	EC gate
00	0.29	0.99	-	1.93	0.52	1.71	1.67
01	0.39	0.82	-	1.98	0.54	1.59	1.80
10	1.19	0.90	0.53	2.18	-	1.76	2.14
11	1.23	1.06	0.82	2.16	-	1.66	2.26
++	1.03	0.68	0.52	1.50	0.59	1.44	1.53
+-	1.32	0.58	0.50	1.59	0.51	1.32	1.45
-+	1.20	0.66	0.62	1.47	0.59	1.38	1.42
--	1.48	0.45	0.54	1.34	0.54	1.46	1.32
avg	1.02	0.77	0.44	1.77	0.41	1.54	1.70

Table VII

SENSITIVITY OF DIFFERENT PERFORMANCE METRICS AGAINST ALL CONTROL STACK DEPOLARISE PROBABILITY PARAMETERS. ONLY THE SENSITIVITY IS DISPLAYED OF THE EXPERIMENTS WHICH ARE LABELLED AS SENSITIVE IN TABLE IV.

	single qubit gate	two qubit gate	electron single qubit	carbon init	carbon z rotation	EC gate	0 error	1 error
00	0.29	0.60	1.32	0.50	0.58	1.37	2.19	0.49
01	1.06	0.52	1.63	0.56	1.23	1.41	2.37	0.56
10	0.74	0.47	1.42	0.55	1.10	1.34	2.30	0.44
11	0.81	0.72	1.50	0.64	1.12	1.32	2.33	0.44
++	1.66	0.87	1.91	0.63	1.37	1.81	2.14	0.52
+-	2.09	1.00	2.04	0.49	1.88	1.63	2.32	0.56
-+	1.46	1.03	1.84	0.62	1.29	1.50	2.30	0.49
--	1.98	1.09	2.15	0.61	1.75	1.71	2.28	0.43
avg	1.26	0.79	1.73	0.58	1.29	1.51	2.28	0.49

Table VIII

SENSITIVITY OF DIFFERENT PERFORMANCE METRICS AGAINST ALL TARGET STACK DEPOLARISE PROBABILITY PARAMETERS. ONLY THE SENSITIVITY IS DISPLAYED OF THE EXPERIMENTS WHICH ARE LABELLED AS SENSITIVE IN TABLE V.

way less sensitive than  $|01\rangle$ ,  $|10\rangle$  and  $|11\rangle$  to the depolarise probability of both the single qubit gate and carbon z rotation. The sensitivities of metrics in the x-basis is more or less the same per parameter.

To summarize, any performance metric with an input in the x-basis will recognise all errors in the given parameters the distributed CNOT application is able to recognise as a benchmark. However, if more sensitivity is needed, it can be combined with an input in the z-basis, excluding  $|00\rangle$  and  $|01\rangle$ .

The validity of these observations is limited by certain factors however. First and foremost, the accuracy of the sensitivity is quite low, due to the it being based on a model which may not represent the relationship between parameters and metrics well enough. The data points are now interpolated by minimizing the squared error of a second degree polynomial. Adding more data points could already improve the accuracy by a lot. Moreover, the polynomial being modelled could also be replaced by a function with a more accurate representation of the data points. This process uses more trial and error per plot, but give more accurate results. Lastly, the data points itself could be more accurate by increasing the amount of runs of the application. This is now 250, but with an increase to 1000 runs per data point the standard error will already shrink by a factor of 2. A

more accurate estimation for the sensitivity of the metrics may yield even more insights on what performance metrics would be better to include in a benchmark suite.

Secondly, the estimation whether these performance metrics are informative in properties of the total quantum system is limited by the fact that the focus on this research lies on measuring fidelity and not metrics such as runtime. Even though quantum systems in the NISQ-era are primarily limited by fidelity constraints [8], in the future runtime may play a more important role as stressed in the paper [6]. At that time, this benchmark alone will fall short by being less relevant [11].

There are several suggestions for future research on this topic. First of all, it would be interesting to see whether more insights on the proposed performance metrics would arise by better modeling the effect of hardware parameters. Moreover, maybe the performance metrics are sensitive to more parameters when some parameters do not have a perfect value. This is for example the case for length loss probability, for which the length parameter of the heralded link was set to 100km for the experiments. Additionally, the metrics were only sensitive to the dark count probability when no photon-number-resolving detectors are used.

Another suggestion would be to search for measurable effects outside the domains proposed by this paper. To start the gate times could be tested against the performance metrics to see if actually no different effect is present. More fidelity performance metrics could be constructed for entangled inputs and outputs of the control and target qubit. As already stated, runtime metrics could be constructed of the distributed CNOT application which stress the compilation and execution capabilities of the quantum network stack. Lastly, it could be an interesting to assess scalability. It is possible to construct larger applications from the distributed CNOT gate application to stress the stack even more and maybe amplify the sensitivity of the benchmark. An example of such an application is a distributed quantum swap gate. Which is constructed by three consecutive and alternating CNOT gates.

## VI. CONCLUSIONS AND FUTURE WORK

The goal of this research is to evaluate and explore whether the distributed CNOT gate application could be used for a benchmark which is informative about properties of the total quantum network system. An informative benchmark would mean it is sensitive to errors in a wide arrange of parts of the quantum network system. Moreover, if it were to be included in a benchmark suite together with other applications, it is important to know to which properties the distributed CNOT gate application is sensitive and to which it is not.

The distributed CNOT gate application behaves similar to a CNOT quantum gate. However, it is composed of multiple gates on two separate nodes, connected via a quantum link. The fact that it has all these components makes it an application which interacts with a quantum network system.

Some performance metrics can be constructed from the functionalities of this distributed CNOT gate. Since it behaves similar to a CNOT circuit gate, it is possible to compare the input with the expected output and calculate the error rate by running the application multiple times.

For the experiment the application is simulated using the SquidASM SDK. The error rate of inputs in the z and x-basis are tested against multiple hardware parameters in both the link and stack devices in the nodes. These results are plotted and the sensitivity of these inputs is calculated.

From these experiments it is concluded that the application is sensitive to a wide arrange of errors in different hardware parameters. It was most sensitive to errors in the depolarise probabilities in the stack devices and to a few in the link parameters, namely the fidelity of a magic state distributor link and dark count probability and visibility of a heralded link.

Additionally, there was a difference in sensitivity to parameters depending on the input used. The inputs in the x-basis are sensitive to all parameters the application is sensitive to. However, when the optimal sensitivity want to be achieved it can be combined with either the input  $|10\rangle$  or  $|11\rangle$ . No noticeable difference between the sensitivity of the inputs in the x-basis could be concluded. However, specific inputs could be recommended for a benchmark suite if the sensitivity is more accurately calculated by running the application a bigger amount of times, running more data points or modelling the plot more accurately.

The application being sensitive to a range of different properties in both the quantum link and stack devices makes it a viable option to put in a benchmark suite. However, it being suitable to put in a benchmark suite also depends to what degree it is mutually exclusive in which parameters it is sensitive with respect to the other applications in the suite. The way the performance metrics are chosen is that it is made sure that input in both the control and target node have an effect on the output of the other node. Therefore, it is designed that the performance metrics are as sensitive as possible to errors in both nodes. If other network applications in the suite have a more one way sensitivity to a node, the distributed CNOT gate application as a benchmark could be particularly useful.

The future work of this research includes the following. Parallel to this research, three other quantum network applications are graded in a similar fashion. When the results of four different quantum network applications are concluded, it is possible to evaluate whether these applications can be composed in a benchmark suite which test properties of the total quantum networking system. Moreover, even when a quantum network system benchmark is developed from the results of this research it is important that it is frequently updated to the needs of the rapidly changing field of quantum computation [20].

APPENDIX A  
IDEAL DISTRIBUTED CNOT PROOF

Proof.  $Dist\_CNOT|ct\rangle = CNOT|ct\rangle$  in the ideal case.

Assume the entangled EPR-pair at  $t_0$  is the following:

$$|\psi\rangle = |\psi_0\psi_1\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Moreover we represent  $|c\rangle$  and  $|t\rangle$  at  $t_0$  as follows:

$$|c\rangle = \alpha|0\rangle + \beta|1\rangle, |t\rangle = \gamma|0\rangle + \delta|1\rangle$$

$t_0$ :

$$|c\psi\rangle = \frac{1}{\sqrt{2}}(\alpha|0\rangle(|00\rangle + |11\rangle) + \beta|1\rangle(|00\rangle + |11\rangle))$$

$t_1$ : perform  $CNOT|c\psi_0\rangle$

$$|c\psi\rangle = \frac{1}{\sqrt{2}}(\alpha|0\rangle(|00\rangle + |11\rangle) + \beta|1\rangle(|10\rangle + |01\rangle))$$

$$|c\psi\rangle = \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|110\rangle + \beta|101\rangle)$$

$t_2$ : measure  $\psi_0$ , which is either 0 or 1. When it is 1 perform X on  $\psi_1$ .

$$|c\psi_1\rangle(0) = \alpha|00\rangle + \beta|11\rangle$$

$$|c\psi_1\rangle(1) = \alpha|00\rangle + \beta|11\rangle$$

$$|c\psi_1t\rangle = (\alpha|00\rangle + \beta|11\rangle)(\gamma|0\rangle + \delta|1\rangle)$$

$$|c\psi_1t\rangle = \alpha\gamma|000\rangle + \alpha\delta|001\rangle + \beta\gamma|110\rangle + \beta\delta|111\rangle$$

$t_3$ : perform  $CNOT|\psi_1t\rangle$

$$|c\psi_1t\rangle = \alpha\gamma|000\rangle + \alpha\delta|001\rangle + \beta\gamma|111\rangle + \beta\delta|110\rangle$$

$t_4$ :

$$|c\psi_1t\rangle = \frac{1}{\sqrt{2}}(\alpha\gamma|0\rangle(|0\rangle + |1\rangle)|0\rangle + \alpha\delta|0\rangle(|0\rangle + |1\rangle)|1\rangle + \beta\gamma|1\rangle(|0\rangle - |1\rangle)|1\rangle + \beta\delta|1\rangle(|0\rangle - |1\rangle)|0\rangle)$$

$$|c\psi_1t\rangle = \frac{1}{\sqrt{2}}((\alpha\gamma|0\rangle + \beta\delta|1\rangle)|00\rangle + (\alpha\delta|0\rangle + \beta\gamma|1\rangle)|01\rangle + (\alpha\gamma|0\rangle - \beta\delta|1\rangle)|10\rangle + (\alpha\delta|0\rangle - \beta\gamma|1\rangle)|11\rangle)$$

$t_5$ : measure  $\psi_1$ , which is either 0 or 1. When it is 1 perform Z on  $c$ .

$$|ct\rangle(0) = (\alpha\gamma|0\rangle + \beta\delta|1\rangle)|0\rangle + (\alpha\delta|0\rangle + \beta\gamma|1\rangle)|1\rangle$$

$$|ct\rangle(1) = (\alpha\gamma|0\rangle + \beta\delta|1\rangle)|0\rangle + (\alpha\delta|0\rangle + \beta\gamma|1\rangle)|1\rangle$$

$$|ct\rangle = \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|11\rangle + \beta\delta|10\rangle$$

Which means that  $Dist\_CNOT$  does the following to  $|ct\rangle$ :

$$Dist\_CNOT|ct\rangle = \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|11\rangle + \beta\delta|10\rangle$$

When performing CNOT on  $|ct\rangle$  directly instead of  $Dist\_CNOT$ :

$$CNOT|ct\rangle = \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|11\rangle + \beta\delta|10\rangle$$

Therefore the following equation holds in the ideal case:

$$Dist\_CNOT|ct\rangle = CNOT|ct\rangle$$

QED.

APPENDIX B  
PERFECT CONFIGURATION

The following parameter values are assumed to be the perfect configuration for the magic state distributor link, heralded link, generic device and NV device, used in SquidASM to simulate the experiments conducted in Section III.

**Magic state distributor link**

- fidelity: 1.0
- t\_cycle: 0.0
- prob\_success: 1.0

**Heralded link**

- length: 0.0
- p\_loss\_init: 0.0
- p\_loss\_length: 0.0
- speed\_of\_light: 200\_000
- dark\_count\_probability: 0.0
- detector\_efficiency: 1.0
- visibility: 1.0
- num\_resolving: False

**Generic device**

- num\_qubits: 2
- T1: 10\_000\_000\_000
- T2: 1\_000\_000\_000
- init\_time: 0
- single\_qubit\_gate\_time: 0
- two\_qubit\_gate\_time: 0
- measure\_time: 0
- single\_qubit\_gate\_depolar\_prob: 0.0
- two\_qubit\_gate\_depolar\_prob: 0.0

**NV device**

- num\_qubits: 2
- electron\_init\_depolar\_prob: 0.0
- electron\_single\_qubit\_depolar\_prob: 0.0
- prob\_error\_0: 0.0
- prob\_error\_1: 0.0
- carbon\_init\_depolar\_prob: 0.0
- carbon\_z\_rot\_depolar\_prob: 0.0
- ec\_gate\_depolar\_prob: 0.0
- electron\_T1: 1\_000\_000\_000
- electron\_T2: 30\_000\_000
- carbon\_T1: 150\_000\_000\_000
- carbon\_T2: 1\_500\_000\_000
- carbon\_init: 0
- carbon\_rot\_x: 0
- carbon\_rot\_y: 0
- carbon\_rot\_z: 0
- electron\_init: 0
- electron\_rot\_x: 0
- electron\_rot\_y: 0
- electron\_rot\_z: 0
- ec\_controlled\_dir\_x: 0
- ec\_controlled\_dir\_y: 0
- measure: 0

## REFERENCES

- [1] D. Mayers, “Unconditional security in quantum cryptography,” *Journal of the ACM*, vol. 48, no. 3, pp. 351–406, May 2001.
- [2] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*, 10th ed. Cambridge UK: Cambridge University Press, 2010.
- [3] S. Wehner, D. Elkouss, and R. Hanson, “Quantum internet: A vision for the road ahead,” *Science*, vol. 362, no. 6412, pp. 1–9, Oct 2018.
- [4] T. Tomesh, P. Gokhale, V. Omole, G. S. Ravi, K. N. Smith, J. Vizslai, X.-C. Wu, N. Hardavellas, M. R. Martonosi, and F. T. Chong, “Supermarq: A scalable quantum benchmark suite,” <https://arxiv.org/abs/2202.11045>, Apr 2022.
- [5] A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, and J. M. Gambetta, “Validating quantum computers using randomized model circuits,” *Physical Review A*, vol. 100, no. 3, Sep 2019.
- [6] K. Mesman, Z. Al-Ars, and M. Möller, “Qpack: Quantum approximate optimization algorithms as universal benchmark for quantum computers,” <https://arxiv.org/abs/2103.17193>, Apr 2022.
- [7] “Distributed cnot github repository,” [https://github.com/QuTech-Delft/netqasm/tree/develop/netqasm/examples/apps/dist\\_cnot](https://github.com/QuTech-Delft/netqasm/tree/develop/netqasm/examples/apps/dist_cnot), Jun 2023.
- [8] J. Preskill, “Quantum Computing in the NISQ era and beyond,” *Quantum*, vol. 2, p. 79, Aug 2018.
- [9] B. van der Vecht, S. Wehner, and A. Dahlberg, “SquidASM GitHub repository,” Jun 2023. [Online]. Available: <https://github.com/QuTech-Delft/squidasm>
- [10] J. von Kistowski, J. Arnold, K. Huppler, K.-D. Lange, J. Henning, and P. Cao, “How to build a benchmark,” *ICPE 2015 - Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering*, Feb 2015.
- [11] W. Dai and D. Berleant, “Benchmarking contemporary deep learning hardware and frameworks: A survey of qualitative metrics,” *2019 IEEE First International Conference on Cognitive Machine Intelligence (CogMI)*, Dec 2019.
- [12] A. Dahlberg, B. van der Vecht, C. D. Donne, M. Skrzypczyk, I. te Raa, W. Kozłowski, and S. Wehner, “NetQASM—a low-level instruction set architecture for hybrid quantum–classical programs in a quantum internet,” *Quantum Science and Technology*, vol. 7, no. 3, p. 035023, Jun 2022.
- [13] T. Coopmans, R. Knegjens, and A. D. et al., “Netsquid, a network simulator for quantum information using discrete events,” *Communications Physics*, vol. 4, no. 164, Dec 2020.
- [14] M. Delavar, B. Dirke, M. Doosti, V. Lipinska, N. Mathur, G. Murta, R. Parekh, J. Ribeiro, S. Singh, G. Ustun, and C. Wadhwa, “Quantum protocol zoo,” <https://wiki.veriqcloud.fr/>, Jun 2023.
- [15] B. van der Vecht, S. Wehner, and M. van Hooft, “Squidasm documentation,” <https://squidasm.readthedocs.io/>, Jun. 2023.
- [16] B. Hensen, H. Bernien, A. Dréau, A. Reiserer, N. Kalb, M. Blok, J. Ruitenbergh, R. Vermeulen, R. Schouten, C. Abellán, V. P. W. Amaya, M. W. Mitchell, M. Markham, D. Twitchen, D. Elkouss, S. Wehner, T. Taminiau, and R. Hanson, “Loophole-free bell inequality violation using electron spins separated by 1.3 kilometres,” *Nature*, vol. 526, pp. 682–686, Aug 2015.
- [17] B. Hensen, N. Kalb, M. S. Blok, A. E. Dréau, A. Reiserer, R. F. L. Vermeulen, R. N. Schouten, M. Markham, D. J. Twitchen, K. Goodenough, D. Elkouss, S. Wehner, T. H. Taminiau, and R. Hanson, “Loophole-free bell test using electron spins in diamond: second experiment and additional analysis,” *Scientific Reports*, vol. 6, no. 1, Aug 2016.
- [18] “4TU research data repository,” <https://data.4tu.nl/>, Jun 2023.
- [19] W. van Loevezijn, “Distributed cnot gate application experimental setup repository,” <https://github.com/woutvanlovezijn/Distributed-CNOT-gate-application-experimental-setup>, Jun 2023.
- [20] H. Donkers, K. Mesman, Z. Al-Ars, and M. Möller, “Qpack scores: Quantitative performance metrics for application-oriented quantum computer benchmarking,” <https://arxiv.org/abs/2205.12142>, May 2022.