Noisy Byzantine Agreement in Quantum Networks

Impact of Gate Errors on a Weak Broadcast Protocol

Alexandru-Andrei Cirjaliu-Davidescu[1]
Supervisor(s): 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: Alexandru-Andrei Cirjaliu-Davidescu
Final project course: CSE3000 Research Project
Thesis committee: Tim Coopmans, Arie van Deursen

Abstract

The Weak Broadcast protocol is a quantum solution to the Byzantine agreement problem. However, its practical applicability remains uncertain due to the impact of noise in quantum systems. Building on prior work by Guba et al., which presented a quantum Byzantine agreement protocol using a four-qubit singlet state, this research investigates how gate-level noise affects the success probability of the protocol, focusing on a Linear Circuit implementation. Gate noise is a critical challenge in quantum hardware, as quantum gates are a primary source of errors in current devices. Using the NetSquid and SquidASM frameworks, the protocol was reproduced and extended with a realistic depolarizing noise model to simulate noisy conditions across different scenarios. Results show that even modest noise levels ($0.001\% - 0.01\%$) lead to a sharp rise in failure probability, and at 1% noise, the protocol fails often. These findings highlight the protocol's vulnerability to gate noise and suggest that its practical deployment would require significant error mitigation and fault-tolerant architectures, or a design adapted to better tolerate gate-level noise. This work offers a reproducible simulation framework and provides insights into the protocol's robustness under noisy conditions, addressing a gap in current literature.

1 Introduction

Imagine a group of generals surrounding a city - some loyal, some traitors - trying to agree on a plan of attack, yet only able to communicate through messengers. This metaphor, introduced in 1982, still defines one of the hardest problems in distributed computing today [1]. A reliable computer system should be able to cope with the failure of one or more of its components [2]. This class of failure is abstracted by the Byzantine Generals Problem.

A Byzantine agreement requires a set of parties in a distributed environment to agree on a value even in the presence of faulty or malicious actors. To address this challenge, several quantum approaches to Byzantine agreement have been devised, such as the Byzantine Reliable Broadcast [3] or the Dolev-Strong Broadcast [4]. While a number of classical solutions to the Byzantine agreement problem exist, they are often complex or computationally expensive [5]. One notable study by Zoltan Guba et al. [6] presents a quantum implementation of a Weak Broadcast protocol, using two different circuits developed on the IBM Q and IonQ backends [6], to address limitations of classical variants.

Although the protocol has been studied in detail, its robustness to physical noise, particularly gate errors and channel imperfections, remains underexplored. In their future outlook, Guba et al. [6] explicitly highlight the need for research into the challenges posed by physical errors in the source, quantum channels, and measurement devices, noting that quantitative investigations in this area are lacking [6]. This raises important questions about the practical usability of such protocols.

This research aims to address that gap by further emphasising the impact of noise - specifically gate errors - on the performance of Guba et al.'s Linear Circuit implementation [6]. The goal is to evaluate how gate-level noise affects the success probability of the protocol, to examine whether its performance remains viable under realistic conditions, and to determine if a broader adoption of the Weak Broadcast protocol is supported. This qualitative assessment will be based on quantitative results from simulations of the Linear Circuit implementation under varying gate error rates.

The remainder of this report is structured as follows. Section 2 provides the necessary theoretical background, including an overview of the Weak Broadcast protocol and the Linear Circuit proposed by Guba et al [6]. Section 3 emphasises the motivation and significance of the study, while Section 4 outlines the procedure used to address the research tasks. The experimental setup and noise model are described in more detail in Section 5, while the simulation results are presented in Section 6. Section 7 discusses the results in relation to existing literature and evaluates the practical implications of deploying the protocol in real-world quantum networks. Responsible research aspects of this work, such as reproducibility and ethical considerations, are reflected upon in Section 8. Finally, Section 9 concludes the study and outlines directions for future research.

2 Background

There are multiple approaches to implementing a Weak Broadcast Protocol and its circuits, so it's essential that details are provided for transparency. This section is dedicated to outlining the foundational concepts, defining key terms, and presenting an overview of the relevant protocols and circuit designs.

2.1 Brief Introduction to Quantum Computing

Quantum computing relies on principles of linear algebra and quantum mechanics to process information in fundamentally different ways from classical computing. For this study, only a minimal understanding of the formalism is required. An n-qubit quantum state can be represented as a complex vector of dimension 2^n, where each basis element corresponds to a possible classical configuration of the qubits. Quantum gates are unitary operations on these states and are expressed as $2^n \times 2^n$ unitary matrices, ensuring the preservation of probability outcomes. The evolution of a quantum system is determined by the sequential application of such gates.

A fundamental feature of quantum systems is entanglement, where the state of one qubit cannot be described independently of another. In entangled systems, measuring one qubit instantaneously affects the possible outcomes of measurements on the others. This property forms the basis for many quantum communication protocols. Measurement involves projecting a quantum state onto a classical computational basis, typically $|0\rangle, |1\rangle$. Upon measurement, the quantum state probabilistically collapses to one of the basis states. Measurement reveals and irreversibly modifies the quantum state, making it a key operation in extracting classical information from quantum systems.

For a more comprehensive introduction to the mathematical foundations of quantum computing, please refer to standard quantum computing textbooks such as *Quantum Computation and Quantum Information* by Nielsen and Chuang [7].

2.2 Quantum Teleportation

The generation of the quantum state used in the protocol requires the use of quantum teleportation. Quantum teleportation is a foundational quantum communication protocol that

allows the transfer of a qubit state from one party to another without physically transmitting the qubit itself. First proposed by Bennett et al. [8], the process relies on entanglement and classical communication. Briefly, the process is the following: (1) two quantum particles are entangled at the location where the teleportation will begin; (2) the sender performs a measurement on the qubit to be teleported and one half of an entangled pair, after which the measurement result is sent via a classical channel to the receiver; (3) the receiver, using the other half of the entangled pair, applies a corresponding correction based on the received results, effectively recreating the original quantum state.

2.3 Broadcast and Weak Broadcast

Lamport et al. [2] introduced the term "Byzantine Agreement" to refer to the challenge of achieving consensus in spite of the presence of faulty or malicious participants. A broadcast problem is characterised by a protocol involving n parties, including a designated sender, that satisfies two key properties: (1) *agreement* - if any honest party outputs a value, then all honest parties must also output that value; (2) *validity* - If the sender is honest, then all honest parties must output the sender's original value.

Various solutions have been developed in the classical setting to address this challenge, including the Byzantine Reliable Broadcast [3] and the Dolev-Strong Broadcast protocol [4], both of which tolerate up to one-third of the participants being faulty or malicious.

In 1983, Lamport [9] introduced the idea of "weak broadcast", which, in addition to satisfying the agreement property, divided and extended validity into two further properties: (1) *weak validity* - if the sender is honest, then all honest parties output either the sender's value or abort; (2) *non-triviality* - if all parties are honest, then all parties output the sender's value. Weak broadcast allows protocols to detect and avoid faulty behaviour without forcing consensus at all costs, offering the option to abort early rather than agree on bad data.

2.4 Weak Broadcast Protocol for Byzantine Agreement

In their work, Guba et al. [6] introduced a quantum variation of a Weak Broadcast protocol, capable of tolerating up to half of the participants being faulty or malicious - a notable advancement over classical counterparts. Their study outlines an implementation of a three-party communication scheme, utilising both a classical and a quantum channel. This implementation, henceforth referred to as WBC(3,1) following the nomenclature in their paper, employs a 4-qubit quantum singlet state.

The structure of the 4-qubit singlet state used in the protocol can be represented using a classical probability histogram of measurement outcomes, such as the one in Figure 1, which is based on 12 measurements. The state consists of six non-zero basis states, with two of them ($|0011\rangle$ and $|1100\rangle$) having a higher probability amplitude, and the remaining four basis states ($|0101\rangle$, $|0110\rangle$, $|1001\rangle$, $|1010\rangle$) each occurring with a lower probability. This reflects the entangled nature of the state, while offering a more intuitive visual representation of its structure for readers unfamiliar with Dirac formalism.

In this setup, represented in Figure 2, qubits are distributed such that the sender holds two qubits, while each of the two receivers holds one qubit. Furthermore, the protocol is

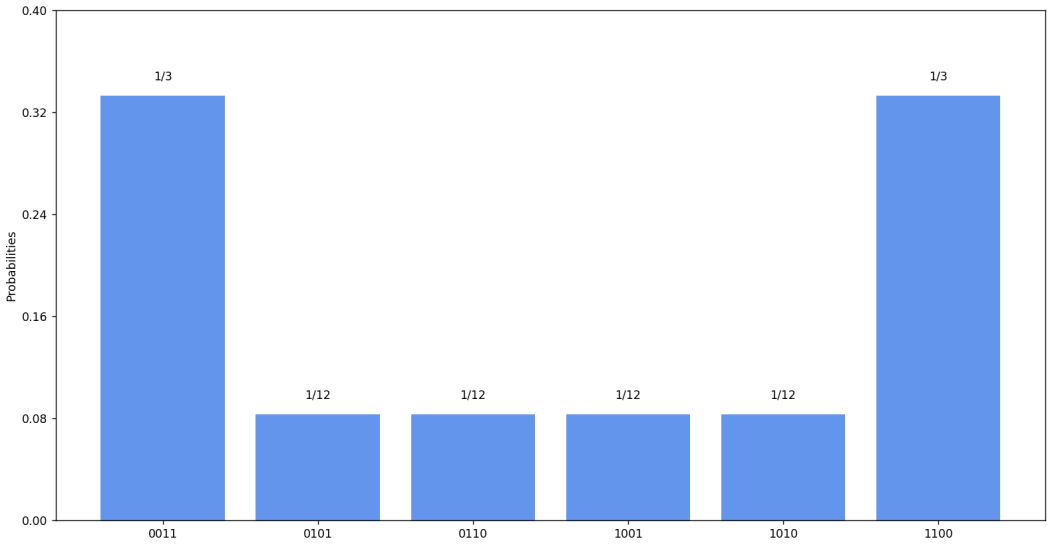

Figure 1: Probability distribution of the 4-qubit singlet state based on 12 measurements, of the four-qubit singlet state used in the WBC(3,1) protocol. The state includes six computational basis outcomes with varying probabilities.

characterised by three key parameters. The first two, μ and λ, play a critical role in defining the operational thresholds. μ controls how many entangled states need to pass a consistency check before a receiver trusts the sender's data, determining how strict the protocol is in accepting a message. λ is used by the second receiver (R_1) to decide whether to trust the other receiver (R_0)'s forwarded message; if too many inconsistencies are found, R_1 doesn't trust R_0. The third, m, represents the number of individual 4-qubit quantum states used to send a single bit of data. It provides redundancy, as the more states the protocol uses, the better the chances of detecting tampering or noise. A higher m can make the protocol more reliable, but also more resource-intensive.

Once the quantum state has been built and the classical channel established, the WBC(3,1) can begin, being composed of 4 classically computed phases:

1. Invocation Phase: Sender S sends their data bit x_s to each receiver R_0 and R_1, who receive values denoted as x_0 and x_1. Now, the sender measures all its corresponding qubits (m pairs of two qubits). For each pair, if both measurements yield x_s, S adds the state's index to a check set σ_s. After assembling, the sender forwards this check set to each receiver, then sets its own output. This phase ensures the sender transmits their information to each receiver.

2. Check Phase: Each receiver checks their received data based on two conditions:

   (a) Consistency: After measuring all the qubits corresponding to indexes in their received check set, all results must differ from the received data bit.

   (b) Length: The check set must have a size of at least $\lceil \mu \cdot m \rceil$, where $0 < \mu < 1/3$.

   If any of those conditions are violated, communication has been tampered with and the receiver outputs an abort $\perp$. Let y_0 represent the output chosen by R_0 (so $y_0 = x_0$

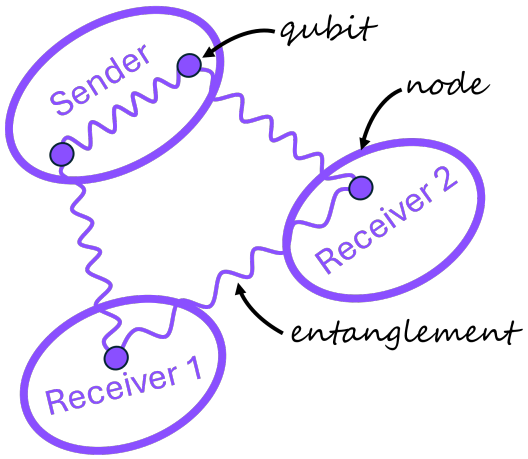

Figure 2: Representation of a 4-qubit node network with 3 nodes

or $y_0 = \perp$), while R_1 stores an intermediate value $\tilde{y}_1$ (so $\tilde{y}_1 = x_1$ or $\tilde{y}_1 = \perp$). This phase allows receivers to catch a malicious sender early by performing checks on the received data.

3. Cross-calling Phase: R_0 sends to R_1 its output y_0 and the check set received from the sender σ_0. R_1 stores those values as y_{01} and ρ_{01} respectively. This phase ensures the first receiver forwards their data to the second receiver.

4. Cross-check Phase: R_1 checks for three conditions:

   (a) Confusion: y_{01} is different from $\tilde{y}_1$, and neither are aborts $\perp$.

   (b) Length: Size of check set ρ_{01} is at least $\mathrm{T} = \lceil \mu \cdot m \rceil$, where $0 < \mu < 1/3$.

   (c) Consistency: The count of indices in ρ_{01} where R_1 measures the bit opposite to y_{01} is greater than or equal to $\lambda T + |\rho_{01}| - T$.

   If all three conditions are met, the receiver outputs the received value y_{01}. Otherwise, it outputs their intermediary value $\tilde{y}_1$. This phase allows the second receivers to catch a malicious receiver by performing checks on the received data.

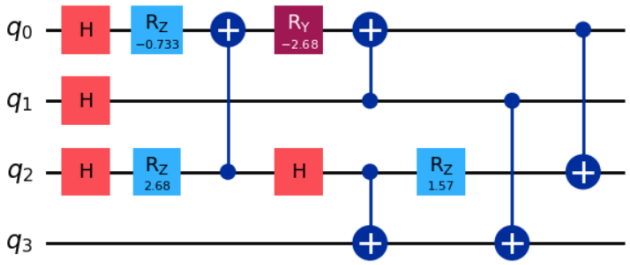

Figure 3: Loop Circuit, a 5-CNOT circuit that prepares the four-qubit singlet state as per specification in Guba et al.[6]

The WBC(3,1) protocol was implemented on two quantum computing backends: IonQ and IBM Q. For each backend, a tailored circuit design was developed - a "Loop Circuit" for IonQ (figure 3) and a "Linear Circuit" for IBM Q (figure 4). The primary difference between these designs lies in their complexity: the Linear Circuit is longer and utilises a greater number of quantum gates, particularly two-qubit CNOT gates, compared to the more compact alternative. Furthermore, to verify the correctness of the protocol, three adversary models were designed: (1) *no* faulty, in which no party acts maliciously and sends honest data; (2) S faulty, in which the sender is malicious and will send conflicting data to the two receivers; (3) R_0 faulty, in which one of the senders is malicious and attempts to manipulate the other sender into outputting a specific value, different from the sender's value. The R_1 faulty case is considered functionally equivalent to the R_0 faulty case.

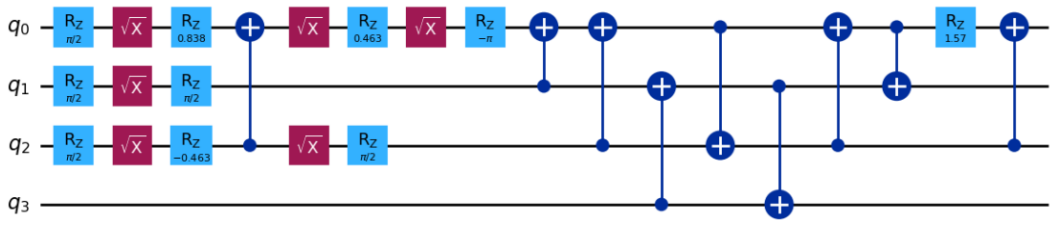

Figure 4: Linear Circuit, a 9-CNOT circuit that prepares the four-qubit singlet state as per specification in Guba et al.[6]

2.5 Depolarization Noise Model

To evaluate the resilience of the WBC(3,1) protocol under realistic conditions, we incorporated a depolarizing noise model [10]: with a fixed probability p, a depolarizing channel was applied after each gate operation. This simulates the complete randomisation of a qubit's state, representing a general model of noise.

Depolarization is a common quantum noise model that represents the loss of information in a qubit due to interactions with its environment. In a depolarizing channel, a qubit's state is replaced with a maximally mixed state with some probability p, and remains unchanged with probability $1 - p$. Mathematically, this process is described as:

$$\rho \to (1-p)\rho + \frac{p}{3}\left(X\rho X + Y\rho Y + Z\rho Z\right)$$

where ρ represents the original qubit state (as a density matrix), and X, Y and Z are the Pauli matrices, matrices that form a fundamental basis for quantum operations. This model effectively introduces random Pauli errors (bit-flips X, phase-flips Z, or both Y) with equal probability, making it a useful abstraction for simulating generic gate imperfections.

3 Problem Statement

This section presents the central research problem addressed in this study, focusing on the practical limitations of quantum Byzantine agreement protocols under realistic conditions.

It introduces the motivation behind the research, identifies specific challenges posed by quantum noise, and outlines the primary goal and objectives of the project.

Quantum Byzantine agreement protocols, such as the Weak Broadcast protocol proposed by Guba et al. [6], offer improvements over classical counterparts by tolerating a higher number of malicious participants. However, these advantages come with a cost. Qubits are more delicate than classical bits and highly prone to errors [11], so in practice, current quantum hardware is characterised by significant imperfections [12]. These issues raise important questions regarding the implementation of such protocols on real-world quantum devices.

Physical gate-level noise presents a major threat to the reliability of protocols, as small imperfections in gate operations can accumulate rapidly and disrupt their outputs. These effects directly impact protocols dependent on entanglement, essential for quantum consensus such as the WBC(3,1), by causing state loss in states generated through multi-gate circuits [13], such as the Linear Circuit.

While Guba et al. [6] introduced a circuit-level implementation of the Weak Broadcast protocol and established analytical upper bounds on protocol failure by modelling outcome-level leakage, their approach did not account for specific hardware-induced noise sources such as gate errors. Additionally, they highlighted in their conclusions section the need for further analysis under physical noise conditions.

This research addresses that gap by investigating how depolarizing gate noise affects the performance of the Linear Circuit implementation of the protocol, providing a practical perspective on how real-world quantum hardware affects protocol viability. The primary objective is to determine whether the protocol remains functionally viable under realistic noise conditions and to what extent noise undermines its correctness. This study aims to evaluate the resilience of the protocol against gate depolarization noise.

Therefore, this study aims to answer the following question: *How does gate-level noise affect the Linear Circuit implementation of the WBC(3,1) Weak Broadcast protocol?* To achieve an answer, the study will investigate the following research sub-questions:

1. How does the probability of protocol failure vary as a function of gate-level depolarizing noise in the Linear Circuit implementation of the WBC(3,1) protocol?
2. What is the minimum noise threshold at which the WBC(3,1) protocol becomes unreliable for practical use, and how does this compare to noise levels typically observed in current quantum hardware?

Addressing these questions necessitates a dedicated experimental approach, which will be described in the following sections.

4 Methodology

To assess the viability of the WBC(3,1) protocol, an experimental study was conducted. This section will present the methodological approach used to answer the question outlined above.

To implement noise models and assess their effects on the WBC(3,1) protocol, a faithful

reproduction of the protocol is necessary. To achieve this goal, this study will focus on reproducing and further extending the Linear Implementation of the protocol, represented in Figure 4. The protocol will be reproduced in a Python-based simulation environment, leveraging two quantum computing frameworks: NetSquid [14], a software tool for the modelling and simulation of scalable quantum networks developed at QuTech, and SquidASM [15], a higher-level simulator based on NetSquid that can execute applications written using NetQASM. The protocol will be extended with gate depolarization noise, allowing an injection of gate-level imperfections to evaluate their impact on protocol performance under various adversarial conditions.

This simulation-based approach enables controlled testing across different error scenarios, forming the basis for the noise-resilience analysis. The methodology of this study consists of three stages: (1) implementation of the Linear Circuit WBC(3,1) protocol in simulation, (2) injection of noise models into quantum operations, and (3) tracking the failure probability across varying parameters and performing qualitative analysis. The specific design of these experiments and the results of this analysis will be presented in further sections.

5 Experimental Setup

This section outlines the simulation environment used to ensure the reproducibility of results. It will detail the process of recreating the Linear Circuit implementation and describe the noise model used for reliability analysis.

The experiments were conducted in a fully simulated environment to allow controlled testing and ensure reproducibility. All simulations were performed using SquidASM [15], a high-level quantum network simulator package for Python, built on top of NetSquid. NetSquid [14] is a discrete-event simulation framework designed for modelling and analysing quantum networks, while SquidASM provides a higher-level interface for running quantum applications written in NetQASM, including support for gate-level operations, quantum memory, and custom noise models.

The experiments were executed on a local machine running Windows 10 and Linux through the Windows Subsystem for Linux. The implementation environment used Python 3.12, along with some relevant libraries and packages such as *numpy*, *matplotlib*, *math*, and *random*. The simulations focused exclusively on the Linear Circuit version of the WBC(3,1) protocol as described in Section 2. An implementation of the protocol, including additional utilities and extensions, is available in a GitHub repository at `https://github.com/AlexDC-2003/research_project_noisy_wbc.git`.

To assess the effect of gate-level noise on the WBC(3,1) protocol, multiple simulations of the protocol enhanced with noise were conducted for a fixed number of singlet states $m = 280$ and parameters $\mu = 0.3$ and $\lambda = 0.94$. These parameter values are similar to values recommended by Guba et al. as they lie in the protocol's optimised region, ensuring minimal quantum resource consumption while maintaining a provably low failure probability of around 5% [6]. For each adversary configuration, the protocol was executed for 12 noise error rate values, and the failure probabilities were collected from 500 runs per setting. The simulation set was performed for each of the adversary configurations (*no* faulty, S faulty, and R_0 faulty, as described in Section 2). Afterwards, data was collected and plots were

generated for easy visualization.

The depolarization noise was implemented in SquidASM via modifications to the configuration file, focusing on the noise error that affects two-qubit gates specifically, such as the CNOT gates that are used by the Linear Circuit implementation. By allowing for gate depolarization, the respective gate operations now probabilistically have an error injection applied to the configured noise profile. This approach easily integrates into SquidASM's execution of quantum programs, while the chosen model reflects well recurring imperfections observed in contemporary devices, where the fidelity of gates is limited by noise, crosstalk, and hardware variance. By modelling common error sources, it captures failure behaviours that would arise in realistic scenarios. Furthermore, no noise was applied during state preparation or measurement in order to isolate the impact of gate-level imperfections on performance.

6 Results

This section will present the results obtained after simulations, that will serve as the basis for the discussion regarding protocol viability.

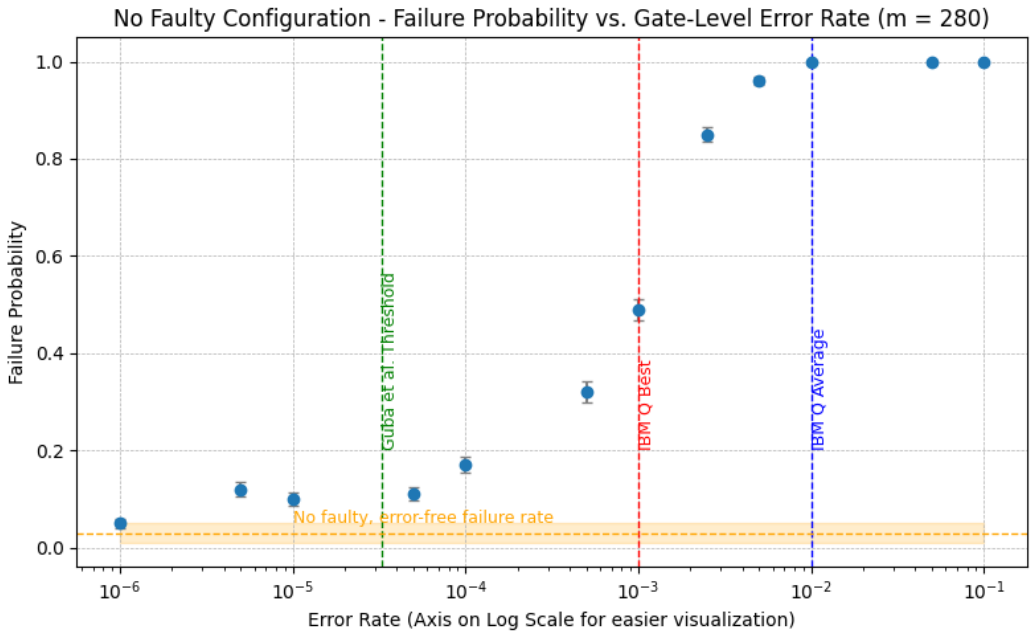

Figure 5: Failure Probability vs. Depolarization Probability (No Faulty configuration)

The simulations were performed based on each adversary configuration. Each plot presents the resulting failure probability as a function of the noise level. Each data point reflects the average of runs at the given error rate, with error bars representing the standard deviation across trials. The x-axis is presented on a logarithmic scale to capture the steep transition

occurring. The vertical dashed lines represent various realistic thresholds for acceptable error rates. The green vertical line at $x = 3.3 \times 10^{-5}$ marks a reference threshold from Guba et al.'s study [6], under which the additional failure, determined strictly due to errors and not the protocol's general failure, remains below 1%. The red and blue vertical lines, at $x = 0.001$ and $x = 0.01$ respectively, represent two thresholds for IBM Q hardware, averaged over two chips, Heron R2 and Eagle R3 [16]. The horizontal orange dashed line presents the respective baseline failure rate observed in the noise-free case for $m = 280$ for each configuration, as observed in the original study [6].

Figure 5 presents the failure probability of the WBC(3,1) protocol under the "no faulty" adversary configuration, where no party acts maliciously or sends wrong data. The baseline error-free failure rate is drawn in orange, while vertical markers indicate reference noise thresholds from Guba et al. and IBM Q hardware platforms. In this configuration, failure is defined as any disagreement among honest parties, either due to mismatched outcomes or aborts [6].

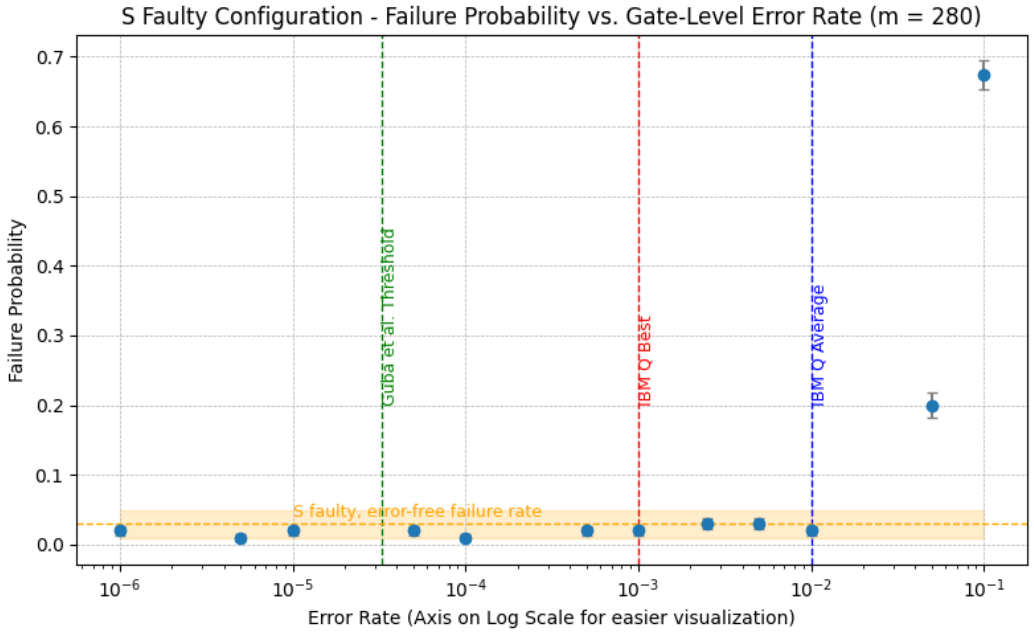

Figure 6: Failure Probability vs. Depolarization Probability (S Faulty configuration)

Figure 6 shows the failure probability of the WBC(3,1) protocol under the "S faulty" adversary configuration, where the sender intentionally sends opposing data to each receiver. Vertical and horizontal guidelines are retained to align with the previous case, enabling a clear comparison between adversarial models and noise sensitivity. Failure occurs either when the sender is unable to produce a sufficiently large check set to pass the check phase, or when the two receivers disagree on their outputs [6]. Notably, aborts from either receiver are considered a valid outcome and do not count as failures, as opposed to the other configurations.

Figure 7 presents the failure probability of the WBC(3,1) protocol under the "R_0 faulty" adversary configuration, where the first receiver manipulates the data sent to its peer to reach a specific outcome. The same thresholds are maintained across all plots for consistency in comparison, with the orange baseline line reflecting the protocol's performance in the absence of noise under this adversary setting. Failure is defined as disagreement between the sender's output and that of the second receiver.

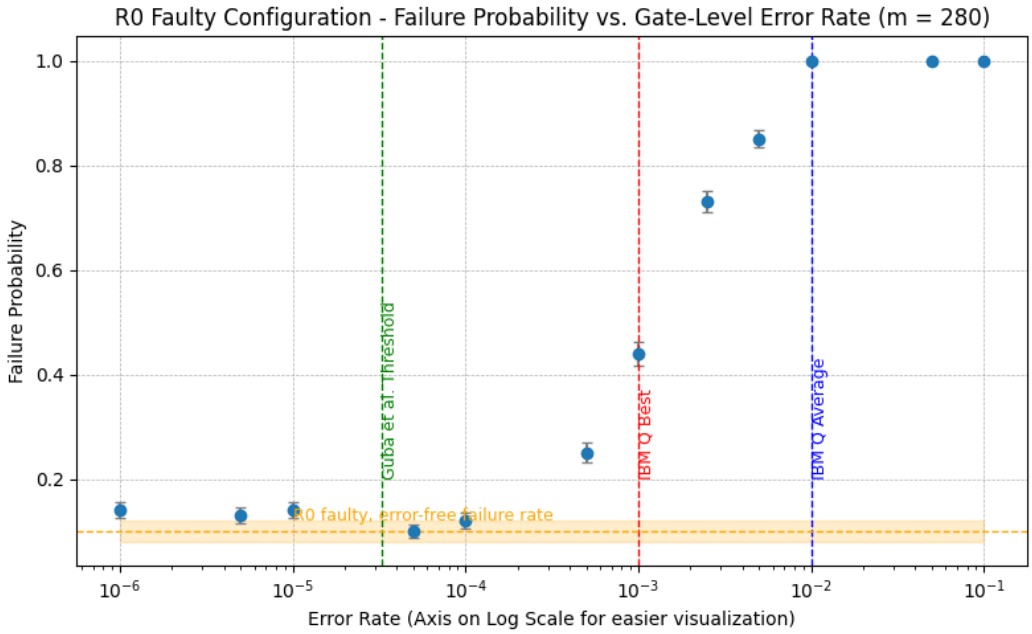

Figure 7: Failure Probability vs. Depolarization Probability (R_0 Faulty configuration)

Collectively, these plots illustrate how the protocol responds to varying levels of gate-level depolarizing noise across all adversarial configurations. The data presents a sharp threshold around noise error rates 10^{-4} and 10^{-3}, especially in the no faulty and R_0 faulty scenarios, suggesting the protocol is highly sensitive to gate-level imperfections. Across the three configurations, the S faulty case is notably more resilient, with failure probability remaining near baseline until error rates exceed 10^{-2}. In contrast, the no faulty and R_0 faulty cases show a much steeper increase, failing consistently at error rates as low as 10^{-3}. This highlights that the protocol is more vulnerable to receiver faults and general noise than to sender dishonesty. A more detailed interpretation of trends and practical implications is provided in the Discussion section.

7 Discussion

This section analyses the simulation results in light of the research questions and theoretical context introduced earlier, how the introduction of gate-level noise affects the failure

probability of the WBC(3,1) protocol, and how these results compare against the noise-free baseline and prior literature.

Figures 5 and 7, corresponding to the no faulty and R_0 faulty configurations, show that the failure rate remains near baseline for very low noise levels but increases sharply around a depolarizing rate of 10^{-3}. At noise rate 0.01 (1% noise), the protocol fails with near 100% probability across all adversary configurations. This signifies a clear breakdown of protocol correctness under modest noise.

In contrast to the baseline case, where failure remains below 5% for $m = 280$, introducing noise as low as 10^{-3} pushes failure rates around 50%. While Guba et al. estimated that outcome leakage must stay below 3.3×10^{-5} to maintain an additional 1% failure [6], these results show that physical gate noise causes significantly more failure almost immediately past that threshold.

Differences between adversary configurations are also notable. The no faulty and R_0 faulty cases degrade quickly, while the S faulty case remains robust up to 10^{-2}. This suggests the protocol is more sensitive to receiver-side faults or noise than to sender dishonesty, exposing a structural vulnerability. The behaviour exhibited by the S faulty case might be caused by the higher number of valid outcome combinations compared to the other cases in the theoretical truth table provided by Guba et al. [6], which helps explain its greater tolerance to noise observed in simulation.

These findings present a clear idea. Gate-level noise significantly increases failure probability, and the protocol becomes unreliable at noise rates commonly observed on current devices, such as IBM hardware [16]. Thus, without error correction or mitigation that reduces effective noise below 10^{-4}, practical deployment seems infeasible. These findings deepen our understanding of the protocol's behaviour under realistic noise and help clarify where its theoretical guarantees begin to break down in practice.

8 Responsible Research

The purpose of this section is to reflect on and address the ethical aspects of the research, as well as to evaluate the reproducibility of the methods and results presented in this study.

All simulations were performed using known quantum computing simulation frameworks: SquidASM and NetSquid. The base circuit used in the WBC(3,1) protocol was reproduced based on the design and methodology proposed by Guba et al. [6]. All additional logic for simulation control and noise modelling is documented and was implemented using publicly available Python libraries, supporting the reproducibility of this study.

With regard to data integrity and transparency, no simulation results were selectively omitted or filtered. All outcomes were included in the analysis to preserve the credibility and neutrality of the experimental findings.

Ethical considerations were minimal, as the research did not involve human subjects or personal data. However, quantum-based consensus protocols may play a role in future applications involving secure communication, distributed systems, and blockchain technologies,

by extending existing implementations with improved performance [17], or combined with classical implementations to form complete solutions [18]. While the protocol examined in this study is designed to improve reliability and trust, similar mechanisms could also be misused, for example, to undermine fairness in distributed systems if malicious actors gain control over resources or exploit vulnerabilities in implementation.

9 Conclusions and Future Work

This research investigated the resilience of the WBC(3,1) Weak Broadcast Protocol under realistic noise conditions. The main goal was to assess whether this quantum protocol could remain viable on noisy quantum hardware. To achieve that, the Linear Circuit implementation of the protocol was reproduced, and its failure probability under gate-level noise models was evaluated.

The results show that even modest gate noise (around 10^{-3}) causes a sharp increase in failure probability, with the protocol becoming entirely unreliable at noise rates over 0.01. These findings hold across all configurations, although the S faulty adversary configuration showed greater robustness than the other two scenarios. Overall, the outcomes suggest that, in its current form, the protocol is not suitable for deployment on quantum hardware devices unless noise levels are significantly small.

Despite these conclusions, several questions remain. This study considered only one type of noise model and a fixed number of singlet states. Future work could explore the impact of combining noise types, as well as the role of measurement and communication errors. There is also potential in investigating if certain errors can be detected and mitigated through local checks or different thresholds, or whether the protocol itself can be modified to become more error resilient, as an alternative to improving the hardware to match the low noise requirements. To advance towards practical implementation, future work should explore these strategies, tailored to the protocol's structure.

A Appendix A - Linear Circuit Recreation

As mentioned in Section 4, before introducing noise into the protocol, the Linear Circuit implementation of the WBC(3,1) protocol was recreated under ideal, noise-free conditions to validate the correctness of the simulation environment. The goal was to replicate the baseline behaviour reported by Guba et al. in their study [6], specifically focusing on the relationship between the number of singlet states m and the protocol's failure probability across different adversarial configurations [6]. For each adversary configuration, the protocol was executed for increasing values of states count used per bit sent m, and the failure probability was collected from 500 runs per setting.

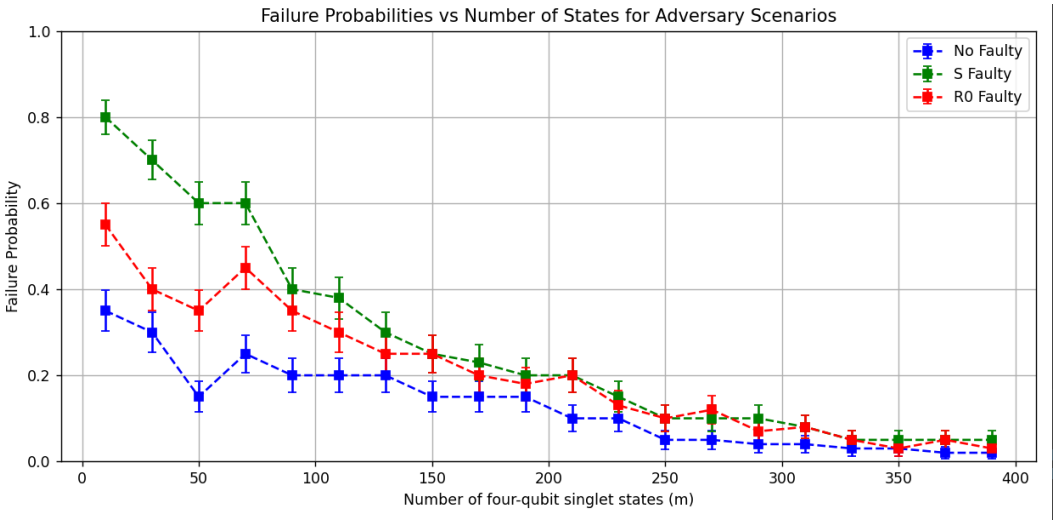

Figure 8: Failure probabilities for adversary configurations under noise-free conditions.

Figure 8 presents the baseline results for all three adversarial configurations. The results follow those published in the original study by presenting several similarities.

All three adversary configurations show a decreasing failure probability as m (number of singlet states) increases. In the no faulty case (blue), the failure probability starts the lowest and further decreases, confirming the protocol's theoretical guarantee of correctness under honest conditions. The S faulty configuration (green) follows a similar trend but starts with the highest failure rate, illustrating the impact of a malicious sender in disrupting the agreement. The R_0 faulty scenario (red) follows suit, but has a slower decay than the S faulty scenario. This behaviour mimics expectations from the original theoretical analysis.

Furthermore, for $m \geq 300$, the failure rate in the no faulty configuration flattens to a 5% failure probability, and the faulty configurations follow suit. Guba et al. [6] also report a $\sim 5\%$ failure rate in the no faulty case for $m = 280$, using parameters $\mu = 0.272$ and $\lambda = 0.94$, which are values used in the noisy simulations.

The observed behaviour aligns closely with the analytical and simulated outcomes presented in the original study, validating the reproduction of the protocol logic.

References

[1] M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of distributed consensus with one faulty process", *J. ACM*, vol. 32, no. 2, 374â382, Apr. 1985, ISSN: 0004-5411. DOI: 10.1145/3149.214121. [Online]. Available: https://doi.org/10.1145/3149.214121.

[2] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem", *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, Jul. 1982, ISSN: 0164-0925. DOI: 10.1145/357172.357176. [Online]. Available: https://doi.org/10.1145/357172.357176.

[3] T. Anema and J. Decouchant, *Message efficient byzantine reliable broadcast protocols on known topologies*, Bachelor Seminar of Computer Science and Engineering, 2024.

[4] D. Dolev and H. R. Strong, "Authenticated algorithms for byzantine agreement", 4, vol. 12, Society for Industrial and Applied Mathematics, 1983, pp. 656–666. DOI: 10.1137/0212045.

[5] M. Fitzi, N. Gisin, and U. Maurer, "Quantum solution to the byzantine agreement problem", *Phys. Rev. Lett.*, vol. 87, p. 217 901, 21 Nov. 2001. DOI: 10.1103/PhysRevLett.87.217901. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.87.217901.

[6] Z. Guba, I. Finta, A. Budai, L. Farkas, Z. Zimboras, and A. Palyi, "Resource analysis for quantum-aided byzantine agreement with the four-qubit singlet state", *Quantum*, 2024, Accepted 2024-03-21, Published under CC-BY 4.0. DOI: 10.22331/q-2024-04-30-1324. [Online]. Available: https://quantum-journal.org/papers/q-2024-04-30-1324/.

[7] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[8] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, "Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels", *Physical Review Letters*, vol. 70, no. 13, pp. 1895–1899, 1993. DOI: 10.1103/PhysRevLett.70.1895. [Online]. Available: https://doi.org/10.1103/PhysRevLett.70.1895.

[9] L. Lamport, "The weak byzantine generals problem", *J. ACM*, vol. 30, no. 3, pp. 668–676, Jul. 1983, ISSN: 0004-5411. DOI: 10.1145/2402.322398. [Online]. Available: https://doi.org/10.1145/2402.322398.

[10] M. M. Wilde, *Quantum Information Theory*. Cambridge University Press, 2007, pp. 1–655. DOI: 10.1017/CBO9781139525343. [Online]. Available: https://repository.lsu.edu/physics_astronomy_pubs/51.

[11] D. Gottesman, *An introduction to quantum error correction and fault-tolerant quantum computation*, 2009. arXiv: 0904.2557 [quant-ph]. [Online]. Available: https://arxiv.org/abs/0904.2557.

[12] M. Urbanek, B. Nachman, V. R. Pascuzzi, A. He, C. W. Bauer, and W. A. de Jong, "Mitigating depolarizing noise on quantum computers with noise-estimation circuits", *Physical Review Letters*, vol. 127, no. 27, Dec. 2021, ISSN: 1079-7114. DOI: 10.1103/physrevlett.127.270502. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.127.270502.

[13] M. F. Mor-Ruiz and W. Dür, "Influence of noise in entanglement-based quantum networks", *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 7, pp. 1793–1807, 2024. DOI: 10.1109/JSAC.2024.3380089.

[14] QuTech. "Netsquid - network simulator for quantum information using discrete events". Accessed: 2025-04-23. (2020), [Online]. Available: https://netsquid.org/.

[15] QuTech. "Squidasm: A modular quantum network simulator interface". Accessed: 2025-04-23. (2023), [Online]. Available: https://github.com/QuTech-Delft/squidasm.

[16] IBM Quantum, *Ibm quantum services resources*, Accessed: 2025-06-22, 2025. [Online]. Available: https://quantum.ibm.com/services/resources.

[17] W. Cui, T. Dou, and S. Yan, "Threats and opportunities: Blockchain meets quantum computation", in *2020 39th Chinese Control Conference (CCC)*, IEEE, Jul. 2020, 5822â5824. DOI: 10.23919/ccc50068.2020.9189608. [Online]. Available: http://dx.doi.org/10.23919/CCC50068.2020.9189608.

[18] J. Lin, H. Li, H. Xing, *et al.*, *Q-pnv: A quantum consensus mechanism for security consortium blockchains*, arXiv:2412.06325v1 [cs.CR], Version 1, submitted 9 Dec 2024, Dec. 2024. [Online]. Available: https://arxiv.org/abs/2412.06325v1.