Quantum Byzantine Agreement Protocol Under Noisy Conditions

Evaluating the Impact of Qubit Decoherence on the Protocol's Success Rate

Prisha Meswani[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: Prisha Meswani
Final project course: CSE3000 Research Project
Thesis committee: Tim Coopmans, Arie van Deursen

Abstract

Byzantine agreement protocols allow distributed systems to achieve consensus despite faulty nodes. The classical solution to the Byzantine agreement can only tolerate less than 1/3 of the total nodes being faulty, while a quantum-aided protocol has the potential to tolerate up to 1/2 of the nodes being faulty. However, the quantum states used in the protocol are vulnerable to decoherence, a process that results in the degradation of a quantum state. This research investigates how memory decoherence affects the failure rate of a three-party quantum Byzantine agreement protocol. The primary contribution is demonstrating that the protocol is resilient to carbon T_2 decoherence, as its Z-basis measurement scheme is insensitive to phase errors. Conversely, the failure probability increases with decreasing carbon T_1 decoherence values. Extrapolating from the observed trend of decreasing failure with longer T_1 coherence times, the protocol's performance on nitrogen vacancy center hardware is expected to approach the ideal, noiseless limit due to their experimentally established long T_1 times.

1 Introduction

The allegory introducing the Byzantine Generals Problem [10] depicts several Byzantine army generals, each commanding their own division, who must collectively decide whether to attack or retreat. The challenge, however, is that a fraction of these generals are traitors, deliberately relaying incorrect messages to sabotage the loyal ones. How do the loyal generals decide on a unified plan, even in the presence of meddling traitors? They must use a strategy that allows them to identify and overcome false messages. However, instead of a battalion of generals, there is a network of nodes, a bit-sequence to be agreed upon in place of a battle plan, and faulty or non-faulty nodes instead of traitorous or loyal generals.

In computer science, Byzantine agreement is the process of multiple nodes in a network coming to a consensus on a common value, despite the presence of faulty nodes. A faulty node, like the disloyal general, behaves inconsistently, sending incorrect data to other participating nodes and interfering with the consensus process. Said nodes can be acting with malicious intent or can just be malfunctioning. This problem is relevant in fault-tolerant distributed systems communications [8]. For instance, in blockchain technologies, nodes must agree on the order and validity of transactions to maintain a consistent ledger, while avoiding fraudulent entries. One example of a ledger that uses a consensus protocol to manage Byzantine faults is the XRP Ledger (associated with Ripple) [18].

Achieving a Byzantine agreement has constraints on the faulty parties that can be tolerated. In a system of n nodes, the classical (i.e., non-quantum, based purely on digital communication/computation) agreement protocol can only tolerate less than a third of the nodes being faulty [10]. Any more, and consensus cannot be reliably reached. Whereas, the quantum Byzantine agreement protocol can tolerate less than half of the total nodes being faulty [7]. For example, classical consensus requires at least three non-faulty nodes to tolerate one faulty node, whereas the quantum protocol needs only two. As the network scales, this advantage becomes more pronounced, highlighting its improved performance with increasing network size [6]. What makes the protocol a quantum one is the usage of multiple entangled multi-qubit states [2].

Previous work in [9] studied optimal protocol parameters and how measurement and state quality degrade with quantum noise. While it confirms noisy hardware affects the quantum states, the contributions of different noise sources and their quantitative impact on failure rates in a quantum network environment remain unexplored. One such source of noise, and the focus of this report, is decoherence. Qubits cannot maintain their quantum

state indefinitely, gradually drifting away from their originally prepared states. This is known as decoherence. In this paper, the Byzantine agreement protocol's failure probability is simulated. This paper builds on the work of [9] by investigating an aspect not addressed in their simulation: the impact of memory decoherence on the functioning of the protocol. The research question this paper aims to answer is: How is the success probability of the quantum Byzantine agreement protocol affected by memory decoherence in a three-party network? Three nodes are used as they are the minimum amount required to demonstrate the quantum protocol with faulty nodes, which is impossible with the classical one [6]. The problem is not applicable to smaller networks [6].

This paper is structured as follows: Chapters 2 and 3 provide the necessary background and formalize the research problem. The simulation methodology and experimental details are in Chapters 4-5. The results are presented in Chapter 6, and analyzed and discussed in Chapter 7, leading to the final conclusion in Chapter 9. Ethical considerations are in Chapter 8.

2 Research Background

This section outlines the foundational works for this research. Section 2.1 introduces key quantum information concepts. Section 2.2 details the quantum-aided protocol. Section 2.3 explains the experiment's noise source, and Section 2.4 describes the noise model used to assess the protocol under decoherence.

2.1 Fundamental Quantum Concepts

Whereas a bit must be either zero or one, a qubit, the fundamental unit of quantum information, can be a combination of both states simultaneously. This is known as a superposition. Quantum states are represented as a complex vector and are written using Dirac notation, where $|0\rangle$ and $|1\rangle$ represent the basis states of a qubit [13]. A general qubit state is written as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle)$, where α and β are the amplitudes of the qubit. They are complex numbers such that $|\alpha|^2+|\beta|^2 = 1$. The probability of measuring zero or one is determined by the squared magnitude of its amplitude [13]. In the example in Equation 1, the probability of measuring zero is $|\frac{1}{\sqrt{2}}|^2 = \frac{1}{2}$.

Measuring a qubit removes it from superposition and into a single classical state; from that point onward, measuring the qubit will always give the same outcome. Like bits may be manipulated with logic gates, qubits can be controlled by quantum gates, which are represented by unitary matrices [13]. An example of a quantum gate is a Pauli-Z (Z) gate. The Z gate applies a phase flip to the $|1\rangle$ state, but leaves $|0\rangle$ unchanged. The effect of the Z gate on the quantum state in Equation 1 is shown in Equation 2.

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{1}$$

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, Z|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{2}$$

Furthermore, qubits can be entangled, a phenomenon where the measurement of one qubit instantaneously influences the outcome of another, no matter how far apart they are [21]. An analogy that describes entanglement is that of a "magic" coin. Assume you

and your friend each have a coin and go to a location far and unknown to the other. If you flip your coin and see heads, you know with 100% certainty that your friend's coin will also fall on heads when they flip it. The magic coin represents two entangled qubits. Flipping a coin to see its outcome corresponds to measuring a qubit, and the result, heads or tails, is equivalent to measuring a zero or one.

Physically, in the scope of this research, qubits can be represented either using the spin of an electron or the spin of a nearby carbon-13 nucleus. Electrons are light and thus easier to control, making it convenient to perform operations, such as quantum gates, on them. Carbon qubits are stable and less likely to be affected by external forces, so they are preferred for storing quantum information for longer periods. Readers seeking a deeper understanding of the concepts in this section may consult [13].

2.2 Weak Broadcast with Entangled States

A Byzantine agreement includes a network of nodes where a single node aims to broadcast a message (or bit, in the case of this research) to all other nodes [7]. All nodes must decide on an output value, x_i, and the protocol is deemed a success if:

- Given a non-faulty sender that broadcasts the bit x_s, non-faulty receivers output x_s.
- Given a faulty sender, the non-faulty receivers all output the same value or $\perp$.

$\perp$ signals an abort, meaning the node does not decide on any output value. This process is also referred to as a Weak Broadcast. This paper will focus on a Weak Broadcast among three parties (one sender and two receivers). What makes this protocol a quantum one is the usage of multiple copies of an entangled four-qubit state [2]:

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

The state aids in verifying honesty by distributing the qubits among the three nodes. The measurement outcomes are used to cross-check if a node has received the right value. Multiple copies of the state are used because the failure probability of the protocol converges to zero as the number of states used approaches infinity [9]. The usage of the quantum states enables the protocol to tolerate less than half of the nodes being faulty [7], which is an enhancement compared to the classical version, which can only tolerate fewer than a third of the total nodes being faulty [10].

2.3 Quantum Teleportation and Decoherence

For each node to make use of the quantum state, the qubits in the quantum state need to be distributed between the nodes in the network. The qubits are distributed using a process called quantum teleportation: the transfer of a qubit from one place to another without changing its quantum state, even when neither the sender nor receiver has information on how the quantum state was prepared [1]. For the sake of an example that is similar to the process in [1], let us assume Alice wants to send the qubit $|\phi\rangle$, which is in an arbitrary and unknown state, to Bob. An Einstein-Podolsky-Rosen (EPR) pair [13], $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, is created. One qubit of $|\Phi^+\rangle$ is given to Alice and the other to Bob. Alice measures $|\phi\rangle$ and her EPR qubit in the Bell basis and stores the outcomes in the classical bits m_0 and m_1 respectively. She sends m_0 and m_1 to Bob (also classically). Bob applies a Pauli-Z gate to his EPR qubit if m_0 is 1 and a Pauli-X gate to his EPR qubit if m_1 is 1. The Pauli-Z

gate is described in Section 2.1 and the Pauli-X gate swaps the $|0\rangle$ state with $|1\rangle$ and vice versa. His EPR qubit is now transformed into $|\phi\rangle$. Since Alice measured $|\phi\rangle$, it is no longer in the state $|\phi\rangle$; thus, the "teleportation" is complete.

We have established that the Byzantine agreement protocol relies on quantum teleportation, which itself depends on generating an EPR pair between two nodes. This raises the question of how the EPR pair is created between each node. One of the ways entanglement is distributed across long distances is entanglement swapping [20]. This process is not always successful upon its first attempt due to imperfections encountered when physically controlling quantum information over a distance [20]. Realistically, multiple attempts to establish an EPR pair are made before one is actually created, introducing latency to the quantum teleportation process. Furthermore, quantum memories also have their imperfections; one of them being that they cannot preserve quantum states indefinitely [13]. The qubits held in memory can start to get corrupted, and the repeated attempts required to establish an EPR pair prolong this storage duration. By corrupted, it means they gradually deviate from the original state they were initialized to. This process is known as decoherence [13].

The research by [9] calculates the optimal values for the protocol's parameters μ and λ, and finds the experimental failure rates of the protocol given a varying number of quantum states used. They also evaluate the fidelity of both the measurements and the quantum states; that is, how closely the experimentally prepared state and the resulting measurements match the ideal quantum state mentioned above and its expected measurement outcomes when implemented on quantum computing hardware. This demonstrates that the quantum state, and subsequently the protocol, is impacted by noisy hardware. Some aspects that still need to be investigated are: (i) Noise arising from quantum hardware is not a single, uniform phenomenon; there are specific sources of noise, and how these individually impact the quality of the quantum state. (ii) How noise quantitatively impacts the protocol's failure rate in a quantum network environment.

2.4 Decoherence Noise Models

Since we want the quantum state to be intact for the duration of the protocol, carbon qubits are the better-suited choice due to their longer stability. Unfortunately, carbon qubits also decohere, albeit slower than their electron counterparts, so the degradation of a qubit is still unavoidable. Therefore, investigating the impact of decoherence aids in understanding the protocol's practical performance. There are two types of decoherence a carbon qubit is subject to: T_1 and T_2 decoherence. The former impacts a qubit's amplitudes, and the latter its phase. This research aims to investigate the impact of both decoherence types. The probability a single qubit is subject to both types of decoherence is defined by Equation 3 [3].

The model used to demonstrate the effects of T_2 decoherence is a dephasing channel. A dephasing channel probabilistically introduces a phase flip, by applying a Pauli-Z gate, to a qubit. This gate flips the phase of the $|1\rangle$ component of a qubit state, but leaves the probabilities of measuring $|0\rangle$ and $|1\rangle$ unchanged. As T_1 approaches infinity, the term $e^{\Delta t/2T_1}$ will approach one. Thus, the probability, p^{T_2}, of a qubit undergoing T_2 decoherence is defined in Equation 4. The model used to demonstrate the effects of T_1 noise is amplitude damping [3]. Amplitude damping gradually decreases the chances of measuring $|1\rangle$ and increases the probability of measuring $|0\rangle$. Equation 5 represents the formula for the probability, p^{T_1}, of a qubit undergoing T_1 decoherence [3].

$$p^{T_1T_2} = \frac{1}{2}\left(1 - e^{-\Delta t/T_2} \cdot e^{\Delta t/2T_1}\right) \tag{3}$$

$$p^{T_2} = \frac{1}{2}\left(1 - e^{-\Delta t/T_2}\right) \tag{4}$$

$$p^{T_1} = 1 - e^{-\Delta t/T_1} \tag{5}$$

In equations 3, 4, and 5, T_1 and T_2 represent the respective decoherence time values and Δt is the "idle time" of a qubit: while a qubit isn't actively participating in operations (like gates, teleportation, or measurement), it is vulnerable to decoherence noise. Since measurement collapses a quantum state to a classical bit, decoherence also does not impact measured qubits. Therefore, in this report, 'idle time' refers to the duration during which a qubit is exposed to decoherence and is not acted upon by any operations prior to its first measurement. Additionally, the decoherence times are not entirely independent. The T_1 time sets an upper bound on the T_2, resulting in the constraint $T_2 \leq 2T_1$ [4]. Studying the effect of decoherence helps evaluate the robustness of the agreement protocol under realistic network conditions where it is unavoidable.

3 Problem Statement

This section describes in depth the problem the research question aims to solve, framed as follows: What is the failure probability of the Byzantine agreement protocol under noiseless conditions, and how is this failure probability affected by noise? More specifically, the noise source this research considers is the effect of T_1 and T_2 carbon decoherence. For both cases, the failure probabilities will also take into account the presence of faulty nodes in the network. Faulty nodes refer to nodes in the network that behave incorrectly, either due to crashes, communication errors, or malicious intent, and may send incorrect or inconsistent messages that disrupt the protocol.

4 Methodology

In this section, the formalization of the research methodology will be outlined. Section 4.1 describes the protocol's step-by-step procedure, and section 4.2 includes how faulty behavior is incorporated into the simulation. Section 4.3 describes the noisy scenarios used.

4.1 Protocol Overview

The protocol uses three parameters: $0 < \mu < 1/3$, $1/2 < \lambda < 1$, and m, which is the number of shared quantum states. The step-by-step procedure of the protocol is as follows:

1. **Invocation Phase:** In three-node network, the sender, S, creates the entangled state $|\psi\rangle$ with a quantum circuit (detailed in Section 5). The state is distributed via quantum teleportation such that S keeps the first two qubits, and each receiver, R_0 and R_1, gets one of the remaining two. Multiple shared states are needed to ensure statistical reliability. Thus, quantum states are created and teleported m times, resulting in S holding $2m$ qubits, and R_0 and R_1 holding m qubits each. All nodes measure their respective qubits. S sends a bit x to R_0 and R_1. A check set is sent to each receiver, containing the indices of the quantum states where both of S's measurements equal x. The check set aids the receivers in independently confirming that the correct value has been sent by S.

2. **Check Phase:** The protocol proceeds with classical checks and communication. Each receiver verifies:
   (a) The check set from S contains at least $T = \lceil m \cdot \mu \rceil$ indices.
   (b) The measurement outcomes for all of their qubits in the check set are opposite to x.

   If both hold, the receivers accept x (setting $x_0 = x$ and $x_1 = x$); otherwise, they abort by setting $x_i = \perp$. The threshold T prevents a faulty sender from passing check (b) with a small check set, increasing its reliability. μ has an upper bound of $1/3$ as the probability of obtaining either the measurement outcome $|0011\rangle$ or $|1100\rangle$ is $1/3$ each. As μ gets closer to $1/3$, the check in (a) gets stricter. μ cannot be zero, as that would allow an empty check set, defeating the threshold's purpose. The opposite outcome is expected due to the entanglement of $|\psi\rangle$. If this does not hold, it indicates to the receivers that x is erroneous, leading to an abort.
3. **Cross-calling Phase:** R_0 sends its output (x_0) and check set (ρ_0) to R_1 so R_1 can verify it has agreed on the right value.
4. **Cross-check Phase:** R_1 updates its value to $x_1 = x_0$ if all of the following are true:
   (a) $x_0 \neq x_1$, $x_0 \neq \perp$, and $x_1 \neq \perp$
   (b) ρ_0 contains at least $T = \lceil m \cdot \mu \rceil$ indices.
   (c) For the qubits at the indices in ρ_0, R_1 measures the opposite value to x_0 at least $\lambda \cdot T + |\rho_0| - T$ times.

   By verifying the opposite value is measured for a large fraction of the indices, R_1 confirms R_0 has decided on x_0 correctly. Said fraction of indices is decided by λ. λ has an upper bound of one, exclusive, to account for measurement errors. However, go too low and R_1 may end up accepting an incorrect value, thus a lower bound of $1/2$ to ensure R_1 measures the appropriate value for a majority of the qubits. R_1 adopts R_0's output value if the checks pass and assumes the value S sent is incorrect, since by the definition of a Weak Broadcast, all non-faulty nodes should agree on the same value. Otherwise, it retains its previous value, assuming R_0 is the faulty node. Without this, R_1 would blindly accept its initial output from the check phase.

The protocol's outcome is a failure if it does not match the conditions for a Weak Broadcast mentioned in Section 2. Decoherence exclusively occurs before the generation of the check sets by the sender in the Invocation step. The idle time during which qubits can decohere is the period between their initial preparation and first measurement in Step 2. This excludes time spent on teleportation. Qubits do not undergo decoherence while a gate is acting on them. A detailed timeline of each qubit's decoherence events is shown in Figure 1. For this simulation, all classical communication channels are assumed to be error-free.

4.2 Simulated Faulty Behavior

The simulation includes three fault cases: one with no faulty nodes, one where S is faulty and another where R_0 is faulty. The faulty strategies used align with those in [9], with at most one faulty node per scenario, since the quantum Byzantine agreement requires less than half of the nodes to be faulty [7].

1. S **Faulty:** In the invocation phase, S sends x to R_0 and $1 - x$ to R_1, manipulating check sets such that R_1 receives indices where S measures two $1 - x$ outcomes, while R_0 receives a mix of indices from the remaining outcomes. It aims to pass checks in

Step 2 but fail Step 4(c), causing R_0 to output x and R_1 to output $1-x$, thus violating Weak Broadcast.

2. R_0 **Faulty:** S behaves honestly and sends x along with valid check sets. During cross-calling, R_0 sends $1-x$ and a manipulated check set, which contains the indices where R_0 expects R_1 to measure x enough times to pass the check. As a result, R_1 accepts the incorrect value, while R_0 outputs the original, violating Weak Broadcast.

A third possible outcome in these faulty scenarios is a *domain violation*, treated as an immediate protocol failure to ensure a secure upper bound on the failure probability [9]. This occurs when a faulty node lacks sufficient measurement outcomes to construct convincing check sets. For S faulty, a domain violation occurs when sets l_1, l_2, or l_3 from Algorithm 1 are too small, as constrained by Equation 6 [9]. For R_0 faulty, a violation occurs if set l_4 from Algorithm 2 is too small, as defined in Equation 7 [9]. In both equations 6 and 7, $Q = T - \lceil T\lambda \rceil + 1$ and $T = \lceil m \cdot \mu \rceil$.

$$T - Q \leq |l_1| \quad Q \leq |l_2| \quad T \leq |l_3| \tag{6}$$

$$|l_4| \leq m - T \tag{7}$$

The case of R_1 being faulty is not considered, as this node does not send information to any of the others. If R_1 were the (only) faulty node, the other two nodes would reach an agreement, guaranteeing success [9].

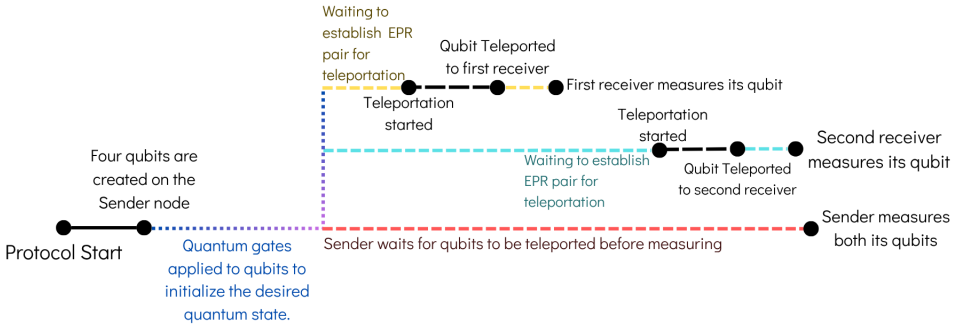


Figure 1: Decoherence timeline of a quantum state. Purple dots mark decoherence of all qubits except those in gate operations. Yellow, light blue, and red dashes indicate decoherence of the first receiver's, second receiver's, and sender's qubits, respectively. Black dashes mark intervals where only other qubits are decohering, and black lines indicate no decoherence. Qubits cease to decohere after their first measurement.

4.3 Simulated Decoherence Scenarios

First, the simulations will be run in a noiseless environment to establish a baseline for comparison. Then, the noise source from section 2.4 will be introduced, and the failure probabilities will be compared. An example run of the protocol under noise can be found in Appendix C. This research considers two noisy scenarios:

1. **Only T_2 Decoherence:** The effect of varying T_2 times is investigated first, while assuming T_1 is infinite and not in effect.

Algorithm 1 How the sender (S) creates the check sets

Require: x (Value to agree on)
  let $l_1 \leftarrow [\,]$, $l_2 \leftarrow [\,]$, and $l_3 \leftarrow [\,]$
  for $i \leftarrow 1$ to m **do**
    q_0, q_1, q_2, $q_3 \leftarrow$ PrepareQubits
    Teleport(q_2 to R_0, q_3 to R_1)
    b_0, $b_1 \leftarrow$ Measure(q_0, q_1)
    if $b_0 = x$ **and** $b_1 = x$ **then**
      append i to l_1
    else if $b_0 \neq x$ **and** $b_1 \neq x$ **then**
      append i to l_3
    else
      append i to l_2
    end if
  end for
  if S is faulty **then**
    let $check\_set_0 \leftarrow$ random sample of size Q from l_2
    append random sample of size $T - Q$ from l_1 to $check\_set_0$
    Send $check\_set_0$ to R_0 and l_3 as $check\_set_1$ to R_1
  else
    Send the check set l_1 to R_0 and R_1
  end if

Algorithm 2 How the faulty receiver R_0 manipulates check sets

Require: x_0 (Value agreed on by R_0), $check\_set_0$ (Check set sent to R_0 by S), $qubits$ (List of qubits of length m)
  let $l_4 \leftarrow [\,]$, $l_5 \leftarrow [\,]$, $l_6 \leftarrow [\,]$
  for $i \leftarrow 1$ to m **do**
    $q \leftarrow qubits[i]$
    if $i \notin check\_set_0$ **and** $q \neq x_0$ **then**
      append i to l_5
    else if $q = x_0$ **then**
      append i to l_6
    else
      append i to l_4
    end if
  end for
  $x_0 \leftarrow 1 - x_0$
  $check\_set_{01} \leftarrow l_5$
  if $|l_5| < T$ **then**
    $n_{min} \leftarrow T - |l_5|$
  else
    $n_{min} \leftarrow 0$
  end if
  append random sample of size n_{min} from l_6 to $check\_set_{01}$
  Send x_0 and $check\_set_{01}$ to R_1

2. **Combined T_1 and T_2 Decoherence:** A "T_1-only" scenario, where T_2 is infinite and ineffective, is not achievable, as the constraint $T_2 \leq 2T_1$ requires that T_2 is finite. Therefore, the combined effect, of varying both T_1 and T_2 is analyzed.

5 Experimental Setup

This section outlines the simulation environment used to compute the failure probability of the agreement protocol in both noiseless and noisy scenarios. Section 5.1 describes the general and noiseless setup, while Section 5.2 details the modifications made to introduce noise.

5.1 General Simulation Setup

Simulations were performed in Python using SquidASM [14], which supports quantum networks and noise modeling. The values chosen for the parameters μ and λ were 0.272 and 0.94 respectively. These were the optimal values calculated by [9]. For m, values from 20 to 400 in increments of 10 were used, with each m value simulated separately.

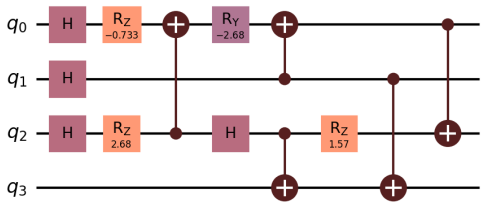


Figure 2: Loop circuit preparing the quantum state: q_0 and q_1 go to S, q_2 and q_3 to R_0 and R_1 each [9].

The network configuration of three nodes, for S, R_0, and R_1, was used. For the noiseless simulation, a generic network model was used, meaning the communication between nodes is instant and error-free. The quantum states were prepared on S's node using the loop circuit in Figure 2. The steps in Section 4.1 are implemented by the nodes, and each of them returns either a one or zero, an abort, or a domain violation. The return values are used to evaluate the success of the protocol, following the truth table in Table 1. The success or failure is indifferent to the input bit x, so the noiseless experiment was conducted with only $x = 0$.

S	**R_0**	**R_1**	**No faulty**	**S faulty**	**R_0 faulty**
x	x	x	✓	✓	✓
x	x	$1-x$	×	×	×
x	x	$\perp$	×	✓	×
x	$1-x$	x	×	×	✓
x	$1-x$	$1-x$	×	✓	×
x	$1-x$	$\perp$	×	✓	×
x	$\perp$	x	×	✓	✓
x	$\perp$	$1-x$	×	✓	×
x	$\perp$	$\perp$	×	✓	×
ϵ	$\perp$ or x or $1-x$	$\perp$ or x or $1-x$	N/a	×	N/a
$\perp$ or x or $1-x$	ϵ	$\perp$ or x or $1-x$	N/a	N/a	×

Table 1: Protocol outcomes for various fault scenarios with $x \in 0, 1$ [9]. ✓: success, ×: failure, $\perp$: abort, ϵ: domain violation, N/a: not applicable.

Monte Carlo simulations with 10,000 samples were used to calculate the failure probabilities corresponding to the cases of no faults, S being faulty, and R_0 being faulty. [5] aided in running the large amount of simulations needed for generating the aforementioned number of samples. The exact values for each m, calculated with the formulas in Appendix B [9], were also plotted.

5.2 Noisy Experiment Setup

To simulate noise, changes were made to the network. First, a nitrogen vacancy (NV) device [16] was used to model the nodes. The device models nitrogen-vacancy centers in diamonds, including their electron and nuclear spin qubits and noise parameters, which are

used in this context to vary the carbon qubit's T_1 and T_2 decoherence times. The value of m was fixed at 280. This value is chosen as it is a balance between maintaining a low failure rate and not using too many shared quantum states (resource efficiency). Ideally, the number of states for a zero percent failure rate would be used, but as proven by [9], that would take an infinite amount of states to achieve.

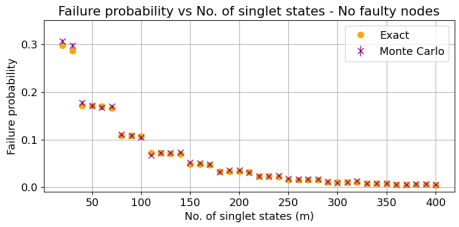
To evaluate the impact of T_2 on the failure rate, carbon T_2 times were varied from 10^{-6} to 10 seconds. In SquidASM, noise parameters must be integers, making it impossible to set T_1 to infinity. Instead, T_1 was set to zero, which effectively disables dephasing based on NV device implementation behavior [3]. For the second experiment, assessing the joint impact of T_1 and T_2, both values were varied across the same range, 10^{-6} to 10 seconds. For both noisy simulations, decoherence values were incremented following a cubic growth pattern. This non-linear scaling captures sensitivity to short decoherence times while showing trends at higher values. The lower bound of 10^{-6} simulates near-instantaneous decoherence, and 10 seconds was chosen as an upper bound to represent a tenfold improvement [3].

The quantum links between the sender and each receiver were modeled with probabilistic entanglement generation using depolarizing links, which allow specification of fidelity (how close the generated state is to a perfect EPR pair), per generation attempt duration, and success probability [15]. The success probability of generating an EPR pair was set to 0.001 [3], per attempt duration to 10^-6 seconds, and fidelity 1 to ensure the EPR pairs generated are ideal. This means entanglement generation, on average, takes $10^3 \times 10^-6 = 0.001$ seconds. Again, the failure probability was calculated using the Monte Carlo method, with 1,200 samples used for the T_1 and T_2 experiment and 500 for T_2. To ensure balanced input evaluation, half of the samples used $x = 0$ and the other half $x = 1$.

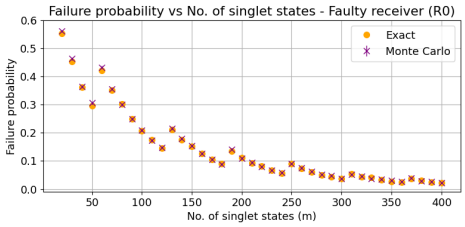
6 Results

This section presents the results of all the simulations. Figure 3 shows the noiseless simulation results, plotting failure probability against the number of shared states (m) for all fault configurations. The orange circles are the exact failure probabilities for each m value, and purple crosses the Monte Carlo simulation outcomes. Error bars, calculated as the standard error of a Bernoulli trial, are plotted on the Monte Carlo points; however, due to the large sample size, they are not distinguishable. Most of the Monte Carlo points overlap with the exact values for the upper bounds of the failure rate. This is by design, as the simulations model the worst-case scenario by assuming a protocol failure occurs whenever the faulty node's optimal strategy cannot be executed due to a domain violation.

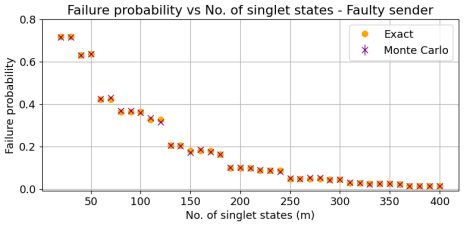
Figure 5 and 4 depict the failure probability results against varying noise levels for T_2 decoherence and the combination of T_1 and T_2 respectively. The red dashed and blue lines represent the exact and Monte Carlo noiseless failure probabilities (from Figure 3) respectively for $m = 280$. It can be observed from all the sub-figures in Figure 5 that the failure probability is not significantly affected by the decoherence values and all points remain close to the exact value/noiseless line. In Figure 4, it can be observed that the failure probabilities follow a curve, decreasing as T_1 and T_2 increase.



(a) No faulty nodes

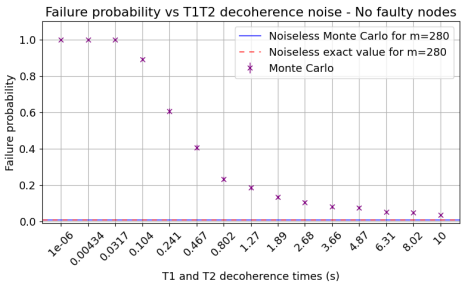


(b) R_0 faulty

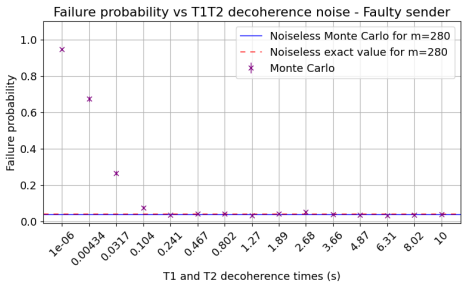


(c) S faulty

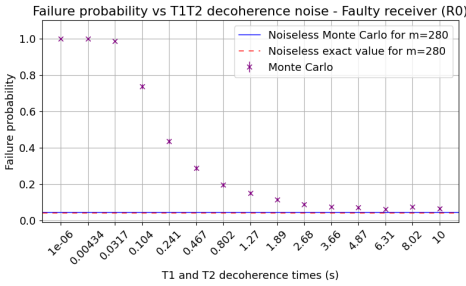
Figure 3: Failure probability vs No. of shared states, for $m = 20$ to 400



(a) No faulty nodes

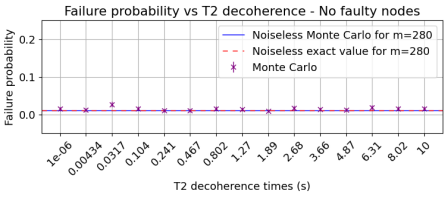


(b) Sender faulty

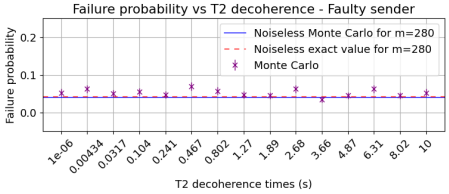


(c) Receiver 0 faulty

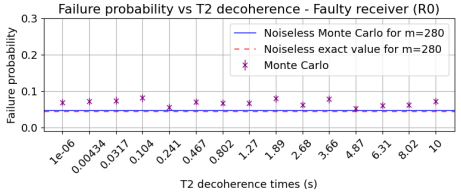
Figure 4: Failure probability vs T_1 and T_2 Carbon Decoherence Time with $p = 0.001$ and $m = 280$



(a) No faulty nodes



(b) S faulty



(c) R_0 faulty

Figure 5: Failure probability vs T_2 Carbon Decoherence Time with $p = 0.001$ and $m = 280$

7 Discussion

A key finding is that carbon T_2 decoherence has negligible impact on the protocol's failure probability, due to dephasing affecting only the phase, not the amplitude, of quantum states [13]. A maximally dephased state, where all phases are flipped, illustrates this:

$$|\sigma\rangle = \frac{1}{2\sqrt{3}}\big(-2\,|0011\rangle + |0101\rangle + |0110\rangle + |1010\rangle + |1001\rangle - 2\,|1100\rangle\big)$$

$$p_{|0011\rangle} = p_{|1100\rangle} = \left|\frac{-2}{2\sqrt{3}}\right|^2 = \frac{1}{3},\ p_{|1001\rangle} = p_{|0110\rangle} = p_{|1010\rangle} = p_{|0101\rangle} = \left|\frac{1}{2\sqrt{3}}\right|^2 = \frac{1}{12}$$

While the state is different, the measurement outcomes and their probabilities, $p_{|state\rangle}$, are not, as seen in the equation above. This is because of the sole reliance on Z-basis measurements, which distinguish between $|0\rangle$ and $|1\rangle$ only. This outcome holds regardless of the EPR generation probability, as results are not influenced by dephasing to begin with.

In contrast, the results for T_1 and T_2 decoherence combined suggest that shorter times lead to poorer protocol performance. Earlier, we proved that the effect of T_2 is negligible, thus the conclusion can be drawn that the change in performance is attributed to the effect of T_1 noise. Also, following Figure 4's downward curve, it can also be deduced that, with increasing values of T_1, the failure probability will approach its noiseless value. This is supported by the probability that none of the four qubits in the quantum state decohere for $T_1 = 10$ hours and $T_2 = 1$ second (realistic T_1 and T_2 time estimations on NV hardware [3].) is $\left(1 - \left[\frac{1}{2}\left(1 - e^{-\frac{0.001}{1}} \cdot e^{\frac{0.001}{2 \cdot 36000}}\right)\right]\right)^4 \approx 0.998$. The equation raised to the fourth power is from Section 2.4.

In Figure 1, much of the decoherence experienced by the qubits of S and R_1 occurs while waiting for teleportation to complete sequentially. Ideally, teleportation would occur simultaneously to reduce idle time. However, this sequential teleportation is due to the

limitations of SquidASM's NV model [16]. The NV device can store one electron spin qubit and several carbon-13 spin qubits [16]. As described in Section 2.3, teleportation requires an EPR pair that must be created on the electron spin qubits of both the sender and receiver. Since each node has only one electron spin qubit, simultaneous teleportation is not possible. The practicality of this protocol is another point of consideration. When the protocol achieves Weak Broadcast by aborting in the case of faulty nodes, it prevents a fail but does not identify which node was the source of the error. This may limit its usefulness in real-world systems where identifying faulty nodes is necessary. Furthermore, the method for calculating the protocol's failure rate may be too pessimistic, due to the exclusion of domain violations. These instances of successful but passive defense could provide a more realistic measure of the protocol's resilience, as these outcomes could be observed in the protocol's practical application.

Future work could explore the protocol's vulnerability to decoherence in non-Z bases, which would reveal the impact of Pauli-Z-induced phase changes and allow analysis of T_2 decoherence effects. The protocol could also be modified to allow some tolerance in the Check Phase to account for T_1-induced bit flips. Instead of requiring all outcomes to oppose bit x, a parameter could define the minimum fraction of matching outcomes. Future work includes optimizing this new parameter to include noise acceptance without being so lenient that faulty senders can succeed.

8 Responsible Research

Part of this research involves the reproduction of the methodology and experiment presented by Guba et al. [9]. All sections of this research that are taken from [9] are clearly cited. While the methodology is built on [9], the simulation code was developed independently by the researcher. The results of the reproduction of the experiment have been honestly presented, even in the event of deviations from the original.

Secondly, to ensure that the original experiment conducted in this research is also reproducible, the code used was uploaded to GitHub with clear instructions on how to run it. The source code is publicly available and can be found here [12]. All software tools used, as well as the steps taken to conduct this experiment, are also mentioned in this paper. This aligns with the reproducibility recommendations given in [19]. This research does not involve the collection of human subject data, therefore it does not require ethical approval related to human participation.

Beyond the reproducibility of this research, the quantum Byzantine agreement protocol has the potential to strengthen fault-tolerant distributed communication against malicious behavior and transmission errors. The protocol may be applied in fields like blockchain technologies [18] and database management systems [8], which all involve the mitigation of Byzantine faults. From an ethical standpoint, the protocol may contribute to building systems that are more resilient to manipulation and corruption, which this research has a positive impact on as it evaluates one of the hurdles the protocol may face in its deployment.

Another point of consideration is that the concept of quantum networks is fairly new and not yet widespread in its use. Just like the propagation of the contemporary internet [17], if quantum networking technology were to be publicly available in the future, some regions and companies would have access to it earlier than others [11]. This creates an imbalance where only some parties have the ability to use and reap the benefits of the quantum Byzantine agreement while others remain vulnerable. This imbalance does not imply that research into quantum-aided protocols should be halted. Rather, it highlights the importance of ensuring

that access to quantum technologies becomes sufficiently widespread and equal before such protocols are implemented in critical infrastructure, preventing disparity in its usage.

9 Conclusion

This research set out to answer the question: How is the success probability of the quantum Byzantine agreement protocol affected by memory decoherence in a three-party network? To address this, the noiseless failure probability of the protocol, as a function of the number of shared states, m, including three scenarios: no faulty nodes, a faulty sender, and a faulty receiver, was simulated. The simulation results matched the results from [9], confirming the validity of our simulation framework. Then, the impact of carbon T_1 and T_2 decoherence noise levels were investigated by introducing two decoherence scenarios using an NV device model: one isolating the effects of T_2 decoherence and another combining T_1 and T_2 noise.

T_2 carbon decoherence has negligible impact on failure probability because dephasing errors (modeled by Pauli-Z) only affect phase, which the protocol disregards due to exclusive Z-basis measurements that decided only by amplitudes. This is observed regardless of how long the quantum state is exposed to T_2 decoherence, demonstrating resilience. In contrast, the second finding is that T_1 and T_2 decoherence combined impacts the protocol performance, with shorter coherence times leading to higher failure rates. Since the effect of T_2 was established to be negligible, this performance degradation can be attributed to T_1 noise. The protocol's performance on NV hardware should approach the noiseless failure rate due to their long T_1 times. In conclusion, while this study confirms the protocol's resilience to T_2 decoherence, it highlights its vulnerability to short T_1 decoherence times.

The protocol simulation's limitations include its inability to identify faulty nodes, an overly pessimistic failure calculation by treating all domain violations as failures, and decoherence caused by sequential teleportation delays, which arise from the NV device hardware restricting each node to a single electron spin qubit, preventing simultaneous teleportation. A next step could be to investigate the protocol's performance with measurements in different bases, which would make the protocol sensitive to T_2 decoherence. The protocol can also be adapted to tolerate some bit-flip errors during the Check Phase by setting a minimum fraction of matching outcomes, balancing noise robustness and security.

References

[1] 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. *Phys. Rev. Lett.*, 70:1895–1899, Mar 1993.

[2] Adán Cabello. Solving the liar detection problem using the four-qubit singlet state. *Phys. Rev. A*, 68:012304, Jul 2003.

[3] 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, David Elkouss, and Stephanie Wehner. Netsquid, a network simulator for quantum information using discrete events. *Communications Physics*, 4(1):164, 2021.

[4] T.J. Coopmans. *Tools for the design of quantum repeater networks.* Dissertation (tu delft), Delft University of Technology, 2021.

[5] Delft High Performance Computing Centre (DHPC). DelftBlue Supercomputer (Phase 2). https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2, 2024.

[6] Matthias Fitzi. Generalized communication and security models in byzantine agreement. 2002.

[7] Matthias Fitzi, Nicolas Gisin, Ueli Maurer, and Oliver von Rotz. Unconditional byzantine agreement and multi-party computation secure against dishonest minorities from scratch. In Lars R. Knudsen, editor, *Advances in Cryptology — EUROCRYPT 2002*, pages 482–501, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[8] Hector Garcia Molina, Frank Pittelli, and Susan Davidson. Applications of byzantine agreement in database systems. *ACM Trans. Database Syst.*, 11(1):27–47, March 1986.

[9] 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. *Quantum*, 8:1324, April 2024.

[10] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, July 1982.

[11] McKinsey & Company. Quantum technology monitor – april 2024. https://www.mckinsey.com/~/media/mckinsey/business%20functions/mckinsey%20digital/our%20insights/steady%20progress%20in%20approaching%20the%20quantum%20advantage/quantum-technology-monitor-april-2024.pdf, 2024. Accessed: 2025-06-20.

[12] Prisha Meswani. Noisy quantum byzantine agreement protocol. https://github.com/prisha-m/noisy-quantum-byzantine-agreement-protocol, 2025. GitHub repository.

[13] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 2010.

[14] QuTech. SquidASM: A simulator based on netsquid that can execute applications written using netqasm. https://github.com/QuTech-Delft/squidasm, 2021. Accessed: 2025-05-18.

[15] QuTech. Squidasm documentation: Depolarise link, 2023. Accessed: 2025-05-31.

[16] QuTech. Squidasm documentation: Nv qdevice configuration, 2023. Accessed: 2025-05-31.

[17] Hannah Ritchie, Edouard Mathieu, Max Roser, and Esteban Ortiz-Ospina. Data page: Share of the population using the internet. https://ourworldindata.org/grapher/share-of-individuals-using-the-internet, 2023. Part of the publication: *Internet*. Data adapted from International Telecommunication Union (ITU), via World Bank. Online resource.

[18] David Schwartz, Noah Youngs, and Arthur Britto. The ripple protocol consensus algorithm. Technical report, Ripple Labs Inc., 2014.

[19] V.C. Stodden. Reproducible research: Addressing the need for data and code sharing in computational science. *Computing in Science and Engineering*, 12:8–13, 01 2010.

[20] Stephanie Wehner, David Elkouss, and Ronald Hanson. Quantum internet: A vision for the road ahead. *Science*, 362(6412):eaam9288, 2018.

[21] Mark M. Wilde. *Quantum Information Theory*. Cambridge University Press, 2007.

A Figure 4 of [9]

This appendix presents the collection of graphs depicting the failure probabilities of the Byzantine agreement protocol from [9]. They served as a reference point for comparing the protocol simulations developed in this research.

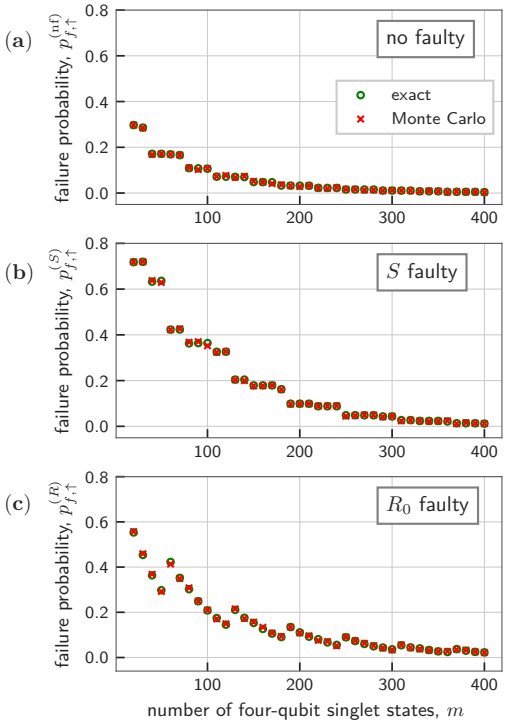


Figure 6: Failure probability of the Weak Broadcast protocol under three adversary configurations. (a) No faulty nodes (b) S faulty (c) R_0 faulty. Exact results (green circles) and Monte Carlo estimates (red crosses) for failure probability. Parameters used: $\mu = 0.272$, $\lambda = 0.94$. Simulation results use 10,000 random events for each m.

B Exact Value Formulas

This appendix includes the formulas used to estimate the upper bounds of the failure probabilities of the protocol given varying values of m. These formulas were taken from [9]. They assume x, the value to be agreed upon, is zero, $T = \lceil m \cdot \mu \rceil$, and $Q = T - \lceil T \cdot \lambda \rceil + 1$.

The exact value formula for $p_f^{(nf)}$, for the no faulty nodes configuration:

$$p_f^{(nf)} = \sum_{m_{0011}=0}^{T-1} \binom{m}{m_{0011}} \left(\frac{1}{3}\right)^{m_{0011}} \left(\frac{2}{3}\right)^{m-m_{0011}}$$

The exact value formula for $p_{f,\uparrow}^{(S)}$, for the S faulty configuration:

$$p_{f,\uparrow}^{(S)} = p_f^{(S)} + \left(1 - \sum_{l_3=T}^{m-T} \sum_{l_1=T-Q}^{m-Q-l_3} \binom{m}{l_3, l_1, m-l_1-l_3} \left(\frac{1}{3}\right)^m\right)$$

$$p_f^{(S)} = \sum_{l_3=T}^{m-T} \sum_{l_1=T-Q}^{m-Q-l_3} \binom{m}{l_3, l_1, m-l_1-l_3} \left(\frac{1}{3}\right)^m 2^{-Q}$$

The exact value formula for $p_{f,\uparrow}^{(R)}$, the R_0 faulty configuration:

$$p_{f,\uparrow}^{(R)} = p_f^{(R)} + \sum_{l_1=m-T+1}^{m} \binom{m}{l_1} \left(\frac{1}{3}\right)^{l_1} \left(\frac{2}{3}\right)^{m-l_1}$$

$$\begin{aligned} p_f^{(R)} = & \sum_{l_1=T}^{m-T} \sum_{l_2=0}^{T-Q} \binom{m}{l_1, l_2, l_3} \left(\frac{1}{3}\right)^{l_1} \left(\frac{1}{6}\right)^{l_2} \left(\frac{1}{2}\right)^{l_3} \left[\sum_{k=T-Q+1-l_2}^{T-l_2} \binom{T-l_2}{k} \left(\frac{2}{3}\right)^k \left(\frac{1}{3}\right)^{T-l_2-k}\right] \\ & + \sum_{l_1=T}^{m-T} \sum_{l_2=T-Q+1}^{m-l_1} \binom{m}{l_1, l_2, l_3} \left(\frac{1}{3}\right)^{l_1} \left(\frac{1}{6}\right)^{l_2} \left(\frac{1}{2}\right)^{l_3} \\ & + \sum_{l_1=0}^{T-1} \binom{m}{l_1} \left(\frac{1}{3}\right)^{l_1} \left(\frac{2}{3}\right)^{m-l_1} \end{aligned}$$

C Example Run with Decoherence Errors

This appendix traces a single run (with $m = 1$) of the protocol where an honest sender S broadcasts the bit $x = 0$ to two honest receivers. However, noise is present, and its consequence on the protocol is highlighted.

C.1 T_1 Decoherence Error

In this run, a T_1 decoherence error occurs and influences the outcome of the protocol as follows:

1. **Invocation and Noise:** The process begins with S measuring its qubits as '00' and sending the check set $\rho = \{0\}$ to both receivers. In a noiseless scenario, the receivers' qubits would be in the state $|11\rangle$. However, we assume that during the idle time before measurement, the qubit held by receiver R_0 is affected by T_1 decoherence. This noise results in the joint state of the receivers' qubits being $|01\rangle$.

2. **Check Phase:** The receivers measure their qubits, but their paths diverge due to the error.
    - R_0 measures its corrupted qubit and gets the outcome '0'. It then checks if this outcome is opposite to the sender's bit $x = 0$. The check fails, because '0' is not the opposite of '0'.
    - R_1's qubit is unaffected. It measures the outcome '1', which is the correct opposite value. Its check passes.
3. **Cross calling and check phase:** Since its check failed, R_0 must abort and sets its final output to $x_0 = \perp$. Since its check passed, R_1 sets its output to $x_1 = 0$. R_0 sends its outcome and check set to R_1, who will then see that R_0 has aborted; thus, it will retain its original value '0'.

According to the Weak Broadcast success conditions from Section 2.2, when a non-faulty sender broadcasts x_s, all non-faulty receivers must also output x_s. Since R_0 outputs $\perp$, this condition is not met, and the protocol run is considered a failure.

C.2 T_2 Decoherence Error

In this run, a T_2 decoherence error occurs and influences the outcome of the protocol as follows:

1. **Invocation and Noise:** The process begins with S measuring its qubits as '00' and sending the check set $\rho = \{0\}$ to both receivers. In this scenario, S's first qubit and R_1's qubit are subject to dephasing. The quantum state is now:
$$|\psi_Z\rangle = \frac{1}{2\sqrt{3}} \left(-2\,|0011\rangle + |0101\rangle - |0110\rangle + |1010\rangle - |1001\rangle - 2\,|1100\rangle\right)$$
2. **Check Phase:** The receivers measure their qubits, but their paths diverge due to the error.
    - R_0 measures its qubit and gets the outcome '1'. It then checks if this outcome is opposite to the sender's bit $x = 0$. The check passes.
    - R_1 also measures a '1'. Its check also passes.
3. **Cross calling and check phase:** R_0 and R_1 both agree on the sender's bit $x = 0$, so no changes are made to the outputs in this phase.

According to the Weak Broadcast success conditions from Section 2.2, when a non-faulty sender broadcasts x_s, all non-faulty receivers must also output x_s. Since all nodes output x, this condition is met, and the protocol run is considered a success.