

Article

Leadership Uniformity in Timeout-Based Quorum Byzantine Fault Tolerance (QBFT) Consensus

Andreas Polyvios Delladetsimas , Stamatis Papangelou , Elias Iosif *  and George Giaglis 

Department of Digital Innovation, School of Business, University of Nicosia, Nicosia 2417, Cyprus; delladetsimas.a@unic.ac.cy (A.P.D.); papangelou.m@unic.ac.cy (S.P.); giaglis.g@unic.ac.cy (G.G.)

* Correspondence: iosif.e@unic.ac.cy

Abstract

This study evaluates leadership uniformity—the degree to which the proposer role is evenly distributed among validator nodes over time—in Quorum-based Byzantine Fault Tolerance (QBFT), a Byzantine Fault-Tolerant (BFT) consensus algorithm used in permissioned blockchain networks. By introducing simulated follower timeouts derived from uniform, normal, lognormal, and Weibull distributions, it models a range of network conditions and latency patterns across nodes. This approach integrates Raft-inspired timeout mechanisms into the QBFT framework, enabling a more detailed analysis of leader selection under different network conditions. Three leader selection strategies are tested: Direct selection of the node with the shortest timeout, and two quorum-based approaches selecting from the top 20% and 30% of nodes with the shortest timeouts. Simulations were conducted over 200 rounds in a 10-node network. Results show that leader selection was most equitable under the Weibull distribution with shape $k = 0.5$, which captures delay behavior observed in real-world networks. In contrast, the uniform distribution did not consistently yield the most balanced outcomes. The findings also highlight the effectiveness of quorum-based selection: While choosing the node with the lowest timeout ensures responsiveness in each round, it does not guarantee uniform leadership over time. In low-variability distributions, certain nodes may be repeatedly selected by chance, as similar timeout values increase the likelihood of the same nodes appearing among the fastest. Incorporating controlled randomness through quorum-based voting improves rotation consistency and promotes fairer leader distribution, especially under heavy-tailed latency conditions. However, expanding the candidate pool beyond 30% (e.g., to 40% or 50%) introduced vote fragmentation, which complicated quorum formation in small networks and led to consensus failure. Overall, the study demonstrates the potential of timeout-aware, quorum-based leader selection as a more adaptive and equitable alternative to round-robin approaches, and provides a foundation for developing more sophisticated QBFT variants tailored to latency-sensitive networks.

Keywords: QBFT; consensus algorithms; permissioned systems; leadership uniformity; blockchain optimization; distributed consensus



Academic Editor: Carson K. Leung

Received: 13 May 2025

Revised: 14 July 2025

Accepted: 22 July 2025

Published: 24 July 2025

Citation: Delladetsimas, A.P.; Papangelou, S.; Iosif, E.; Giaglis, G. Leadership Uniformity in Timeout-Based Quorum Byzantine Fault Tolerance (QBFT) Consensus. *Big Data Cogn. Comput.* **2025**, *9*, 196. <https://doi.org/10.3390/bdcc9080196>

Copyright: © 2025 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the Creative Commons

Attribution (CC BY) license

(<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Consensus algorithms are fundamental to blockchain and distributed systems, enabling agreement even in the presence of malicious nodes [1]. As the technology matures, private and permissioned blockchain systems are seeing increased use in settings where secure data sharing and efficient coordination are essential [2]. According to a 2023 ProveAI

(formerly Casper Labs) survey, 87% of 603 business leaders in the UK, China, and the United States indicated plans to invest in blockchain technology within a year. China showed the strongest intent, with 55% of the respondents very likely to invest, followed by the US (47%) and the UK (42%) [3]. Despite this growing interest, recent studies suggest that actual adoption remains relatively limited, primarily due to a lack of blockchain-related knowledge in combination with unclear technical standards [4]. A systematic review of studies on blockchain adoption by AlShamsi et al. [5] found that organizational adoption decisions are often influenced by factors such as lack of confidence in blockchain reliability, difficulty integrating with existing systems, and high implementation costs. In this respect, improving the performance of consensus algorithms becomes an important technical consideration, as issues such as low throughput or high latency, reported in both experimental and applied evaluations of BFT algorithms such as QBFT, PBFT, and Clique, can significantly impact blockchain scalability and hinder enterprise adoption [6–8].

Building on this context, the present study examines the Quorum Byzantine Fault Tolerance (QBFT) algorithm, a consensus mechanism used in permissioned systems, where a limited number of trusted validators are responsible for maintaining the network through a voting-based process [9,10]. This work focuses on the concept of “leadership uniformity” introduced by [11], a metric that evaluates how evenly leadership roles are distributed among nodes over time.

Since QBFT typically uses a round-robin scheme for leader rotation, it does not account for network conditions or performance variability between nodes. Inspired by techniques found in other consensus algorithms such as Raft [12], this study introduces a timeout-based leader selection mechanism within QBFT. To simulate network delays and computational latencies, timeout values are sampled from four distributions: Uniform, normal, lognormal, and Weibull. These values serve as the sole input to the leader selection logic, allowing for controlled simulations of network behavior.

The research addresses two primary goals: The first is to assess the impact of different timeout distributions on leadership uniformity, and the second is to evaluate the role of quorum-based voting under these conditions. In order to examine both, three leader selection strategies were implemented: One that directly selected the node with the minimum timeout, and two others that introduced quorum voting, where the leader was elected from among the 20% or 30% of nodes with the shortest timeouts. The underlying rationale is that electing leaders from these subsets helps reduce selection bias caused by extreme latency values, promoting a more consistent distribution of leadership over time, while also enabling a direct comparison between purely timeout-driven selection and scenarios where rotation is influenced by QBFT’s quorum voting.

Experiments were conducted on a network of 10 nodes across 200 election rounds, using the four timeout distributions mentioned above to simulate diverse network conditions. The results indicate that uniform timeout assignment does not consistently lead to fair leadership distribution. More specifically, the Weibull distribution with shape $k = 0.5$, which was chosen for its moderate skewness [13] and its ability to reflect delay patterns observed in real-world systems [14,15], achieved the best overall fairness and leader rotation consistency. This suggests that leadership distribution benefits from timeout behavior that more accurately reflects real-world network conditions. Furthermore, incorporating quorum-based selection from the top 20% and 30% of nodes further improved the balance of leadership across rounds. In this sense, consistently favoring the nodes with the shortest timeouts may restrict leader rotation over time, while introducing limited randomness helps mitigate this effect and improve fairness. These findings suggest that incorporating timeout-aware strategies into QBFT can complement or enhance round-robin selection by making leader election more adaptive to network conditions, highlighting how timeout-

driven randomness may contribute to fairer leadership distribution in permissioned BFT consensus and informing the design of future QBFT variants.

From a broader perspective, the present work relates to the field of distributed and decentralized computing by enhancing leader selection strategies in Byzantine Fault-Tolerant (BFT) consensus. By modeling network latencies and evaluating adaptive quorum-based mechanisms, it offers design insights for building more resilient and equitable consensus protocols in latency-sensitive systems.

The main contributions of this paper are as follows:

- Propose and evaluate a simulated timeout-based leader selection approach for QBFT, combining node timeouts with quorum-based voting to model network variability and assess its impact on fairness.
- Simulate and compare four timeout distributions (uniform, normal, lognormal, Weibull) to assess their impact on leadership uniformity.
- Demonstrate through simulation that the Weibull distribution, when combined with structured quorum-based voting, results in improved fairness and lower error rates compared to other tested distributions.
- Provide simulation-backed insights to inform the design of more adaptive, latency-aware BFT consensus strategies for permissioned blockchain networks.

Section 2 of this paper provides a brief historical overview of the development of BFT algorithms and examines the use of BFT-Proof of Authority (PoA) mechanisms in private blockchain networks. Next, Section 3 provides a detailed overview of the simulation framework, explaining the rationale behind timeout-based voting and the various testing scenarios that were implemented. Section 4 presents the results of the study, which are then interpreted and discussed in detail, along with limitations and directions for future research in Section 5. Finally, Section 6 summarizes the key findings.

2. Background and Related Work

This section provides a brief overview of BFT algorithms, focusing on their evolution from the Byzantine Generals Problem (BGP) to modern protocols, such as IBFT 2.0 and QBFT. It also highlights key characteristics of QBFT and summarizes recent studies comparing its performance to other consensus algorithms in permissioned blockchain networks.

2.1. A Brief History of BFT Algorithms

Originally introduced in 1982 by Lamport et al. [16], the BGP posed a fundamental challenge in achieving distributed consensus in the presence of malicious actors. It describes a scenario in which a group of generals want to coordinate an attack on an enemy city, but among them are traitors who attempt to spread misinformation and disrupt coordination. The authors proposed the concept of “Byzantine agreement”—a process where generals exchange multiple rounds of messaging to detect and filter out false information. By allowing a majority of honest participants to reach a reliable decision, the work established a key framework for designing secure fault-tolerant distributed systems. Figure 1 below provides a simple illustration of the BGP.

Perhaps the most important insight from the work of Lamport et al. [16] is that solving the Byzantine Generals Problem in the presence of f faulty nodes requires at least $3f + 1$ total participants. For instance, in order to tolerate a single malicious node, a system must include at least four. This requirement laid the theoretical foundation for Byzantine Fault Tolerance and remains central to all modern BFT algorithms. The BGP has since inspired the development of numerous consensus mechanisms designed to tolerate Byzantine

faults. Among the earliest were the Oral Messages and Signed Messages algorithms, both introduced in the same work.

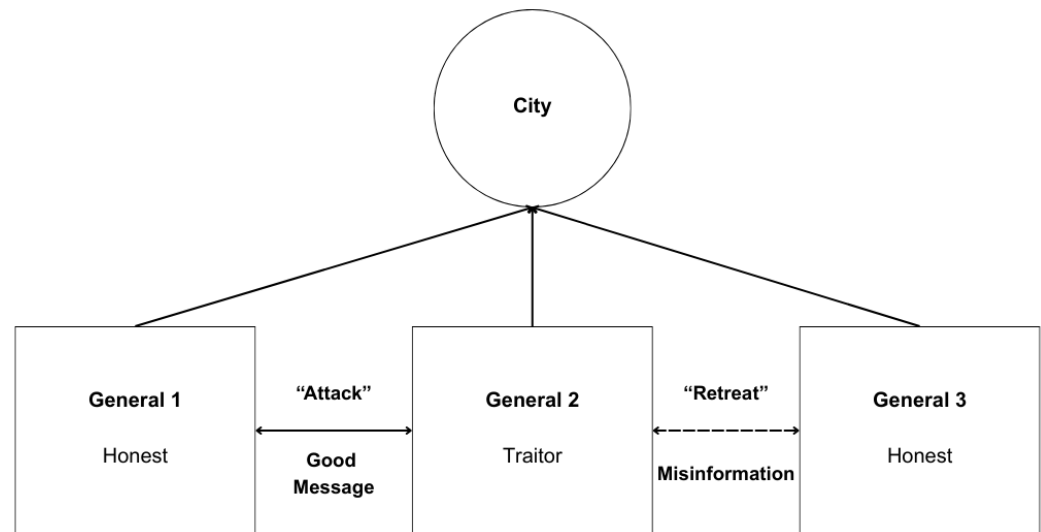


Figure 1. Illustration of the Byzantine Generals Problem.

In 1999, Castro and Liskov [17] proposed “Practical Byzantine Fault Tolerance” (PBFT), an improved BFT algorithm that significantly enhanced the scalability of earlier approaches. In particular, it reduced message complexity from exponential $O(2^n)$ to quadratic ($O(n^2)$) and introduced a three-phase communication process (Pre_Prep, Prepare, and Commit), enabling consensus in larger networks while preserving liveness (i.e., the guarantee that new blocks will continue to be added to the chain). These design choices made PBFT a widely adopted reference point for modern BFT implementations.

As outlined by Zhong et al. [18], nodes in distributed systems can be categorized into “Byzantine fault nodes” and “non-responsive nodes”. The former refers to malicious nodes that actively try to disrupt consensus by sending misleading messages, whereas the latter fail to respond. BFT protocols are specifically designed to mitigate Byzantine faults and ensure that consensus remains achievable.

In this respect, a prominent family of BFT consensus protocols, known as quorum-based BFT protocols, emerged. These protocols operate with a fixed set of validators who take turns proposing and verifying blocks. Once a proposal is made, validators cast their votes, and if a quorum is reached (typically $2f + 1$), the block is committed. At this point, a “quorum certificate” is generated, serving as proof that a supermajority agreed on the proposal. This prevents forks and ensures only agreed-upon blocks are appended to the main chain [19].

Hybrid Quorum (HQ), introduced by Cowling et al. [20], was the first implementation that managed to successfully integrate BFT logic with quorum-based execution. Earlier quorum systems such as “Query/Update” (Q/U) [21] required $5f + 1$ participants to tolerate f Byzantine faults due to their stricter quorum intersection requirements. HQ reduced this to the theoretical minimum of $3f + 1$ by introducing a more efficient two-phase write protocol and using BFT-based coordination only when conflicts arise. In doing so, this hybrid model laid the groundwork for future consensus algorithms that combine quorum mechanisms with BFT techniques.

Subsequently, a major advancement was introduced in June 2017 with the development of “Istanbul BFT” (IBFT) by AMIS Technology [22]. By the end of that year, IBFT had been fully integrated into GoQuorum, an enterprise-focused Ethereum client originally developed by J.P. Morgan and now maintained by ConsenSys [23,24]. IBFT is a leader-based consensus protocol designed to ensure that all honest nodes can agree on a single

version of the blockchain state. Operating in a partially synchronous network, IBFT remains secure even under conditions of unpredictable message delays. It includes a round-change mechanism to replace failed leaders, but its liveness cannot be guaranteed under asynchronous network conditions, as a delayed or faulty leader may prevent the protocol from making progress if the process is not initiated quickly enough [23].

To overcome these limitations, ConsenSys Quorum released IBFT 2.0 [24], a refined version that guarantees immediate finality and enhances liveness through two key mechanisms: A dynamic validator set that enables membership changes by majority vote, and a more responsive leader replacement process that maintains consensus even under network disruptions. Together, these enhancements help eliminate consensus stalls and allow the protocol to maintain progress even in unstable conditions [24,25].

More recently, QBFT emerged as a consensus mechanism tailored to permissioned blockchain networks, implemented in both GoQuorum and Hyperledger Besu. Besu is a Java-based Ethereum client maintained by the Hyperledger Foundation and is also a widely adopted option for permissioned blockchain applications. Both clients recommend QBFT, as it improves upon IBFT 2.0 by further optimizing fault tolerance and resilience, addressing the known limitations of its predecessor [9,10,23]. This improvement is primarily due to a more responsive view-change mechanism, which refers to the process of replacing the current leader. It enables the system to automatically advance to the next consensus round when quorum is not reached. QBFT retains the round-robin leader model and preserves IBFT's features such as dynamic validator management and block finality. It also preserves Byzantine fault tolerance, withstanding up to $\left\lfloor \frac{n-1}{3} \right\rfloor$ faulty nodes in a network of n validators, and maintains a message complexity of $O(n^2)$, which can limit the scalability in larger networks [9,26].

Both QBFT and IBFT are implemented within a PoA framework, where a small group of pre-selected, trusted validators are responsible for validating transactions and creating blocks [10,25]. This hybrid model combines BFT's fault tolerance with PoA's efficiency and trusted validator assumptions, making it particularly well suited for private and permissioned blockchain networks.

2.2. Comparative Analysis of BFT Protocols in Permissioned Blockchain Networks

Various recent studies have evaluated QBFT's performance, scalability, and security in comparison to alternative consensus mechanisms. This subsection reviews key findings from multiple papers and theses, highlighting the algorithm's advantages and limitations in practical implementations.

Fan et al. [6] evaluated the performance of BFT-PoA algorithms used in Hyperledger Besu: QBFT, IBFT 2.0, and Clique (another PoA-based consensus mechanism [27]). More specifically, they analyzed various metrics, including transaction send rate, network size, load balancing, consensus time, and block time, identifying the latter two as the most significant factors influencing overall performance. They also found that QBFT outperformed the alternatives in terms of scalability, maintaining stable performance with up to 14 validators. However, they also noted key limitations of the algorithm, including transaction execution overhead, state update inefficiencies, and message complexity. Regarding Clique, the study showed that while shorter block times improved throughput, they also increased fork rates, whereas longer block times caused delays in transaction processing. Furthermore, a larger block size increased throughput but introduced processing delays. The study concluded that QBFT offers a strong balance between performance and consistency, but further optimization is required to address computational overhead, particularly in Besu's implementation.

Similarly, Gerrits et al. [7] examined BFT consensus protocols as implemented in Hyperledger Sawtooth and Ethereum-based platforms used in private and consortium blockchain settings. Their study evaluated network performance based on transaction throughput, node participation, and message complexity while also considering the trade-offs of the “CAP theorem” (Consistency, Availability—i.e., the ability to respond to requests even during failures, and Partition Tolerance—i.e., the ability to function correctly despite network communication failures). Their results identified inconsistencies in Clique, revealing that although the algorithm achieved high throughput and fast block production, it was also unstable at times due to unresolved forks. This can be attributed to Clique’s round-robin leader selection process, where validators take turns proposing blocks. In such cases, network delays or inconsistencies in validator views can lead to simultaneous proposals, resulting in forks. Since Clique lacks finality, these forks remain unresolved until chain reorganization occurs, when nodes eventually agree on the longest chain. This delay in resolving forks reduces transaction reliability, as blocks may be temporarily accepted but later discarded [27]. In contrast, QBFT was found to provide a balance between performance and scalability, while PBFT prioritized consistency at the expense of lower throughput. In general, the study identified key implementation challenges, such as message communication overhead and consensus complexity. Both IBFT 2.0 and QBFT exhibited higher partition tolerance, but were characterized by lower availability, as they required $\frac{n}{3}$ validator responses per consensus round. They demonstrated moderate efficiency but required optimizations when it comes to hash computation. Finally, the highest communication overhead was exhibited by Sawtooth PBFT due to its multi-phase consensus process.

Further supporting these observations, Lopes [28] evaluated QBFT and IBFT 1.0, noting that both BFT protocols delivered similar performance across different metrics. Although none of the GoQuorum consensus mechanisms scaled beyond 17 nodes in private transactions, the study noted that QBFT mitigated IBFT’s known consensus stalls, making it the preferred option between the two. The study concluded that QBFT is best suited for networks that require BFT, whereas Raft is a better alternative for highly trusted environments that prioritize performance.

Focusing beyond general scalability concerns, Catarino [8] compared QBFT and Clique in the context of Peer-to-Peer (P2P) renewable energy trading using Hyperledger Besu and Caliper. The study found that QBFT significantly outperformed Clique in both throughput and latency due to its shorter block time (2 s) and efficient consensus process. In contrast, Clique’s longer block time (15 s) led to higher latency and lower throughput. Furthermore, QBFT demonstrated better scalability under high transaction loads, making it more suitable for high-frequency applications. This study also emphasized the critical role of consensus parameters, such as block time and request timeout, in optimizing network performance, but noted limitations due to the simulated environment and suggested further real-world testing to validate QBFT’s scalability advantages.

While QBFT offers stable performance and scalability for permissioned systems, security remains a critical consideration. Wong et al. [19] analyzed vulnerabilities in BFT consensus protocol implementations, revealing the challenges associated with applying theoretical models to real-world systems. By examining various BFT protocols, as well as DAG-based (Directed Acyclic Graphs) algorithms, the study identified security risks, including synchronization issues, cryptographic vulnerabilities and logical flaws. The authors noted that many BFT protocols lack detailed specifications, which can contribute to security problems. Overall, the study underscored the need for continuous testing, automated validator set updates, and comprehensive specifications to improve the resilience of BFT-based consensus mechanisms. Although the authors did not specifically study QBFT

or its variants, their analysis of different systems such as Narwhal and Paxos offers valuable lessons for improving QBFT-like protocols.

From the studies analyzed above, QBFT emerges as a stable consensus mechanism for private blockchain networks that provides a balance between performance and security. It improves on IBFT by preventing consensus stalls while maintaining similar performance levels. Furthermore, QBFT's ability to effectively support up to 14 validators, alongside its superior throughput and lower latency compared to Clique, makes it a suitable solution for use cases involving high-frequency transactions. However, despite its advantages, its quadratic message complexity poses scalability challenges for larger networks. Ultimately, while QBFT ensures strong Byzantine fault tolerance, its computational overhead must be carefully managed in order to maximize its effectiveness. With that said, QBFT has been widely adopted in real-world permissioned blockchain applications, including secure medical prescription management systems [29], P2P energy trading systems [8], and invoice management systems [30]. As discussed earlier, it serves as a core consensus mechanism in both GoQuorum and Hyperledger Besu [9,10]. Beyond traditional enterprise use, QBFT is also being utilized in demanding environments. One example is the Digi4Eco project [31], which integrates IoT and blockchain technology in order to create a digital twin for real-time ecological monitoring in marine ecosystems [2].

As summarized across the studies in Table 1, QBFT stands out as a practical solution for private blockchain networks, providing a high degree of performance and Byzantine fault tolerance. However, its scalability challenges should be addressed in future optimizations to ensure it remains a viable option for larger-scale deployments.

Table 1. Overview of comparative results for BFT protocols in the recent literature.

Study	QBFT	IBFT ^a	PBFT	Clique
Fan et al. [6]	In Besu, QBFT maintains stable performance with up to 14 validators and shows slightly better scalability than IBFT 2.0 and Clique. However, computational overhead remains a key challenge due to state update inefficiencies and message complexity.	IBFT 2.0 delivers moderate throughput but suffers from inefficient state updates under high transaction volumes. Its performance declines as the number of validators increases.	–	Clique improves throughput with shorter block times but increases fork rates. Longer block times reduce forks but increase latency. Larger block sizes enhance throughput but introduce processing delays.
Gerrits et al. [7]	QBFT offers a practical balance between performance and scalability but faces scalability limits due to quadratic message complexity and hash computation overhead.	IBFT 2.0 remains stable under normal conditions but is vulnerable to consensus stalls in congested or unstable networks. Its availability is also impacted by the requirement for $\frac{N}{3}$ validator responses per round.	PBFT prioritizes consistency and provides deterministic finality, but its multi-phase process creates significant communication overhead, resulting in lower throughput.	Clique achieves fast block production through round-robin leader selection but lacks finality. Network delays can lead to multiple validators proposing blocks simultaneously, causing persistent forks and reducing reliability.
Lopes [28]	QBFT mitigates IBFT's consensus stalls while maintaining comparable performance. It provides a balanced trade-off between security and efficiency but requires further improvements in response time under high-latency conditions.	IBFT 1.0 performs similarly to QBFT but is still susceptible to consensus stalls, especially in network environments with high latency or instability.	–	–

Table 1. Cont.

Study	QBFT	IBFT ^a	PBFT	Clique
Catarino [8]	QBFT demonstrates high throughput and low latency under simulated high-load scenarios. Its short block period and optimized timeout parameters improve transaction finalization and scalability.	–	–	Clique shows reduced scalability due to longer block times, which increase latency and reduce throughput. Performance also varies significantly with different transaction volumes.

^a All studies analyze IBFT 2.0, except [28], which examines IBFT 1.0.

3. Materials and Methods

This section outlines the approach used to analyze leader selection fairness in the QBFT algorithm. The version implemented in this study is a simplified conceptual model, focusing on key aspects of the algorithm, such as leader election, message handling, and quorum validation for a more targeted analysis. It begins by detailing the QBFT protocol, motivates the shift to timeout-based leader selection, and then presents the experimental settings and evaluation metrics.

3.1. QBFT Algorithm Analysis

This subsection examines key aspects of the baseline QBFT implementation, including its leader selection mechanism, quorum logic, and overall computational and communication complexity. The purpose is to evaluate the behavior of the baseline algorithm and identify key limitations and opportunities to enhance performance and fairness.

The algorithm follows a three-phase commit protocol: “Pre_Prepate”, “Prepare”, and “Commit”, in which each node exchanges messages and verifies that the leader’s proposal is valid and that a sufficient number of nodes agree before proceeding to the next phase. This results in a computational complexity of $\mathcal{O}(n)$ per node for each phase, where n is the number of participating nodes. The consensus process is illustrated in Figure 2, depicting the progression through these three phases.

Although QBFT is typically designed to use a round-robin leader selection process [9,10], this study modifies the protocol to support a timeout-based leader selection model better suited for evaluating fairness under varying network conditions (see Section 3.2).

As previously mentioned, quorum verification in QBFT ensures that at least two-thirds of the participating nodes support a proposal before a block can be finalized. This threshold is defined as $\frac{2n}{3}$, where n is the total number of nodes. To determine whether this condition is met, each node iterates through the received messages and verifies their signatures, introducing a linear computational cost ($\mathcal{O}(n)$) per phase.

However, scalability problems begin to arise primarily in the algorithm’s communication model, as in each phase, every node in the network transmits its message to all others [6]. Since each node sends messages to $n - 1$ other nodes per phase, the number of message exchanges grows quadratically as the size of the network increases, resulting in a communication complexity of $\mathcal{O}(n^2)$.

In summary, the per-node computational cost of QBFT is $\mathcal{O}(n)$ for message handling and quorum checks, while the communication overhead is $\mathcal{O}(n^2)$ due to full message broadcasting in each phase. These properties make QBFT suitable for smaller networks, but its efficiency decreases as the number of nodes grows. In particular, the quadratic

message growth becomes a limiting factor in larger networks, as bandwidth and processing demands increase rapidly with node count.

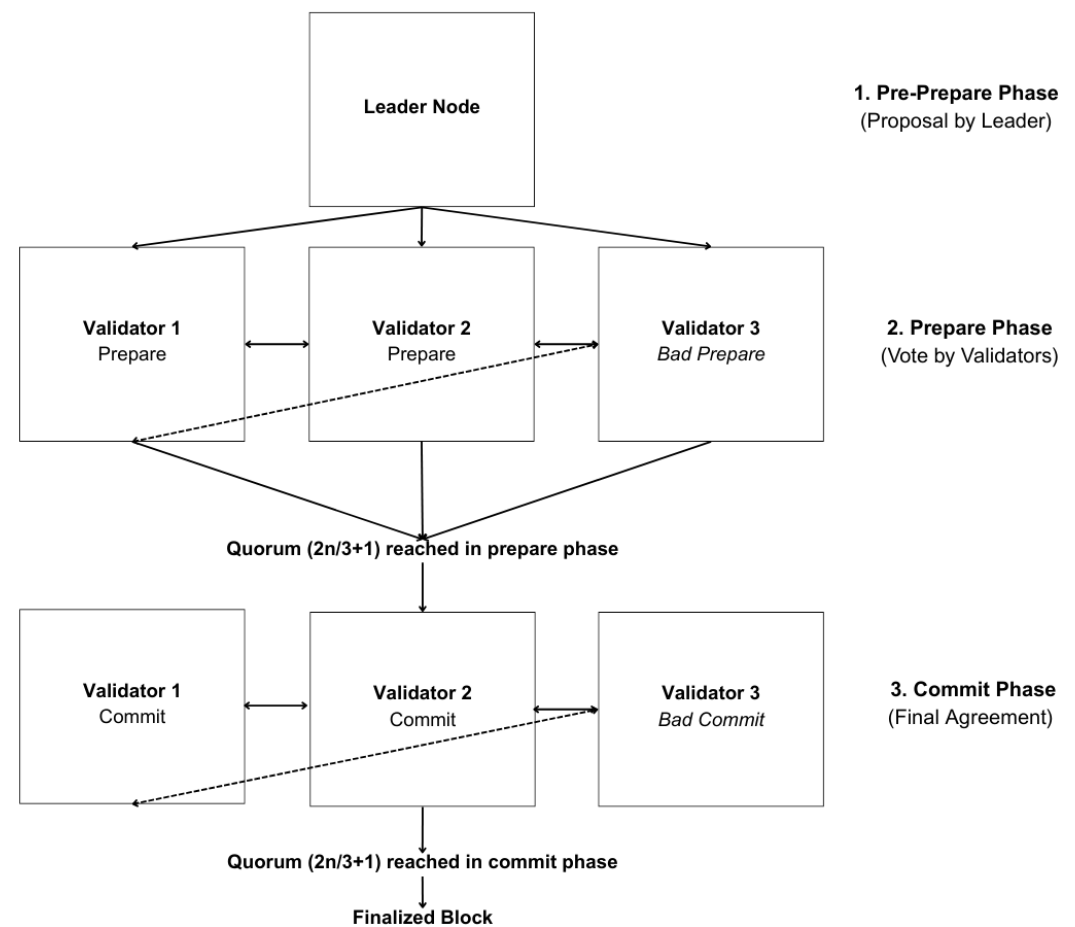


Figure 2. Illustration of the QBFT consensus process.

Based on this analysis, the following aspects of QBFT present opportunities for improving performance and scalability:

1. **Optimized Communication:** Reducing all-to-all broadcasting through message aggregation or grouping (i.e., sending messages to selected subsets of nodes) could help lower communication overhead.
2. **Adjustable Quorum Size:** Dynamically adjusting the quorum threshold based on network conditions could improve scalability while maintaining fault tolerance.
3. **Improved Leader Selection:** Preventing the repeated selection of the same nodes as leaders could lead to a fairer distribution of leadership over time.

Although communication and quorum improvements are important, this paper focuses specifically on refining leader selection to improve fairness via leadership uniformity. To this end, the QBFT implementation is extended to investigate whether leadership can be distributed more evenly over time by replacing the standard deterministic leader rotation with an adaptive, timeout-driven approach. More specifically, the study evaluates how effectively the proposed leader selection mechanism promotes balanced leadership across the consensus process by measuring deviations from an ideal leader distribution.

3.2. Integrating Timeout Leader Selection with QBFT Quorum Voting

This subsection outlines the simulation design, including how timeout values were generated, how leader selection was performed, and how fairness was evaluated.

This paper builds on the methodology proposed by Iosif et al. [11], who investigated leadership uniformity in the Raft consensus algorithm. According to the authors, leadership uniformity refers to the fair and balanced distribution of leadership roles among all participating nodes over time. Their approach is adapted in this study to evaluate leader selection fairness in the context of QBFT.

As noted in Section 2, traditional QBFT relies on a round-robin process for leader selection, which ensures that validators assume leadership roles in an ordered manner. While this approach guarantees equal participation, it does not account for variations in network conditions or computational capacity across nodes. Accordingly, this study introduces a timeout-based leader selection approach, where t_f represents the maximum time a node is willing to wait before timing out and initiating a leader election. Instead of rotating leadership sequentially, nodes with shorter follower timeouts t_f , reflecting faster network responsiveness, have a higher chance of being selected as leaders, thereby allowing the system to adjust to real-time network conditions. This approach is evaluated using three selection strategies: Minimum-timeout and randomized voting among the 20% and 30% of nodes with the shortest timeouts.

To evaluate the effects of timeout-based leader selection, each node is assigned a random timeout value t_f drawn from one of four probability distributions: Uniform, normal, lognormal, and Weibull. These distributions are chosen to simulate a range of network conditions, from stable environments with consistent latencies to highly variable systems characterized by sporadic delays. This design follows prior work by Iosif et al. [11], who used similar distributions to study timing behavior in distributed consensus. The normal distribution serves as a general baseline, while the lognormal distribution captures empirically observed latency patterns in distributed systems.

This methodology incorporates the Weibull distribution as an alternative model for follower timeouts due to its flexibility in modeling real-world delay characteristics. A key parameter influencing Weibull's behavior is k , which controls skewness. As shown by Versluis and Straetmans [13], the distribution becomes increasingly right-skewed as k decreases below 1, with skewness gradually diminishing as k approaches 1. Lower values, such as $k = 0.3$, correspond to highly unstable networks with frequent small delays and rare latency spikes. In contrast, values closer to 1 (but still $k < 1$) tend to result in more consistent response times. In the context of this study, decreasing k leads to heavier tails and greater variability, which can result in premature leader election behavior. When applied to follower timeouts, this may compromise fairness, either by triggering elections too early or introducing rare but extreme delays, both of which can lead to irregular or biased leader selection across rounds. Meanwhile, higher values of k may underrepresent early trigger behaviors observed in practice.

The midrange values of k closely resemble delay patterns reported in IP-based systems, where short latencies are common but are occasionally interrupted by significant spikes [14,15]. For instance, Papagiannaki et al. [14] observed that single-hop queueing delays could be approximated using a Weibull distribution with shape parameters ranging from 0.6 to 0.82. Although these previous studies focus on network-level behavior, the variability in queueing delays they describe is also relevant to permissioned distributed systems, where consensus performance is sensitive to latency fluctuations and timeout mechanisms rely on timely message delivery. Furthermore, since the Weibull distribution with $k < 1$ and the lognormal distribution are heavy-tailed and allow for occasional extreme delays, they are commonly used in areas such as risk theory and queueing theory to model rare but impactful events, such as large losses or extended waiting times [32,33]. Consequently, these distributions provide a useful way to simulate less predictable network

conditions, where infrequent but high-latency timeout values may influence which nodes are selected as leaders.

Based on the above, the present study adopts a fixed value of $k = 0.5$, which falls within this empirically grounded range and represents a moderately skewed configuration. This value captures early-timeout behavior typical of real-world network delays while avoiding extreme variance. While other k values are discussed above to illustrate the Weibull distribution's flexibility, all simulations in this study use $k = 0.5$ to ensure consistency and practical relevance.

The shape parameter's behavior is summarized in Table 2 and illustrated in Figure 3, which plots Weibull probability density functions for various shape parameters on a logarithmic timeout scale, normalized to a mean of 50 ms. The curves demonstrate how changes in k affect the skewness and concentration of the distribution, particularly in the low-timeout region. The x-axis spans timeouts from 0.01 to 150 ms on a logarithmic scale to highlight differences in the early and long-tail behavior of each distribution.

Table 2. Effect of the Weibull shape parameter k on network behavior when $k < 1$.

Range of k	Skewness	Latency Characteristics
$0 < k \leq 0.4$	Very high	Highly unstable networks with extreme delays
$0.4 < k < 0.7$	Moderate	Midrange networks with occasional latency spikes
$0.7 \leq k < 1$	Low	More stable networks with consistent latencies

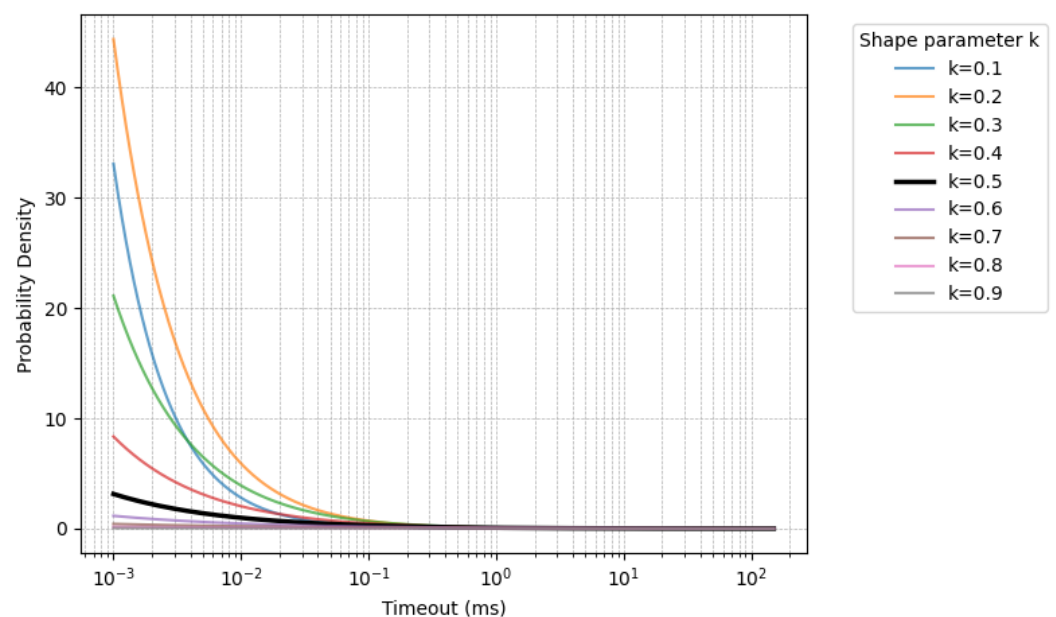


Figure 3. Weibull PDFs for shape parameters $k = 0.1$ – 0.9 (mean timeout $t_f = 50$ ms) on a log-scale. The $k = 0.5$ curve is highlighted.

In order to ensure that leader selection is not solely determined by timeout values, the proposed approach incorporates a structured mechanism consisting of three selection strategies, each evaluated for its impact on leadership uniformity:

1. **Minimum Timeout Selection:** The leader is chosen as the node with the lowest t_f , serving as a baseline. This case provides clear results on the effects of different distributions, as it is purely timeout-based and not influenced by QBFT's quorum voting mechanism.

2. **Randomized Selection from 20% Shortest Timeouts:** A candidate pool is formed from the 20% of nodes with the lowest t_f values. Each node casts a vote for a randomly selected candidate within this pool, with candidates voting for themselves. The candidate receiving the most votes is elected leader. This approach ensures that the process is influenced not only by timeout values, but also by QBFT's quorum-based voting mechanism.
3. **Randomized Selection from 30% Shortest Timeouts:** Similar to the previous case, a candidate pool is formed from the 30% of nodes with the shortest timeouts. The broader selection pool increases variability in voting outcomes, thus amplifying the role of quorum approval and further reducing the dependence of leader selection on exact timeout ranking.

The first method selects the leader deterministically, while the second and third incorporate quorum voting among candidates with the shortest timeouts. These thresholds were chosen to balance variability and quorum formation efficiency, while avoiding excessive vote fragmentation. Collectively, these three leader selection strategies offer a comparative framework to evaluate how timeout-driven randomness and QBFT's quorum mechanism jointly affect leadership fairness and distribution. This deviation from the traditional round-robin QBFT approach provides valuable insight into how timeout-based leader selection and quorum-based validation interact in a permissioned blockchain setting. By integrating network delays and voting dynamics, the proposed method aims to achieve a more adaptive and equitable consensus process. The primary objectives of this experiment are, first, to investigate whether specific timeout distributions result in fairer leader selection, and second, to assess how the quorum mechanism, along with the three proposed selection strategies, influences the overall uniformity of leadership distribution.

Finally, although timeout values in this study are sampled from multiple distributions, the inclusion of the Weibull distribution is further supported by Extreme Value Theory (EVT), which establishes that the minimum of a set of independent and identically distributed random variables, when appropriately scaled, converges to a Weibull-type distribution under suitable conditions [34]. This theoretical result is particularly relevant to the first leader selection strategy, where the node with the minimum timeout is deterministically chosen, providing a strong theoretical basis for modeling this behavior.

3.3. Experimental Settings

This subsection outlines the simulation parameters used to evaluate the fairness of leader selection under varying network conditions. The goal is to understand how different timeout distributions and selection strategies influence the distribution of leadership over time.

The experiments implement the three leader selection strategies and four timeout distributions described in Section 3.2. Given the differences in how each distribution shapes the spread of t_f values, it is useful to examine which configurations lead to more balanced leadership rotation. While some distributions may lead to more frequent changes in leadership, others can cause certain nodes to be chosen repeatedly across rounds.

Since leader selection in QBFT also depends on quorum voting rather than timeout ranking alone, the simulation (specifically in the second and third strategies) requires each candidate to secure the approval of at least $\lceil \frac{2n}{3} \rceil$ nodes in the network. If no quorum is reached, the round is repeated, introducing further variability and helping prevent leadership monopolization. Simulations were conducted using $N = 10$ nodes over $T = 200$ rounds.

To ensure comparability, the mean timeout t_f is fixed across all configurations. To simulate a range of baseline network conditions, leader selection performance is evaluated

across five follower timeout values: $t_f = 1, 10, 50, 500, 1000$ ms. This models environments ranging from highly responsive to high-delay networks.

Additionally, to reflect different levels of network variability, the uniform, normal, and lognormal distributions are evaluated at three standard deviation settings: $\sigma = \frac{1}{2}t_f$, $\frac{1}{4}t_f$, and $\frac{1}{8}t_f$. In contrast, the Weibull distribution is evaluated using a fixed shape parameter $k = 0.5$, while the scale parameter λ is adjusted to preserve the mean $\mathbb{E}[X] = t_f$. Although the standard deviation of the Weibull distribution is not explicitly set, it is determined by k and λ , resulting in a consistent variability level. Consequently, Weibull is evaluated only once per t_f , unlike the other distributions, which are tested at multiple variability levels (see Section 3.2).

Figure 4 visualizes the four timeout distributions used in this study on a linear x-axis scale, each normalized to a mean timeout of $t_f = 50$ ms and a standard deviation of $\frac{1}{4}t_f$. The plot highlights differences in skewness and tail behavior, which play an important role in how timeout values affect leader selection under each strategy.

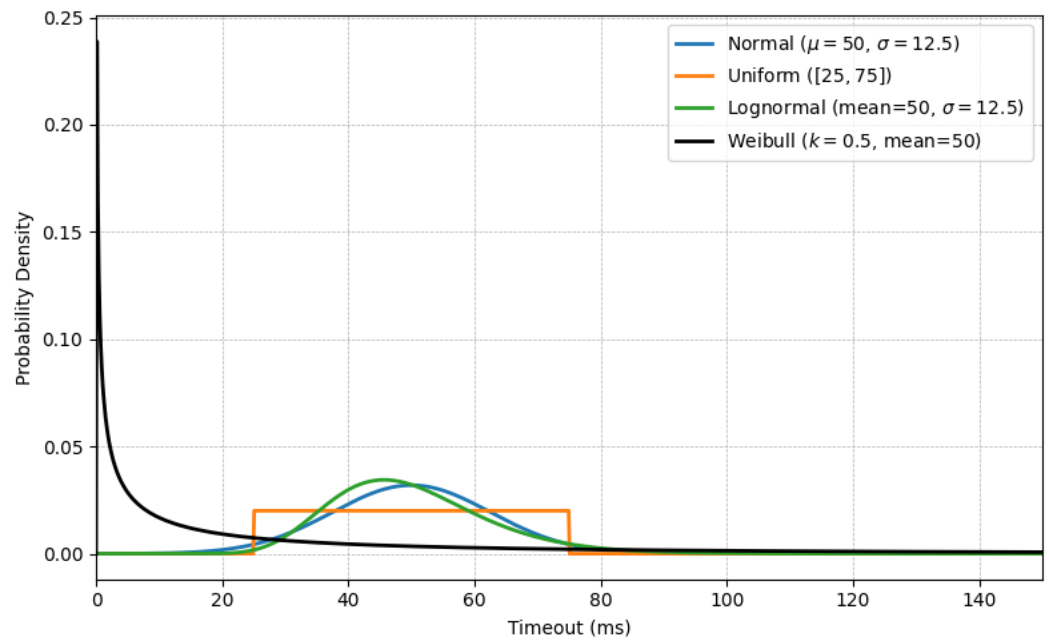


Figure 4. PDFs of the four timeout distributions (mean timeout $t_f = 50$ ms and standard deviation $\sigma = \frac{1}{4}t_f = 12.5$ ms) on a linear scale.

For the uniform distribution, timeout values are sampled from a range $[a, b]$, where:

$$a = t_f - 2\sigma \quad (1)$$

$$\text{and } b = t_f + 2\sigma \quad (2)$$

Thus, the values are drawn from the interval $[t_f - 2\sigma, t_f + 2\sigma]$, which is symmetrically centered around the mean t_f , so the mean is given by $\mu = \frac{a+b}{2} = t_f$. Here, σ defines the spread of the distribution, resulting in a total range of 4σ .

The normal distribution distribution is centered around t_f , with $\mu = t_f$ and standard deviation σ .

For the lognormal distribution, the parameters are adjusted so that the distribution has an expected value (i.e., the mean, denoted $\mathbb{E}[X]$) equal to the desired t_f . The expected value of a lognormal distribution is given by:

$$\mathbb{E}[X] = e^{\mu_{\ln} + \frac{\sigma_{\ln}^2}{2}} \quad (3)$$

Here, $\mu_{\ln}$ and $\sigma_{\ln}$ are the mean and standard deviation of the underlying normal distribution (i.e., the distribution of $\ln(X)$), while σ denotes the desired standard deviation of the lognormal distribution.

Solving for $\mu_{\ln}$ and $\sigma_{\ln}$ in terms of t_f and σ :

$$\sigma_{\ln} = \sqrt{\ln\left(1 + \frac{\sigma^2}{t_f^2}\right)} \quad (4)$$

$$\mu_{\ln} = \ln\left(\frac{t_f^2}{\sqrt{t_f^2 + \sigma^2}}\right) \quad (5)$$

The Weibull distribution allows variability through both the shape and scale parameters. The shape parameter is fixed at $k = 0.5$, and only the scale parameter λ is adjusted in order to maintain a constant mean timeout t_f .

To ensure that $\mathbb{E}[X] = t_f$, the scale parameter is computed as:

$$\lambda = \frac{t_f}{\Gamma\left(1 + \frac{1}{k}\right)}$$

Here, Γ denotes the gamma function, a generalization of the factorial function to real and complex numbers.

The standard deviation of the Weibull distribution is given by:

$$\text{std}(X) = \lambda \sqrt{\Gamma\left(1 + \frac{2}{k}\right) - \left[\Gamma\left(1 + \frac{1}{k}\right)\right]^2} \quad (6)$$

Substituting $k = 0.5$ yields:

$$\lambda = \frac{t_f}{\Gamma(3)} = \frac{t_f}{2} \quad (7)$$

$$\text{std}(X) = \frac{t_f}{2} \cdot \sqrt{\Gamma(5) - [\Gamma(3)]^2} = \frac{t_f}{2} \cdot \sqrt{24 - 4} = \frac{t_f}{2} \cdot \sqrt{20} \quad (8)$$

Thus, the standard deviation is approximately $2.236 \cdot t_f$, establishing a fixed level of variability for the Weibull distribution relative to its mean.

Table 3 summarizes the key properties and parameter settings of the timeout distributions used in the experiments. For all distributions, the mean timeout is set to t_f to ensure consistency across comparisons. For the uniform, normal, and lognormal distributions, this is achieved by varying the standard deviation σ , while the Weibull distribution uses a shape parameter and achieves the desired mean by solving for the appropriate scale parameter.

Table 3. Overview of the timeout distributions utilized in the study.

Distribution	Expected Value Formula	Parameters Used
Uniform	$\mu = \frac{a+b}{2}$	$a = t_f - 2\sigma, b = t_f + 2\sigma$
Normal	$\mu = t_f$	$\mu = t_f, \sigma$
Lognormal	$\mathbb{E}[X] = e^{\mu_{\ln} + \frac{\sigma_{\ln}^2}{2}}$	$\mu_{\ln} = \ln\left(\frac{t_f^2}{\sqrt{t_f^2 + \sigma^2}}\right), \sigma_{\ln} = \sqrt{\ln\left(1 + \frac{\sigma^2}{t_f^2}\right)}$
Weibull	$\mathbb{E}[X] = \lambda \Gamma\left(1 + \frac{1}{k}\right)$	$k = 0.5, \lambda = \frac{t_f}{\Gamma\left(1 + \frac{1}{k}\right)}$

3.4. Leadership Uniformity Evaluation Metrics

Leader selection performance is evaluated by measuring deviation from the ideal leadership uniformity. This uniformity is represented by $\bar{n}_L$, defined as the ideal number of leadership opportunities per node across all trials. It is calculated as follows:

$$\bar{n}_L = \frac{T}{N} \quad (9)$$

where:

- T is the number of successful election trials (rounds) in which exactly one leader was elected.
- N is the total number of nodes.

For this study, the number of consensus rounds is set to $T = 200$, and the number of participating nodes is $N = 10$. Therefore, the ideal number of leadership opportunities per node is $\bar{n}_L = 20$.

To assess leader selection performance under different distributions, two key error metrics are used: Mean Absolute Error (MAE) and Mean Squared Error (MSE).

MAE measures the average absolute deviation between the actual number of times each node is elected as a leader and the ideal value $\bar{n}_L$, representing a perfectly uniform distribution.

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |n_{i,L} - \bar{n}_L| \quad (10)$$

where $n_{i,L}$ represents the number of times node i was elected as a leader across all consensus rounds.

MSE is a variation of MAE that considers the squared differences from the ideal leader distribution:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (n_{i,L} - \bar{n}_L)^2 \quad (11)$$

These metrics (MAE and MSE) are used to quantify leadership fairness across different configurations. The objective is to identify which combinations of timeout distributions and leader selection strategies produce the most uniform distribution of leadership roles under QBFT. Overall, the three primary experimental variables are: The type of timeout distribution, the level of variability (controlled by the standard deviation or shape parameter), and the leader selection strategy. Each configuration is evaluated across five timeout means and assessed using MAE and MSE relative to the ideal uniform rotation.

4. Results

This section presents the evaluation of leader selection performance across different timeout distributions and variability levels. The results are presented in Tables 4, 5, and 6, which report, respectively, the MAE and MSE across all t_f values, the corresponding average performance for each σ (and under Weibull's fixed spread), and an overall performance comparison across all distributions for the minimum timeout selection strategy. Similarly, Tables 7, 8, and 9 present the same metrics for the 20% timeout selection strategy, followed by Tables 10, 11, and 12 for the 30% timeout selection strategy. In the final subsection, Table 13 presents the overall average MAE and MSE values for each timeout selection strategy, while Table 14 compares the average performance across all strategies and distributions, highlighting overall trends.

Table 4. Leader selection performance using minimum timeout selection at different σ and t_f values.

[illegible]

Note: For the Weibull distribution, only a single configuration is shown per t_f , as the standard deviation is determined by the shape parameter $k = 0.5$ (see Section 3).

Table 5. Average MAE and MSE using minimum timeout selection at different σ and t_f values.

[illegible]

Table 6. Overall performance analysis for minimum timeout selection across different distributions.

Distribution	Avg MAE	Avg MSE
Uniform	3.6	22.6
Normal	3.5	22.4
Lognormal	3.3	18.6
Weibull	3.3	14.8

Table 7. Leader selection performance using 20% shortest timeout selection at different σ and t_f Values.

[illegible]

Table 10. Cont.

t_f (ms)	Distribution	$\sigma = \frac{1}{2}t_f$		$\sigma = \frac{1}{4}t_f$		$\sigma = \frac{1}{8}t_f$		Weibull (std $\approx 2.236 \cdot t_f$)	
		MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
50	Uniform	2.2	7.6	3.2	17.8	4.4	25.2	–	–
	Normal	2.2	8.2	4.2	25.0	2.4	9.0	–	–
	Lognormal	2.8	12.0	2.2	8.6	2.6	13.6	–	–
	Weibull	–	–	–	–	–	–	2.4	7.8
500	Uniform	2.4	7.4	2.8	12.0	3.2	15.4	–	–
	Normal	3.6	16.0	4.8	30.4	4.4	26.6	–	–
	Lognormal	3.4	20.6	2.6	9.8	2.8	11.6	–	–
	Weibull	–	–	–	–	–	–	3.2	16.4
1000	Uniform	1.6	4.4	3.0	14.2	3.0	13.4	–	–
	Normal	3.4	14.4	3.0	14.6	3.4	21.6	–	–
	Lognormal	3.8	19.6	4.0	26.4	4.6	25.6	–	–
	Weibull	–	–	–	–	–	–	4.2	23.8

Note: For the Weibull distribution, only a single configuration is shown per t_f , as the standard deviation is determined by the shape parameter $k = 0.5$ (see Section 3).

Table 11. Average MAE and MSE using 30% shortest timeout selection at different σ and t_f values.

Distribution	$\sigma = \frac{1}{2}t_f$		$\sigma = \frac{1}{4}t_f$		$\sigma = \frac{1}{8}t_f$		Weibull (std $\approx 2.236 \cdot t_f$)	
	Avg MAE	Avg MSE	Avg MAE	Avg MSE	Avg MAE	Avg MSE	Avg MAE	Avg MSE
Uniform	2.6	11.8	3.2	17.4	3.8	21.9	–	–
Normal	3.0	13.8	3.5	18.1	2.9	14.6	–	–
Lognormal	3.4	19.2	2.8	14.0	3.3	17.3	–	–
Weibull	–	–	–	–	–	–	2.6	12.0

Table 12. Overall performance analysis for 30% shortest timeout selection across different distributions.

Distribution	Avg MAE	Avg MSE
Uniform	3.2	17.0
Normal	3.1	15.5
Lognormal	3.2	16.8
Weibull	2.6	12.0

Table 13. Overall average MAE and MSE for each timeout selection strategy.

Timeout Selection Strategy	Avg MAE	Avg MSE
Minimum Timeout	3.4	19.6
20% shortest timeouts	3.0	14.5
30% shortest timeouts	3.0	15.3

Table 14. Overall average MAE and MSE per distribution across all timeout selection strategies.

Distribution	Avg MAE	Avg MSE
Uniform	3.2	18.0
Normal	3.3	18.8
Lognormal	3.3	18.0
Weibull	2.7	11.1

4.1. Case 1: Minimum Timeout Selection

In this scenario, leader selection is performed by choosing the node with the absolute minimum timeout. This approach does not involve quorum mechanics and is driven purely by the timeout distributions with no coordination between nodes.

Table 4 presents the results for leader selection using the minimum timeout approach across five follower timeout values (t_f) and three different standard deviations (σ). More specifically, it compares the performance of uniform, normal, lognormal, and Weibull distributions in terms of MAE and MSE, illustrating how each distribution performs under varying timeout conditions and levels of variability. As described in Section 3, quorum voting is not applied in this case, allowing a direct comparison of the effects of the different timeout distributions without the influence of QBFT. For the Weibull distribution, only a single configuration is included for each t_f , due to the fixed spread resulting from its shape parameter. Its MAE and MSE values remain within a competitive range across all timeout values. The normal distribution demonstrates moderate consistency in MAE across midrange timeout values but shows increased variability at higher t_f . The lognormal distribution consistently achieves the lowest MAE at higher timeout values, particularly when σ is small.

Table 5 summarizes the average MAE and MSE leader selection values for the minimum timeout case. For $\sigma = \frac{1}{2}t_f$, the lognormal distribution yields the lowest MAE (2.6) and MSE (11.6). At $\sigma = \frac{1}{4}t_f$, the uniform distribution performs best with an MAE of 3.1 and an MSE of 16.6. In contrast, the highest error is observed for the uniform distribution at $\sigma = \frac{1}{8}t_f$, reaching an MAE of 4.2 and MSE of 28.4. The Weibull distribution, tested at a single variability level, achieves an average MAE of 3.3 and MSE of 14.8.

Table 6 presents the overall performance comparison for the minimum timeout selection method. As shown, the lognormal and Weibull distributions exhibit the lowest errors, with both achieving an MAE of 3.3, as well as an almost identical MSE of 18.6 and 14.8, respectively. Although the Weibull distribution is evaluated one per t_f , its averaged results across all timeout values remain highly competitive. The uniform and normal distributions show slightly higher average errors overall. Finally, for this case, the overall average MAE is 3.4 and the average MSE is 19.6.

4.2. Case 2: Quorum-Based Leader Selection from the Fastest 20% of Nodes

This approach selects a leader from the fastest 20% of nodes based on their timeout values. Unlike the previous method, this strategy combines timeout information with QBFT quorum mechanics, as nodes vote randomly among the eligible candidates and a quorum is required to finalize the election. The results for this strategy across all distributions are provided below.

Table 7 presents the leader selection results for the 20% shortest timeout selection strategy across the five follower timeout values (t_f) and three different standard deviations (σ). This case incorporates quorum-based voting, adding variability to the selection process as nodes vote for candidates from within a pool. At $t_f = 50$, the Weibull distribution achieves the lowest MAE (1.6) across all configurations. The uniform distribution performs best at $t_f = 1000$ with $\sigma = \frac{1}{4}t_f$, where it records the lowest overall MAE of 1.4. Meanwhile, the lognormal distribution remains competitive at higher timeout values under low variability, while the normal distribution shows relatively stable performance but does not achieve the lowest error in any setting.

Table 8 summarizes the average MAE and MSE values for the second case. For $\sigma = \frac{1}{2}t_f$, the uniform distribution achieves the lowest MAE (2.9) and MSE (15.0). At both $\sigma = \frac{1}{4}t_f$ and $\sigma = \frac{1}{8}t_f$, the uniform distribution again performs best, reaching an MAE of 3.1 and an MSE of 15.4 in the former, and an MAE of 2.9 and an MSE of 12.9 in the latter. The Weibull

distribution records the lowest overall error in the table, with an MAE of 2.0 and an MSE of 6.5.

Table 9 presents the overall performance comparison for the 20% shortest timeout selection method. The Weibull distribution records the lowest overall error, with an MAE of 2.0 and an MSE of 6.5 under a single configuration. Among the remaining distributions, the uniform distribution performs best, achieving an overall MAE of 3.0 and an MSE of 14.4, followed closely by the lognormal distribution with an MAE of 3.4 and an MSE of 18.5. Finally, in this case, the overall average MAE is 3.0, and the average MSE is 14.5.

4.3. Case 3: Quorum-Based Leader Selection from the Fastest 30% of Nodes

Similarly to the previous approach, this method integrates timeout-based selection with QBFT quorum mechanics. In this case, the leader is selected from the fastest 30% of nodes, slightly expanding the candidate pool while still requiring quorum agreement. The performance metrics for this strategy across distributions are shown in the table below.

Table 10 presents the leader selection results using the 30% shortest timeout selection strategy, evaluated across five follower timeout values (t_f) and three different standard deviations (σ). As with Tables 4 and 7, it compares the MAE and MSE values for uniform, normal, lognormal, and Weibull distributions. In this case, the normal distribution performs best at $t_f = 10$ and $\sigma = \frac{1}{8}t_f$, achieving the lowest MAE of 1.8. Similarly, the Weibull distribution reaches an MAE of 1.2 at both $t_f = 1$ and $t_f = 50$, the lowest values recorded. On the other hand, the lognormal distribution shows a higher error at higher timeout values, particularly at $t_f = 1000$, where MAE remains above 3.8 across all levels of σ .

Table 11 summarizes the average MAE and MSE values for the final case of 30% shortest timeouts. For $\sigma = \frac{1}{2}t_f$, the uniform distribution records the lowest MAE (2.6) and MSE (11.8). At $\sigma = \frac{1}{4}t_f$, the lognormal distribution performs best, with an MAE of 2.8 and an MSE of 14.0. Finally, under the $\sigma = \frac{1}{8}t_f$ setting, the Weibull distribution achieves the lowest error overall, reaching an MAE of 2.6 and an MSE of 12.0 under its fixed spread per t_f .

Table 12 presents the overall performance comparison for the 30% shortest timeout selection method. As shown, the normal distribution achieves the lowest error overall, with an MAE of 3.1 and an MSE of 15.5. The uniform and lognormal distributions follow closely, both with MAEs of 3.2 and MSEs of 17.0 and 16.8, respectively. In particular, the Weibull distribution records the lowest absolute values in the table, with an MAE of 2.6 and an MSE of 12.0. Finally, in this case, the overall average MAE is 3.0, and the average MSE is 15.3.

4.4. Performance Comparison Across Timeout Selection Strategies and Distributions

This subsection provides a consolidated comparison of the average performance for each selection strategy and timeout distribution. The results highlight overall trends and help identify the most balanced distribution in terms of leadership uniformity.

Table 13 presents the overall average MAE and MSE values for each timeout selection strategy, providing a clear comparison of performance across the minimum timeout selection, followed by the 20% and 30% shortest timeout methods, regardless of the timeout distribution. The second strategy performs best with an MAE of 3.0 and an MSE of 14.5.

Table 14 presents the overall average MAE and MSE values per distribution across all three timeout selection strategies. The results show almost identical performance among the uniform, normal, and lognormal distributions, with average MAEs around 3.3 and MSEs between 18.0 and 18.8. However, the Weibull distribution stands out, achieving a substantially lower average MAE of 2.7 and MSE of 11.1, highlighting its strong and consistent performance.

5. Discussion

This paper evaluated the concept of leadership uniformity, as introduced by Iosif et al. [11], and its interaction with the QBFT consensus model in a timeout-based simulation. In each consensus round, nodes were assigned random timeout values drawn from one of four probability distributions. These timeouts modeled network variability, capturing how nodes may respond at different speeds. Nodes with shorter timeouts were, therefore, more likely to be selected as leader candidates.

The simulations followed three distinct scenarios. In the first, the node with the minimum timeout was automatically selected as the round leader. This provided a baseline where leader selection depended solely on timeout values, without QBFT's voting mechanism. Furthermore, in order to evaluate the influence of QBFT's voting mechanism in combination with node timeouts, two additional scenarios were considered, where candidates were selected from among pools of the 20% or 30% of nodes with the shortest timeouts. In both cases, the candidate nodes voted for themselves, while the remaining nodes cast their votes randomly among the candidates. In order to be elected as the round leader, a candidate had to secure both a majority vote and also meet the quorum threshold. If these conditions were not met, the voting process was repeated until a leader was selected.

All distributions were evaluated across five different t_f values, with results representing averages across these. In addition, the uniform, normal, and lognormal distributions were evaluated under different standard deviations ($\frac{1}{2}t_f$, $\frac{1}{4}t_f$, $\frac{1}{8}t_f$). The Weibull distribution, by contrast, was assessed at a fixed standard deviation determined by its shape parameter $k = 0.5$. Although this limited direct comparison, the averaging of results across standard deviations preserved overall comparability.

The key finding is that the uniform distribution did not consistently promote leadership uniformity. While the uniform, normal, and lognormal distributions exhibited comparable performance with average MAEs around 3.3 and MSEs between 18.0 and 18.8, the Weibull distribution achieved a significantly lower average MAE of 2.7 and MSE of 11.1 (Table 14). This highlights its strong performance and indicates that the Weibull distribution, with $k = 0.5$, supports a fairer distribution of leadership roles compared to the other models. The Weibull distribution's heavy-tailed shape with $k < 1$ effectively models real-world network delays, where frequent fast responses are interspersed with occasional high-latency spikes. The fact that the Weibull distribution both reflects real-world delay patterns and outperforms other models in fairness metrics underscores its practical relevance as a model for timeout-driven leader selection in QBFT.

Further examining the results, under minimum-timeout selection, the lognormal and Weibull distributions had the lowest MAEs (3.3), with MSEs of 18.6 and 14.8, respectively (Table 6). However, quorum-based selection from the 20% or 30% fastest nodes led to even lower error rates. The best overall performance was observed with the Weibull distribution in the 20% scenario (MAE = 2.0, MSE = 6.5) (Table 9). Similarly, in the 30% selection scenario, the Weibull distribution recorded the lowest error, with an MAE of 2.6 and an MSE of 12.0. Meanwhile, the other distributions showed very similar outcomes, with MAEs ranging from 3.1 to 3.2 and MSEs between 15.5 and 17.0 (Table 12).

Across all distributions, quorum-based strategies reduced error: The minimum-timeout method yielded an overall MAE of 3.4 and an MSE of 19.6, while the 20% and 30% shortest-timeout strategies each achieved an MAE of 3.0, with the 20% method achieving a lower MSE (14.5), compared to 15.3 for the 30% method (Table 13). Although selecting the node with the lowest timeout improves per-round responsiveness, it may lead to uneven leadership distribution over time, particularly under low-variability timeout distributions where the same nodes are more likely to be repeatedly selected by chance. In contrast, incorporating controlled randomness through quorum-based voting fosters a more balanced

distribution by reducing the statistical likelihood of repeated selection of the same nodes. These findings suggest that timeout-based quorum voting not only lowers cumulative error rates but also supports more adaptive leader distribution in QBFT. By leveraging latency variability, especially in scenarios with heavy-tailed delays, this approach shows promise for enhancing fairness in timeout-driven BFT consensus schemes.

The choice of quorum size, typically set at $\lceil \frac{2n}{3} \rceil$, also played a significant role. While lower candidate selection thresholds (20% and 30%) improved fairness and reduced error rates, preliminary testing showed that higher thresholds (40% and 50%) led to election deadlocks. More specifically, self-voting candidates combined with randomized voting from the remaining nodes caused excessive vote fragmentation, making it difficult for any candidate to reach quorum. For example, in a 10-node network, if four candidates each voted for themselves and the remaining six cast votes randomly, quorum was rarely achieved, leading to stalled leader elections. These simulation results highlight the importance of carefully designing quorum mechanisms, particularly in smaller networks, to maintain liveness and ensure efficient leader rotation.

Overall, the results show that leadership uniformity in timeout-based QBFT depends on both the timeout distribution and the leader selection strategy. Furthermore, incorporating timeouts into the voting process improves fairness by promoting a more balanced distribution of leadership over time. This highlights timeout-aware selection as a practical alternative to static rotation, particularly in permissioned networks with heterogeneous validator performance, and demonstrates how timeout-driven randomness can support more adaptive QBFT designs.

This study has several limitations: First, extreme conditions such as infinitely high timeouts (representing failures or severe latency) were not modeled. Future approaches should explore these scenarios and evaluate their impact on consensus fairness. Second, standard deviations for MAE and MSE were not reported, which limits statistical depth. Additionally, the impact of varying network sizes was not examined, narrowing the generalizability of results to larger systems. Only a single shape parameter $k = 0.5$ was analyzed for the Weibull distribution, as it represents a moderately skewed setting that balances early timeout responsiveness with tail stability. However, this choice limits insights into how varying degrees of skewness might affect leader rotation under varying network conditions. Finally, the current model independently resampled timeouts in each round, without persistent node characteristics. Behaviors such as consistent latency patterns or faulty nodes, which could potentially affect leader selection dynamics, were not captured.

Future work could pursue several extensions: One promising direction is to test QBFT performance on larger and more heterogeneous blockchain networks to evaluate scalability. Another direction is to integrate real-time latency feedback into the leader selection logic, together with the verification process, to enable more adaptive behavior under dynamic network conditions. It would also be valuable to compare the proposed timeout-based mechanism with other BFT consensus algorithms using equivalent timeout models to benchmark relative performance and fairness. Additionally, future research could explore more sophisticated voting mechanisms, such as weighted schemes, and, with further modifications, integrate a timeout-based voting mechanism into a real-world QBFT setting. For example, dynamic leader scoring or timeout-based reputation systems could be used to reward consistently responsive nodes, while view-change logic may be optimized using timeout-aware triggers. Another possible direction is to explore formal asymptotic analysis, especially to understand how minimum-timeout selection behaves under different timeout distributions. In summary, this study provides a foundation for exploring more dynamic and equitable leader selection in permissioned BFT systems, and

contributes design insights into timeout distributions, candidate selection thresholds, and quorum strategies tailored to latency-sensitive environments.

6. Conclusions

This study demonstrated that leadership uniformity in QBFT can be influenced by both timeout distributions and quorum-based voting mechanisms. The experimental setup explored three leader selection strategies and evaluated their impact under four distinct timeout distributions: Uniform, normal, lognormal, and Weibull.

The findings showed that the uniform distribution, while often assumed to ensure fairness, does not necessarily produce the most balanced leader rotation. In contrast, the Weibull distribution (with shape $k = 0.5$) consistently achieved lower error rates, with an overall average MAE of 2.7 and MSE of 11.1 across scenarios. While the other distributions were evaluated across multiple standard deviations, the Weibull distribution was assessed only once per mean due to its fixed variance. These results suggest that under network conditions modeled by a Weibull distribution, leader selection can be fairer and more consistent.

The introduction of quorum-based voting, in which leaders are elected from among nodes with the shortest timeouts, further improved fairness between rounds. Both the 20% and 30% strategies outperformed minimum-timeout selection (MAE = 3.4, MSE = 19.6), achieving average MAEs of 3.0. The 20% strategy yielded the lowest average MSE (14.5), followed by the 30% strategy (15.3). In particular, when using the Weibull distribution, the 20% selection strategy achieved the best overall performance (MAE = 2.0, MSE = 6.5), with the 30% variant also showing strong results (MAE = 2.6, MSE = 12.0). While minimum-timeout selection improves responsiveness in individual rounds, it does not necessarily promote long-term fairness, as it may lead to repeated selection of the same nodes under low-variability timeout distributions. In contrast, quorum-based selection introduces controlled randomness, which helps mitigate this effect and fosters a more balanced distribution of leadership over time.

Overall, the results suggest that timeout-based dynamics offer a promising direction to improve fairness in BFT consensus protocols. By leveraging latency variability, especially under heavy-tailed delay distributions, timeout-based quorum leader selection demonstrates improved fairness in leader rotation and offers a practical path to more equitable timeout-based BFT protocols. By modifying leader selection to incorporate both distribution-based timeouts and structured voting, the consensus process can become more balanced, less susceptible to selection bias, and better suited to dynamic environments. These simulation-backed findings highlight how timeout-driven randomness, shaped by distribution properties and quorum design, can inform the development of adaptive and equitable QBFT variants for permissioned, latency-sensitive networks.

While the results offer meaningful insights, the study was conducted in a controlled environment with fixed network size and a single Weibull shape parameter ($k = 0.5$). Furthermore, it did not model failure scenarios or report statistical variance across trials, limiting broader generalizability. Future research could explore a broader range of timeout conditions, incorporate real-time latency feedback, compare performance against other BFT algorithms, and assess scalability in larger or more heterogeneous networks.

Author Contributions: Conceptualization: E.I.; Methodology: A.P.D. and E.I.; Literature research: A.P.D.; Writing—original draft preparation: A.P.D. (main author), S.P., and E.I.; Review: E.I., G.G., S.P. and A.P.D.; Writing—final review and editing: E.I.; Supervision of article development: E.I.; Funding supervision: G.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially funded by the European Union—DIGI4ECO project, grant agreement ID: 101112883.

Data Availability Statement: Data are available upon request from the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BFT	Byzantine Fault Tolerance
BGP	Byzantine Generals Problem
CAP	Consistency, Availability, Partition Tolerance
DAG	Directed Acyclic Graph
EVT	Extreme Value Theory
HQ	Hybrid Quorum
IBFT	Istanbul Byzantine Fault Tolerance
IBFT 2.0	Istanbul Byzantine Fault Tolerance version 2.0
IoT	Internet of Things
MAE	Mean Absolute Error
MSE	Mean Squared Error
P2P	Peer-to-Peer
PBFT	Practical Byzantine Fault Tolerance
PoA	Proof-of-Authority
QBFT	Quorum Byzantine Fault Tolerance
Q/U	Query/Update

References

1. Lashkari, B.; Musilek, P. A Comprehensive Review of Blockchain Consensus Mechanisms. *IEEE Access* **2021**, *9*, 43620–43652. [CrossRef]
2. Delladetsimas, A.P.; Papangelou, S.; Iosif, E.; Giaglis, G. Integrating Blockchains with the IoT: A Review of Architectures and Marine Use Cases. *Computers* **2024**, *13*, 329. [CrossRef]
3. Prove AI. The State of Enterprise Blockchain Adoption in 2023. Technical Report, Prove AI, 2023. Prove AI Was Formerly Known as Casper Labs. Available online: <https://proveai.com/news/casper-labs-unveils-2023-enterprise-blockchain-report#:~:text=87%25%20are%20likely%20to%20invest,even%20amid%20an%20expected%20downturn> (accessed on 12 February 2025).
4. Dehghani, M.; Kennedy, R.W.; Mashatan, A.; Rese, A.; Karavidas, D. High Interest, Low Adoption: A Mixed-Method Investigation into the Factors Influencing Organisational Adoption of Blockchain Technology. *J. Bus. Res.* **2022**, *149*, 393–411. [CrossRef]
5. AlShamsi, M.; Al-Emran, M.; Shaalan, K. A Systematic Review on Blockchain Adoption. *Appl. Sci.* **2022**, *12*, 4245. [CrossRef]
6. Fan, C.; Lin, C.; Khazaei, H.; Musilek, P. Performance Analysis of Hyperledger Besu in Private Blockchain. In Proceedings of the 2022 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS), Newark, CA, USA, 15–18 August 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 64–73. [CrossRef]
7. Gerrits, L.; Samuel, C.N.; Kromes, R.; Verdier, F.; Glock, S.; Guitton-Ouhamou, P. Experimental Scalability Study of Consortium Blockchains with BFT Consensus for IoT Automotive Use Case. In Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems, Coimbra, Portugal, 15–17 November 2021; ACM: New York, NY, USA, 2021; pp. 492–498. [CrossRef]
8. Catarino, J.P.F. Scalable Blockchain Consensus for Peer-To-Peer Energy Trading: QBFT vs. Clique. Master's Thesis, Universidade NOVA de Lisboa, Lisboa, Portugal, 2024.
9. GoQuorum Documentation Team. Configure QBFT Consensus Protocol in GoQuorum. 2024. Available online: <https://docs.goquorum.consensus.io/configure-and-manage/configure/consensus-protocols/qbft> (accessed on 30 January 2025).
10. Hyperledger Besu Documentation Team. Configure QBFT Consensus Protocol in Hyperledger Besu. 2025. Available online: <https://besu.hyperledger.org/private-networks/how-to/configure/consensus/cliq> (accessed on 30 January 2025).
11. Iosif, E.; Christodoulou, K.; Touloupou, M.; Inglezakis, A. Leadership Uniformity in Raft Consensus Algorithm. In Proceedings of the Information Systems: 17th European, Mediterranean, and Middle Eastern Conference, EMCIS 2020, Dubai, United Arab Emirates, 25–26 November 2020; Proceedings 17; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 125–136. [CrossRef]

12. Ongaro, D.; Ousterhout, J. In search of an understandable consensus algorithm. In Proceedings of the 2014 USENIX Annual Technical Conference (USENIX ATC 14), Philadelphia, PA, USA, 19–20 June 2014; USENIX Association: Berkeley, CA, USA, 2014; pp. 305–319.
13. Versluis, C.; Straetmans, S. Skewness Measures for the Weibull Distribution. 2015. Available online: <https://ssrn.com/abstract=2590356> (accessed on 8 July 2025).
14. Papagiannaki, K.; Moon, S.; Fraleigh, C.; Thottan, M.; Iannaccone, G.; Diot, C. Measurement and Analysis of Single-Hop Delay on an IP Backbone Network. *IEEE J. Sel. Areas Commun.* **2003**, *21*, 908–921. [CrossRef]
15. Hernández, O.F.; Phillips, W. Weibull Mixture Model to Characterise End-to-End Internet Delay. *IEE Proc.-Commun.* **2006**, *153*, 295–301. [CrossRef]
16. Lamport, L.; Shostak, R.; Pease, M. The Byzantine Generals Problem. In *Concurrency: The Works of Leslie Lamport*; ACM: New York, NY, USA, 2019; pp. 203–226. [CrossRef]
17. Castro, M.; Liskov, B. Practical Byzantine Fault Tolerance. In Proceedings of the OSDI, New Orleans, LA, USA, 22–25 February 1999; USENIX Association: Berkeley, CA, USA, 1999; Volume 99, pp. 173–186.
18. Zhong, W.; Yang, C.; Liang, W.; Cai, J.; Chen, L.; Liao, J.; Xiong, N. Byzantine Fault-Tolerant Consensus Algorithms: A Survey. *Electronics* **2023**, *12*, 3801. [CrossRef]
19. Wong, D.; Kolegov, D.; Mikushin, I. Beyond the Whitepaper: Where BFT Consensus Protocols Meet Reality. (Preprint) *Cryptology ePrint Archive*; 2024. Available online: <https://eprint.iacr.org/2024/1242> (accessed on 30 January 2025).
20. Cowling, J.; Myers, D.; Liskov, B.; Rodrigues, R.; Shrira, L. HQ Replication: A Hybrid Quorum Protocol for Byzantine Fault Tolerance. In Proceedings of the 7th Symposium on Operating Systems Design and Implementation (OSDI), Seattle, WA, USA, 6–8 November 2006; USENIX Association: Berkeley, CA, USA, 2006; pp. 177–190.
21. Abd-El-Malek, M.; Ganger, G.R.; Goodson, G.R.; Reiter, M.K.; Wylie, J.J. *Correctness of the Read/Conditional-Write and Query/Update Protocols*; Technical Report CMU-PDL-05-107; Carnegie Mellon University, Parallel Data Laboratory: Pittsburgh, PA, USA, 2005.
22. AMIS Technology. EIP-650: Istanbul Byzantine Fault Tolerance. *GitHub*; 2017. Available online: <https://github.com/ethereum/EIPs/issues/650> (accessed on 18 February 2025).
23. Moniz, H. The Istanbul BFT Consensus Algorithm. *arXiv* **2020**, arXiv:2002.03613. [CrossRef]
24. Saltini, R.; Hyland-Wood, D. IBFT 2.0: A Safe and Live Variation of the IBFT Blockchain Consensus Protocol for Eventually Synchronous Networks. *arXiv* **2019**, arXiv:1909.10194. [CrossRef]
25. Quorum, C. Istanbul BFT (IBFT) Consensus Algorithm. 2023. Available online: <https://goquorum.readthedocs.io/Consensus/ibft/ibft/> (accessed on 18 February 2025).
26. Enterprise Ethereum Alliance. QBFT Consensus Protocol Specification v1. 2023. Available online: <https://entethalliance.org/specs/qbft/v1/> (accessed on 18 February 2025).
27. Hyperledger Besu Documentation Team. *Clique Consensus Protocol—Hyperledger Besu Documentation*; Consensus: Fort Worth, TX, USA, 2023. Available online: <https://besu.hyperledger.org/private-networks/how-to/configure/consensus/clique> (accessed on 18 February 2025).
28. Lopes, J. Exploring Consensus Algorithms in Quorum. Master’s Thesis, Instituto Politécnico do Porto, Porto, Portugal, 2023.
29. Ahmed, I.; Turki, M.; Baklouti, M.; Dammak, B.; Alshahrani, A. Towards an Optimized Blockchain-Based Secure Medical Prescription-Management System. *Future Internet* **2024**, *16*, 243. [CrossRef]
30. Pichaibunditkun, T.; Joseph, F.J.J. Private Permission Blockchain for Optimized Invoice Management System. In Proceedings of the 2023 8th International Conference on Business and Industrial Research (ICBIR), Bangkok, Thailand, 18–19 May 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1318–1322. [CrossRef]
31. European Union. Digi4Eco: Digital Solutions for Sustainable and Resilient Environments. 2024. Available online: <https://digi4eco.eu> (accessed on 18 February 2025).
32. Embrechts, P.; Klüppelberg, C.; Mikosch, T. *Modelling Extremal Events: For Insurance and Finance*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013; Volume 33.
33. Zwart, B.A.P. Queueing systems with heavy tails. *Perform. Eval.* **2002**, *49*, 301–309.
34. Smith, R.L. Extreme value theory. In *Handbook of Applicable Mathematics*; Wiley: Hoboken, NJ, USA, 1990; Volume 7, pp. 437–471.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.