Effect of measurement errors on the failure probability of quantum-aided Byzantine agreement

Kimon Kyparos[1]
Supervisor: Tim Coopmans[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 22, 2025

Name of the student: Kimon Kyparos
Final project course: CSE3000 Research Project
Thesis committee: Tim Coopmans, Arie van Deursen

Abstract

Most classical Byzantine agreement protocols between n nodes offer a fault tolerance t of up to $t < n/3$. Quantum fault-tolerant consensus protocols have been proposed that achieve a tolerance of $t < n/2$ and are therefore worth studying. In this paper, we assess the failure probability of a previously proposed quantum-aided weak broadcast protocol and how it is affected by a physical error source. Specifically, we study the effect of measurement error noise. We simulate the protocol as-is on a four-node quantum network composed of NV-center devices, both with zero and with one faulty node. We then apply a measurement error noise model and compare the failure probability with the noiseless version. The noise "strength" is also varied in order to assess whether the failure probability can be reduced using improved hardware. We show that measurement errors have a significant negative effect on the failure probability of the protocol. In fact, even with $10x$ improved hardware parameters, the protocol does not achieve an acceptable failure probability.

1 Introduction

Distributed and multi-agent systems often need the participating nodes to reach consensus on a certain value. From blockchain to distributed databases[16], the operation of these systems relies on consensus algorithms, such as Byzantine agreement. A recurring issue in these agreement protocols is fault tolerance, or Byzantine Fault Tolerance (BFT) in the case of Byzantine agreement. Byzantine fault refers to a condition in which some components of a distributed system behave inconsistently, perhaps acting maliciously, exhibiting different symptoms to different observers. BFT in turn refers to the ability of a system to reach consensus in spite of the occurrence of Byzantine faults.

Let n be the number of components in the system and t the maximum number of components exhibiting Byzantine fault. Classical implementations of fault-tolerant protocols have been developed which allow up to $t < n/3$ faulty components[7]. Building upon the ideas of the classical implementation, a quantum-aided "weak broadcast" protocol has been proposed[4], which relies on a four-qubit singlet state. This Weak Broadcast Protocol (WBC) offers higher tolerance, namely $t < n/2$, making it an attractive alternative to the classical implementations.

Subsequent analysis was carried out by Guba et al.[8], which focuses on determining optimal values for the various parameters of WBC. However, little emphasis is placed on evaluating the effect of "noise" on the failure probability of the protocol. In particular, the authors study the effect of physical errors, but do not differentiate between the different sources of noise. Additionally, the authors study WBC and this generalized notion of noise using analytical methods and Monte-Carlo simulations, but no attempt is made to simulate the behaviour of the protocol in an actual quantum network. The paper does mention the use of IBM Q and IonQ hardware, though only for the creation of the four-qubit state on which the protocol is based. The goal of this paper is therefore to assess how WBC would be affected by a certain source of noise, namely measurement errors, in the quantum network. Specifically, we ask the question: "How is the failure probability of the quantum Byzantine agreement protocol influenced by measurement errors?".

The main contribution of this paper is answering this research question by simulating WBC on a four-qubit quantum network, implementing a noise model to mimick measurement errors and analysing its effect on the failure probability of the protocol.

The structure of the paper is as follows. Section 2 provides background information and describes the methodology used to simulate WBC. In Section 3, the measurement error noise model is explained in more detail. Assumptions about the noisiness, or lack thereof, of the various operations of the network are also discussed. In Section 4, the specific simulation setup is described, including the model parameters, the sampling method and the output to be used for data analysis. Results are then produced by running the simulation according to this specification and conclusions are drawn regarding the failure probability of the noisy protocol. In Section 5, the results obtained previously are discussed and compared to the analytical results of [8]. The effects of the noise model are also explained qualitatively. Section 6 focuses on the ethical aspects of the research and the reproducibility of the simulated results. Lastly, Section 7 answers the research question based on the conclusions drawn in Section 4. Recommendations for further research are also provided in this section.

2 Background

In this section we provide the background information necessary to understand the operation of the quantum-aided Weak Broadcast Protocol with and without noise, as well as the foundations it is based on.

2.1 Quantum Computing Basics

In contrast to classical computers, which are based on bits (a unit of data with value 0 or 1), quantum computers rely on qubits. A qubit is the basic unit of quantum information. It does not have a discrete value. Instead, it exists in a "superposition" of 0 and 1, denoted as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

In order to extract a classical value from a qubit, we need to *measure* it. Specifically, we measure it in the computational basis $\{|0\rangle, |1\rangle\}$, where

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle \ = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2}$$

For the general single-qubit state $|\psi\rangle$ of Eq. 1, this measurement has probability $|\alpha|^2$ of producing the value 0 and probability $|\beta|^2$ of producing the value 1. Since these are probabilities it holds that $|\alpha|^2 + |\beta|^2 = 1$. The process of measuring a qubit immediately collapses it, meaning it is no longer in superposition, but either $|0\rangle$ or $|1\rangle$.

Similarly to the classical logic gates, quantum gates can be used. Quantum gates are represented as $2^n \times 2^n$ unitary matrices that act on the states (qubits) to perform transformations. A quantum system can consist of multiple qubits. The protocol described used in our research relies on a four-qubit state. The qubits of a system and the gates applied to them are referred to as a *quantum circuit.*

Depending on the configuration of the quantum circuit, the qubits of the quantum system might become *entangled*, meaning that the probability of measuring 0 or 1 for each qubit is not independent. Instead, the measurement of one qubit provides information on the state of all the others, as the qubits are statistically correlated in a way that cannot be explained by classical means[2].

Consider now a scenario where a sender needs to send quantum information to a receiver over a network. Specifically, suppose the sender wants to send the qubit state of Eq. 1. This entails passing on information about α and β to the receiver. There exists a theorem in quantum mechanics which states that you cannot simply make an exact copy of an unknown quantum state. This is known as the no-cloning theorem[15]. As a result of this we can see that the sender cannot simply generate a copy of $|\psi\rangle$ and give the copy to the receiver. We can only copy classical states (not superpositions). However, by taking advantage of two classical bits and an entangled qubit pair, the sender can transfer its state $|\psi\rangle$ to the receiver. We call this teleportation because, at the end, the receiver will have $|\psi\rangle$ and the sender will not anymore[1]. In practice, teleportation is achieved by first creating an EPR pair shared between the sender and the receiver. An EPR pair is a maximally entangled two-qubit system, where the basis vectors are defined in terms of the computational basis as the four Bell states[11, 10]:

$$\begin{aligned}
|\Phi^+\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\
|\Phi^-\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \\
|\Psi^+\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \\
|\Psi^-\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)
\end{aligned} \tag{3}$$

Suppose we choose the EPR pair in our teleportation procedure to be $|\Psi^-\rangle$ from Eq. 3. The combined system of the state to be teleported $|\psi\rangle$ and the EPR pair is then

$$\begin{aligned}
|\psi_1\rangle \otimes |\Psi^-_{23}\rangle = (\alpha|0_1\rangle + \beta|1_1\rangle) \otimes \frac{1}{\sqrt{2}}(|01_{23}\rangle - |10_{23}\rangle) &= \frac{\alpha}{\sqrt{2}}(|0_1\rangle|0_2\rangle|1_3\rangle - |0_1\rangle|1_2\rangle|0_3\rangle) \\
&+ \frac{\beta}{\sqrt{2}}(|1_1\rangle|0_2\rangle|1_3\rangle - |1_1\rangle|1_2\rangle|0_3\rangle)
\end{aligned} \tag{4}$$

The subscripts $\{1, 2, 3\}$ here label the three qubits in the system, with 1 being the qubit to be teleported, 2 the sender's EPR qubit and 3 the receiver's EPR qubit. The state of Eq. 4 can also be expressed in terms of the four Bell states (Eq. 3), giving us

$$\begin{aligned}
|\psi_1\rangle \otimes |\Psi^-_{23}\rangle = \frac{1}{2}[&|\Psi^-_{12}\rangle(-\alpha|0_3\rangle - \beta|1_3\rangle) + |\Psi^+_{12}\rangle(-\alpha|0_3\rangle + \beta|1_3\rangle) \\
&+ |\Phi^-_{12}\rangle(\alpha|1_3\rangle + \beta|0_3\rangle) + |\Phi^+_{12}\rangle(\alpha|1_3\rangle - \beta|0_3\rangle)]
\end{aligned} \tag{5}$$

The sender performs a measurement on qubits 1 and 2 in the Bell basis (Eq. 3). It follows from Eq. 5 that the four possible measurement outcomes for qubits 1 and 2 are equally likely. Depending on the outcome, qubit 3 ends up in one of the following states:

$$\begin{aligned}
-\alpha|0\rangle - \beta|1\rangle \\
-\alpha|0\rangle + \beta|1\rangle \\
\alpha|1\rangle + \beta|0\rangle \\
\alpha|1\rangle - \beta|0\rangle
\end{aligned} \tag{6}$$

To complete the teleportation, the sender sends the outcome of the measurement to the receiver through a classical communication channel. The receiver then applies the correct quantum gates to bring qubit 3 to the the state of Eq. 1. Thus, qubit 1 collapses into some state and qubit 3 ends up in qubit 1's original superposition. This quantum teleportation protocol was first described by Bennett et al[3].

Another mathematical tool we will use in our simulation is Positive Operator-Valued Measure (POVM) measurement. POVM measurement is a formalism that is well adapted to the analysis of measurements where the main interest is the probabilities of the respective measurement outcomes and not the post-measurement state of the system[10]. Such is the case in our research, as our quantum system is measured only once and we do not interact with it after measurement. POVM measurement is defined by a set of Kraus operators $\{M_0, M_1, ..., M_{n-1}\}$ where n is the number of basis elements of the measurement. In our case, we are measuring in the computational basis $\{|0\rangle, |1\rangle\}$ so the set of Kraus operators consists only of M_0 and M_1. Kraus operators are $2^m \times 2^m$ matrices where m is the number of qubits in the system. A key property of Kraus operators is that they satisfy the completeness condition[2]:

$$\sum_{i}^{n} M_i^\dagger M_i = \mathbb{I} \tag{7}$$

After performing a POVM measurement on a state ρ, outcome i is observed with probability[2]:

$$\Pr(i) = \mathrm{Tr}(M_i \rho M_i^\dagger) \tag{8}$$

Here, the state ρ is expressed as a *density matrix*, which allows us to describe not only *pure*, but also *mixed states*. A pure state is a quantum system whose state $|\psi\rangle$ is known exactly. We can also describe systems whose state is not completely known. In such a system, the set of possible states paired with the probability of each state being the state of the system is called an *ensemble of pure states*. Following this definition, a mixed state is a mixture of the different pure states in the ensemble. A pure state $|\psi\rangle$ can be expressed as a density matrix by taking the outer product with itself[10]:

$$\rho = |\psi\rangle\langle\psi| \tag{9}$$

For a more detailed explanation of these and other quantum computing principles, refer to the textbook by Nielsen and Chuang[10].

2.2 The Weak Broadcast Protocol

The protocol we will be studying is described in [8]. It consists of 3 nodes, a sender (S) and two receivers (R_0 and R_1). S selects a value $x_s \in \{0, 1\}$ and the nodes attempt to reach consensus on x_s. The term "weak broadcast" refers to the fact that after termination, the nodes will either all agree on x_s, or choose to abort. We will denote this protocol as Weak Broadcast Protocol (WBC). The basis for WBC is the four-qubit singlet state:

$$|\psi\rangle = \frac{1}{2\sqrt{3}}(2|0011\rangle - |0101\rangle - |0110\rangle - |1010\rangle - |1001\rangle + 2|1100\rangle) \tag{10}$$

This is an entangled state which, after measurement, produces one of the six possible four-qubit outcomes according to the probabilities shown in Table 1.

Outcome	Probability
0011	1/3
0101	1/12
0110	1/12
1010	1/12
1001	1/12
1100	1/3

Table 1: The probability of observing each of the six valid four-qubit outcomes after measuring each qubit in Eq. 10

The four qubits are distributed as follows: S keeps the first two qubits, R_0 receives the third and R_1 receives the fourth. The distribution of the qubits is achieved through quantum teleportation. The nodes then attempt to reach consensus based on the measurement values of their individual qubits. Specifically, they asses whether the measurement statistics of their qubits are consistent with the expected statistics according to the probabilities of Table 1. A direct result of this evaluation method is that the more times this measurement and evaluation process is performed, the higher the chances of success (consensus or correct detection of faulty nodes). Thus, the nodes base their decisions on not one, but $m \in \mathbb{Z}^+$ four-qubit states. In order to reduce overhead - measured qubits require less overhead than qubits still in superposition - we create and distribute the m states sequentially. The protocol is also defined by two more parameters, $0 < \mu < 1/3$ and $1/2 < \lambda < 1$. It is shown in [8] that the protocol performs best when these parameters are kept close to their upper bounds. In order to produce consistent results and graphs comparable to those presented in [8], we choose $\mu = 0.272$ and $\lambda = 0.94$. The exact steps performed by the nodes in order to achieve weak broadcast are as follows:

1. *Invocation Phase:* S decides on a value x_s, which it sends to R_0 and R_1 as $x_0 = x_s$ and $x_1 = x_s$ respectively. S then creates the state in Eq. 10 and sends the third and fourth qubits to R_0 and R_1 respectively using quantum teleportation. All of the nodes measure their qubits and store the result. The creation, distribution and measurement process is repeated m times, so that each node has a list of m measurements. For each measurement pair $(q_{i,0}, q_{i,1})$, $0 \leq i < m$ in S's measurement list, S adds i to its check set σ_S if $q_{i,0} = q_{i,1} = x_s$. This check set is then sent to both receivers. Now each receiver has a measurement list M_j, a bit value x_j and a check set σ_j.

2. *Check Phase:* Both receivers check the data received by S. For each index i in σ_j, R_j checks that $M_j[i] \neq x_j$ (*Consistency Condition*). R_j then checks that $|\sigma_s| \geq T \equiv \lceil \mu \cdot m \rceil$ (*Length Condition*). If any one of the two conditions is violated, R_0 sets its output value value to "abort", $y_0 = \perp$, while R_1 sets an intermediate value to "abort", $\tilde{y_1} = \perp$. Otherwise $y_0 = x_0$ and $\tilde{y_1} = x_1$.

3. *Cross-calling Phase:* R_0 sends y_0 and σ_0 to R_1, which receives them as y_{01} and ρ_{01} respectively.

4. *Cross-check Phase:* R_1 evaluates the following three conditions: (i) *Confusion Condition:* $y_{01} \neq \tilde{y_1} \wedge y_{01} \neq \perp \wedge \tilde{y_1} \neq \perp$. (ii) *Length Condition:* $|\rho_{01}| \geq T \equiv \lceil \mu \cdot m \rceil$. (iii) *Consistency Condition:* $|L_1| \geq \lambda T + |\rho_{01}| - T$,where $\forall i \in \rho_{01}(M_1[i] \neq y_{01} \Leftrightarrow M_1[i] \in L_1)$. If all of the above conditions hold, then R_1 sets its output value to $y_1 = y_{01}$, otherwise $y_1 = \tilde{y_1}$.

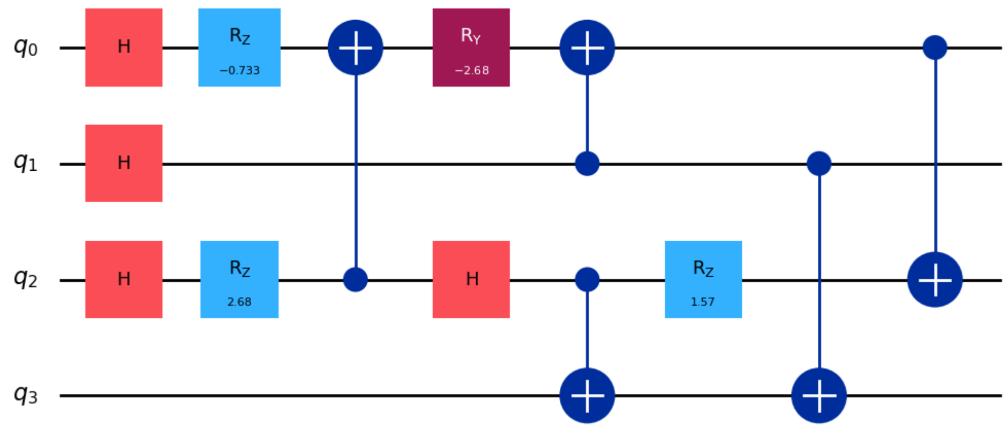

Figure 1: Loop Circuit, the quantum circuit used to create the state from Eq. 10. The fractions shown in the figure correspond to -0.73304, 2.67908 and 1.5708[8]

3 Methodology

In this section, we describe the methodology used to simulate the protocol in a quantum network under the influence of measurement errors. Specifically, we explain the creation of the four-qubit state and the Python implementation of our simulation. We also discuss quantum device specifics and define our noise model.

3.1 Qubit State Creation

There are many ways to create the four-qubit state described in Eq. 10 using a quantum circuit. In our research, we do not concern ourselves with errors produced when applying different gates, so the exact layout of the circuit is irrelevant, as long as the state in Eq. 10 is produced. For the purposes of this simulation, we chose the Loop Circuit proposed in [8]. The circuit consists of H, CNOT, R_Z and R_Y gates. The exact layout of the quantum circuit is shown in 1. The protocol does not specify who is responsible for creating the state of Eq. 10. For the purposes of this research it is irrelevant, since no measurement occurs during the creation of the state. Thus, we assume that the state is created by S.

3.2 Simulation and Device Specification

WBC was implemented using SquidASM. SquidASM is a simulator based on NetSquid that can execute applications written using NetQASM[13]. For the purposes of the simulation, the three nodes (S, R_0, R_1) were configured as quantum devices based on nitrogen-vacancy (NV) centers in diamond. The motivation behind this choice is two-fold. Firstly, NV centers in diamond are particularly well suited for the realisation of quantum networks due to their long decoherence times, even at room temperature[6, 14]. Secondly, NV centers are an important area of research at QuTech, especially in the Quantum Internet Division[12]. This will make this simulation more compatible with current QuTech research.

3.3 Noise Model - Measurement Errors

In this section, we present the approach that is used to simulate the effect of measurement errors. As mentioned in Section 3.2, we model S, R_0, R_1 as NV center devices. To do this, we use NetSquid's NV configuration, described in [5]. In this configuration, the measurement of a qubit is modeled as a POVM measurement with the Kraus operators

$$M_0 = \begin{bmatrix} \sqrt{f_0} & 0 \\ 0 & \sqrt{1-f_1} \end{bmatrix}, M_1 = \begin{bmatrix} \sqrt{1-f_0} & 0 \\ 0 & \sqrt{f_1} \end{bmatrix}$$

As mentioned in Section 2.1, when performing this POVM measurement on a single qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, the probabilities of producing a 0 or a 1 (Pr(0) and Pr(1) respectively) are:

$$\begin{aligned} \Pr(0) &= \mathrm{Tr}(M_0|\psi\rangle\langle\psi|M_0^\dagger) = \alpha^2 f_0 + \beta^2(1-f_1) \\ \Pr(1) &= \mathrm{Tr}(M_1|\psi\rangle\langle\psi|M_1^\dagger) = \alpha^2(1-f_0) + \beta^2 f_1 \end{aligned} \tag{11}$$

In essence, the result of this measurement operation is that a measurement outcome 0 has probability $1 - f_0$ of flipping to a 1, while a measurement outcome 1 has probability $1 - f_1$ of flipping to a 0. To see an example of how a faulty measurement can affect the failure probability of the protocol, consider the following run (for a full description of the protocol see Section 2.2):

1. *Invocation Phase:* S wants to send $x_s = 1$, which it sends to both receivers. S then creates the four-qubit state and sends the third and fourth qubits to R_0 and R_1 respectively using quantum teleportation. All of the nodes measure their qubits and store the result. The creation, distribution and measurement process is repeated m times, so that each node has a list of m measurements. We assume that no error occurred during S and R_0's measurements. However, the kth measurement of R_1 exhibited a measurement error, causing the outcome to be flipped from a 0 to a 1. S sends its check set to both receivers. Now each receiver has a measurement list M_j, a bit value x_j and a check set σ_j.

2. *Check Phase:* Both receivers check the data received by S. R_0's measurements were all correct (no errors occurred), so it successfully verifies the Consistency and Length Conditions and sets $y_0 = 1$. R_0 checks the Consistency Condition. When checking index k, it is expecting to find a measurement outcome 0 in the measurement list. However, since the outcome was flipped, a 1 is found instead. This violates the Consistency Condition and R_1 sets $\tilde{y_1} =\perp$. The Length Condition check is unaffected, but the previous violation is enough for R_1 to abort.

3. *Cross-calling Phase:* R_0 sends y_0 and σ_0 to R_1, which receives them as y_{01} and ρ_{01} respectively.

4. *Cross-check Phase:* R_1 evaluates the Confusion, Length and Consistency Conditions. The Length Condition remains unaffected. The Consistency Condition happpens to be satisfied since it is more "lenient" than the one in the Check Phase. However, the Consistency Condition is violated, since $\tilde{y_1} =\perp$.

These events result in a failure of the protocol, since R_1 chose to abort even though no other nodes exhibited faulty behaviour. In a more general sense, measurement errors cause the protocol to fail more because they alter the outcome statistics (see Table 1), on which the Conditions of WBC are based.

4 Experimental Setup and Results

In this section we present the experimental setup and derived results. First we simulate the WBC protocol described by Guba et al.[8] with no added noise and compare our produced failure probability. Subsequently, we will introduce "measurement error" noise and study its effect on the failure probability. Interested readers should be able to reproduce our experiments and obtain the same results. For all the experiments described below, the software versions used are as follows: Python 3.11.4, SquidASM 0.13.4, NetQASM 2.0 and NetSquid 1.1.8.

4.1 Experiment 1 - Noiseless WBC

In this experiment, we simulate WBC exactly as described in Section 2.2, with no added noise. The goal of this experiment is to assess whether our simulated three-node network behaves similarly to the Monte-Carlo simulations in [8]. In particular, we are interested in whether our failure probability matches the upper bounds calculated by the authors. In order to achieve this, we vary m in $[10, 400]$ and plot for each m value, both the upper bound calculated by Guba et al.[8] and the actual failure probability achieved by our simulation. In order to produce meaningful results, the simulation is repeated $N = 100$ times, the total number of failures N_f is counted and the failure probability is calculated as $\frac{N_f}{N}$.

Part of the experiment is assessing the failure probability for each possible case of Byzantine Fault with tolerance $t < n/2$. In a three-node network, this allows for up to one faulty node. Thus, we repeat the simulation process described above three times, once for the case of no faulty nodes, once for the case where S is faulty and once for the case where R_0 is faulty. R_1 never sends messages to the other nodes, so being faulty has no effect on the failure probability and we do not need to consider this case.

For the purposes of the simulation, specific behaviour has to be implemented for each faulty node. To that end, detailed strategies are proposed by Guba et al.[8] for both the S faulty and the R_0 faulty case. These strategies dictate which indices the faulty node will include in its check set (see Section 2.2) and aim to maximise the failure probability of the protocol, constituting a "worst-case scenario". They are defined by a domain, which is the set of measurement values for which the strategy can be employed. In our simulation, we utilize the same strategies.

For each case (no faulty, S faulty and R_0 faulty) the definition of failure is different. For the no faulty case, we assume success only if $x_S = y_0 = y_1 = 0$ or $x_S = y_0 = y_1 = 1$ and every other outcome is assumed to be a failure. If S is faulty, there are two events that are considered a failure of the protocol: (i) The measurement values of the m states are not in the domain of S's strategy. (ii) R_0 outputs $y_0 = 0$ while R_1 outputs $y_1 = 1$. If R_0 is faulty, the failure cases are: (i) The measurement values of the m states are not in the domain of R_0's strategy. (ii) R_0's output value does not match S's output value.

4.2 Experiment 2 - Noisy WBC

In this experiment, we expand upon Experiment 1 by incorporating measurement error noise into our simulation. Specifically, we use an NV center configuration with $f_0 = 0.95$ and $f_1 = 0.995$ (see Section 3.3). These values are representative of near-term hardware, meaning they are defined as expected to be achieved in the near future by NV hardware[5]. We then follow the same procedure described in Section 4.1 with the same parameters $m \in [20, 400]$ and $N = 100$. We again simulate each faulty case by following the same faulty strategies.

The authors of [5] also provide a function to calculate "improved" parameter values. Here, "improved" means the predicted value of a parameter supposing that all hardware parameters are improved by a factor of k. With that in mind, we repeat the noisy simulation with $3x$ and $10x$ improved (f_0, f_1), corresponding to $(f_0 = 0.983, f_1 = 0.9983)$ and $(f_0 = 0.995, f_1 = 0.9995)$ respectively.

4.3 Results - Experiment 1

The failure probability achieved by the simulated protocol is shown in Fig. 2 along with the upper bounds calculated by Guba et al.[8]. It can be observed that the failure probability follows the same trend as the upper bound (tends to 0 as m tends to infinity). Furthermore, the individual probabilities for each m value are match the corresponding upper bound with small deviations. This observation can be made in all three cases.

4.4 Results - Experiment 2

The failure probability achieved by the simulated protocol with the addition of measurement error noise is shown in Fig. 3 along with the upper bounds calculated by Guba et al.[8]. Specifically, Fig. 3 shows the failure probability achieved using the three different parameter types, namely near-term, $3x$ improved and $10x$ improved. It can be observed that for the no faulty and the R_0 faulty case, the failure probability does not follow the trend of the upper bounds for any parameter type. In fact, it is increasing instead of decreasing w.r.t. m. The improvement factor k does have an effect on the failure probability, with each larger improvement factors producing generally fewer failures. The lowest achievable failure probability in the no faulty case is $p_f = 0.25$ for $m = 40, k = 10$, while for the R_0 faulty case it is $p_f = 0.21$ for $m = 110, k = 10$. In the S faulty case, the failure probability is similar for every parameter type with no discernible "winner". For both the near-term and the improved parameters, the failure probability roughly follows the decreasing trend of the upper bounds and seems to converge to 0. However, much larger deviations are observed compared to the noiseless results of Experiment 1 (see Fig. 2).

5 Discussion

In this section, we analyze how the noise affects the failure probability of the protocol. In Section 4.3 we saw that the failure probability of our implementation of the protocol closely follows the theoretical upper bounds calculated in [8]. This is a good indication that our baseline model is correct, in the sense that it operates as described by Guba et al.[8]. In Section 4.4 we observed that by adding measurement error noise, the reliability of the

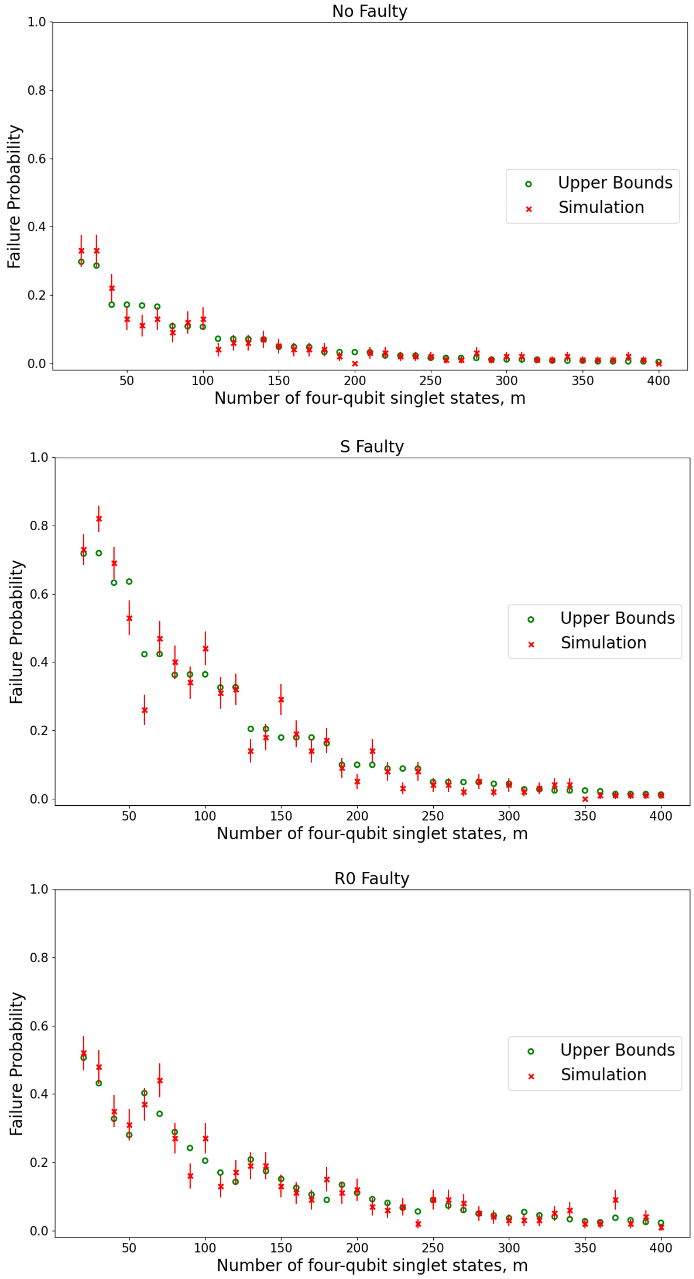

Figure 2: The failure probability achieved by our simulation for each faulty case in comparison to the upper bounds calculated by Guba et al.[8]

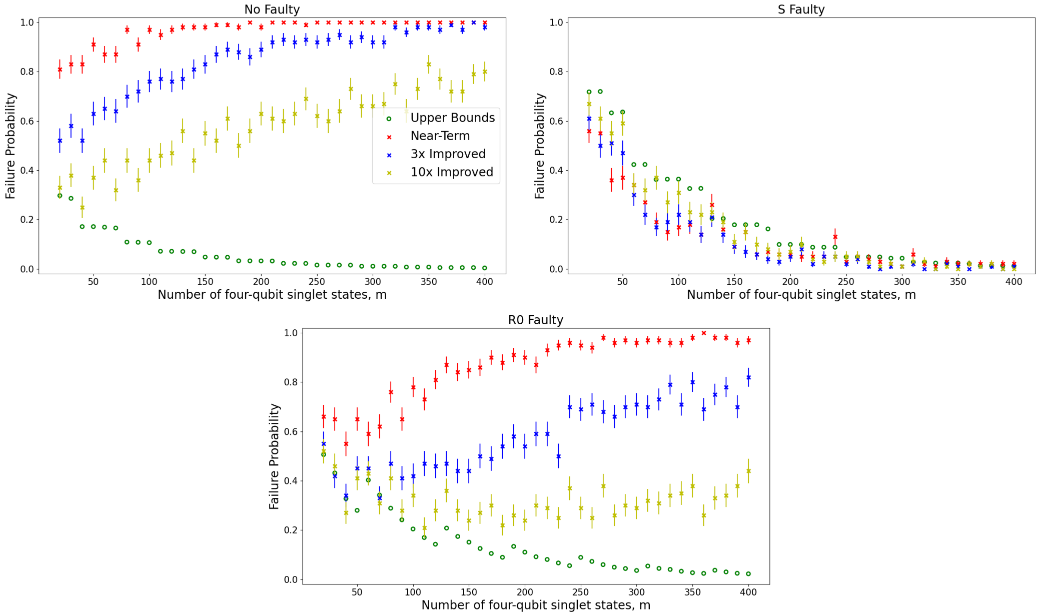

Figure 3: The failure probability achieved by our simulation under the influence of measurement error noise for each faulty case in comparison to the upper bounds calculated by Guba et al.[8]. The noise is applied with $f_0 = 0.95$ and $f_1 = 0.995$ in the near-term case, $f_0 = 0.983$ and $f_1 = 0.9983$ in the $3x$ case and $f_0 = 0.995$ and $f_1 = 0.9995$ in the $10x$ case

protocol changes drastically. In particular, we saw that the failure probability changed from a decreasing to an increasing trend in the no faulty and the R_0 faulty case.

In order to evaluate the reliability of WBC, we first consider the near-term parameters. In the R_0 faulty case, even if we base our conclusions on the lowest observed failure probability ($p_f = 0.55$ for $m = 40$), the protocol is expected to fail the majority of the time. In the no faulty case, the failure probability is even higher, with the lowest observed value being $p_f = 0.81$ for $m = 20$. For the improved parameters, the reliability is slightly improved, with the failure probability reaching $p_f = 0.21$ for $m = 110$ with $10x$ improved parameters in the R_0 faulty case and $p_f = 0.25$ for $m = 40$ with $10x$ improved parameters in the no faulty case. In the S faulty case, we observed the same declining trend as in the noiseless simulation. This is because, as shown in Section 3.3, Consistency Condition violations constitute a very common effect of the measurement errors. In the S faulty case, detection of a Consistency Condition violation is considered a success, thus resulting in reduced failure probability. This does not negate the high failure probability in the other cases, since in a real scenario no assumptions could be made regarding which node is more likely to be faulty. Whether a specific failure probability value is considered acceptable for a consensus protocol depends greatly on the context in which it is used. Nevertheless, failure probability of that magnitude observed in the no faulty and R_0 faulty cases would not be acceptable for most reliable protocols.

Based on our results, we can conclude that measurement error noise has a significant effect on the failure probability of WBC. In its current specification, the protocol would not succeed with an acceptable probability if it were implemented on real NV-based hardware within a reasonable time frame after the publication of the present research. In order to reduce the failure probability of WBC, we considered improved hardware parameters. However, our results also leave room for improvements to the protocol itself. The study of such possible improvements is outside the scope of this work.

6 Responsible Research

In this section, we reflect on and address ethical aspects of our research, as well as the reproducibility of our experiments and results. Regarding ethical concerns, the entirety of our data is derived either from [8] or from our own simulations. Therefore, we do not expect any privacy or data collection concerns.

Throughout this research, great importance has been placed on ensuring the reproducibility of our data. To that end, we have reported all the necessary software tools, including the versions used in our implementation (see Section 4). Furthermore, the full Python implementation has been uploaded to GitHub (see Section 8). In Section 4, we have aimed to include extensive details regarding the simulation setup, as well as all the relevant parameter values. Thus, we expect that any motivated reader is able to recreate our WBC implementation and, supposing that the same simulations are run, produce the same results. It is important to note that if the same setup is implemented using different software tools, for instance a different quantum network simulator, it is not guaranteed that the exact same outcomes will be observed. This is because the resulting failure probability depends on the NV center implementation of the given simulator. Nevertheless, Section 2.1 describes the mathematical tools used to design the simulated quantum network, including the teleportation and measurement procedures. This would allow motivated readers to implement their own quantum simulation with the same behaviour without the need to use the exact software libraries used in this work.

7 Conclusions and Future Work

In this work, we simulated the Weak Broadcast Protocol proposed by Fitzi[7] on a three-node quantum network. We showed that the simulation produces the results predicted by Guba et al.[8], when using the same parameters as the ones used by the authors. We then introduced measurement error noise of three levels, one corresponding to near-term hardware, one with $3x$ and one with $10x$ improved hardware parameters. Through these "noisy" simulations, we showed that measurement error noise has a significant effect on the failure probability of WBC. We concluded that the protocol would not succeed with an acceptable probability if it were implemented on real NV-based hardware within a reasonable time frame after the publication of the present research.

The results obtained throughout this research were focused on answering our research question. However, our work produced several intriguing questions that are worth exploring by future researchers. Firstly, the concept of physical noise was intentionally restricted to measurement errors. While this allows us to study the effect of this particular phenomenon, it does not produce a complete view of the protocol's behaviour on real devices. It would therefore be worthwhile to test WBC on a more generalized simulation model. Certainly, technological means permitting, it would be more valuable still to evaluate the protocol on an actual three-node network composed of real NV devices. Another factor that could affect the performance of the protocol is the number of nodes in the network. In this work, we based our results on a three-node network. If this quantum-aided WBC protocol is to be implemented in larger quantum networks, simulating with more nodes would provide valuable insight. This would, however, require modification of the steps described in Section 2.2 to extend the protocol to more nodes. Lastly, in this work we simulated the weak broadcast protocol introduced by Cabello[4], which is just one member of the family of weak broadcast protocols. Other similar protocols might produce different results, but more research is needed to confirm this.

Reflecting on our experimentation, we note that long simulation times constituted a major hindrance to productivity. Our simulation, when implemented using the software versions mentioned in Section 4, required several hours to complete for the three faulty cases when running on a commercial laptop with 8 CPU cores. Small tests showed that this time scales roughly linearly w.r.t. the number of samples. This would mean that running the simulation with increased sample count to reduce statistical error might require multiple days to complete. Therefore, we would encourage future researchers to consider using a High-Performance Computing (HPC) system to run similar simulations. The main advantage of the HPC system over a commercial computer would be the significantly increased core count, which could provide a major speedup to our multi-threaded approach (see Section 8).

8 Simulation Code

The Python implementation of our simulation is available on GitHub[9]. Interested readers are encouraged to replicate our findings and expand on our work.

References

[1] Amira Abbas, Stina Andersson, Abraham Asfaw, Antonio Corcoles, Luciano Bello, Yael Ben-Haim, Mehdi Bozzo-Rey, Sergey Bravyi, Nicholas Bronn, Lauren Capelluto,

Almudena Carrera Vazquez, Jack Ceroni, Richard Chen, Albert Frisch, Jay Gambetta, Shelly Garion, Leron Gil, Salvador De La Puente Gonzalez, Francis Harkins, Takashi Imamichi, Pavan Jayasinha, Hwajung Kang, Amir H. Karamlou, Robert Loredo, David McKay, Alberto Maldonado, Antonio Macaluso, Antonio Mezzacapo, Zlatko Minev, Ramis Movassagh, Giacomo Nannicini, Paul Nation, Anna Phan, Marco Pistoia, Arthur Rattew, Joachim Schaefer, Javad Shabani, John Smolin, John Stenger, Kristan Temme, Madeleine Tod, Ellinor Wanzambi, Stephen Wood, and James Wootton. *Learn Quantum Computation Using Qiskit.* 2020. URL: `https://weblab.tudelft.nl/docs/qiskit/qiskit.org/textbook/ch-algorithms/teleportation.html`.

[2] Rotem Arnon-Friedman. "Reductions to IID in Device-Independent Quantum Information Processing". ETH Doctoral Thesis. PhD thesis. Zurich, Switzerland: ETH Zurich, 2018. DOI: `https://doi.org/10.3929/ethz-b-000298420`.

[3] Charles H. Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K. Wootters. "Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels". In: *Physical Review Letters* 70.13 (Mar. 1993), pp. 1895–1899. DOI: `https://doi.org/10.1103/PhysRevLett.70.1895`.

[4] Adán Cabello. "Solving the liar detection problem using the four-qubit singlet state". In: *Phys. Rev. A* 68 (1 July 2003). DOI: `https://doi.org/10.1103/PhysRevA.68.012304`.

[5] Tim Coopmans, Robert Knegjens, Axel Dahlberg, David Maier, Loek Nijsten, Julio de Oliveira Filho, Martijn Papendrecht, Julian Rabbie, Filip Rozpędek, Matthew Skrzypczyk, Leon Wubben, Walter de Jong, Damian Podareanu, Ariana Torres-Knoop, and David Elkouss. "NetSquid, a NETwork Simulator for QUantum Information using Discrete events". In: *Communications Physics* 4 (2021), p. 164. DOI: `https://doi.org/10.1038/s42005-021-00647-8`.

[6] Suzanne B. van Dam, Michael P. Walsh, Maarten J. Degen, Eric Bersin, Sara L. Mouradian, Airat Galiullin, Maximilian Ruf, Mark IJspeert, Tim Hugo Taminiau, Ronald Hanson, and Dirk R. Englund. "Optical coherence of diamond nitrogen-vacancy centers formed by ion implantation and annealing". In: *Phys. Rev. B* 99 (16 Apr. 2019), p. 161203. DOI: `https://link.aps.org/doi/10.1103/PhysRevB.99.161203`.

[7] Matthias Fitzi. "Generalized Communication and Security Models in Byzantine Agreement". Reprint as vol. 4 of ETH Series in Information Security and Cryptography, Hartung-Gorre Verlag. PhD thesis. ETH Zurich, 2003. ISBN: 3896498533.

[8] Zoltán Guba, István Finta, Ákos Budai, Lóránt Farkas, Zoltán Zimborás, and András Pályi. "Resource analysis for quantum-aided Byzantine agreement with the four-qubit singlet state". In: *Quantum* 8 (2024). DOI: `https://doi.org/10.22331/q-2024-04-30-1324`.

[9] Kimon Kyparos. *Noizy Byzantine Agreement.* 2025. URL: `https://github.com/kypk/noizy_byzantine_agreement`.

[10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information.* 10th Anniversary Edition. Cambridge, UK: Cambridge University Press, 2010. ISBN: 9781107002173.

[11] Quantiki. *Bell Basis.* URL: `https://www.quantiki.org/wiki/bell-basis`. (accessed: 21.06.2025).

[12] QuTech. *Optical Engineering.* URL: `https://qutech.nl/expertise/optical-engineering/`. (accessed: 24.05.2025).

[13] QuTech. *Welcome to SquidASM's Documentation!* URL: `https://squidasm.readthedocs.io/en/latest/index.html`. (accessed: 24.05.2025).

[14] Bao-Cang Ren, Guan-Yu Wang, and Fuguo Deng. "Universal hyperparallel hybrid photonic quantum gates with dipole-induced transparency in the weak-coupling regime". In: *Physical Review A* 91 (Mar. 2015), p. 032328. URL: `https://doi.org/10.1103/PhysRevA.91.032328`.

[15] W. K. Wootters and W. H. Zurek. "A single quantum cannot be cloned". In: *Nature* 299 (Oct. 1982), pp. 802–803. DOI: `https://doi.org/10.1038/299802a0`.

[16] Weiyu Zhong, Ce Yang, Wei Liang, Jiahong Cai, Lin Chen, Jing Liao, and Naixue Xiong. "Byzantine Fault-Tolerant Consensus Algorithms: A Survey". In: *Electronics* 12.18 (2023), p. 21. DOI: `https://doi.org/10.3390/electronics12183801`.