Journal of Metaverse
Research Article
Received: 2025-01-20 | **Reviewing:** 2025-01-27 & 2025-05-19 | **Accepted:** 2025-05-21 | **Online:** 2025-06-04 | **Issue Date:** 2025-12-31
Year:2025, **Volume:**5, **Issue:**2, **Pages:** 132-144, **Doi:** 10.57019/jmv.1623586

RBFT: Resilience-Oriented Blockchain Consensus Protocol

Oumaima FADI
International University of Rabat
Rabat, Morocco
oumaima.fadi@uir.ac.ma
0000-0003-0666-517X
(Corresponding Author)

Karim ZKIK
RENNES School of Business
Rennes CEDEX, France
karim.zkik@rennes-sb.com
0000-0002-8485-8455

Adil BAHAJ
COLCOM, University Mohammed VI Polytechnic
Rabat, Morocco
adil.bahaj@um6p.ac.ma
0000-0001-7842-8726

Abdellatif EL GHAZI
International University of Rabat
Rabat, Morocco
abdellatif.elghazi@uir.ac.ma
0000-0003-1990-4993

Mohammed BOULMALF
International University of Rabat
Rabat, Morocco
mohammed.boulmalf@uir.ac.ma
0009-0002-9760-0215

***Abstract*—Blockchain resilience is the capacity of a blockchain system to proactively adapt to and recover from disruptions. A resilient blockchain remains operational and effective even in the face of unexpected challenges such as security breaches, system failures, or other unforeseen events. Moreover, consensus protocols play a fundamental role in blockchain networks by providing transparency, traceability, and trust, thereby addressing fundamental system weaknesses. However, before adopting these protocols in real-world applications, it is crucial to evaluate their specific features and how they influence the resilience of blockchain. In this study, a new consensus protocol that is resilience-oriented is developed. To achieve this, a comprehensive analysis of existing consensus protocols is conducted, focusing on identifying key metrics essential for evaluating blockchain resilience and ensuring long-term sustainability. The proposed RBFT protocol has demonstrated enhanced resilience within the blockchain network, primarily due to three key mechanisms: a weak coordinator model, weighted validation, and tolerance for late nodes. Furthermore, RBFT outperformed existing consensus protocols in both latency and throughput, showcasing its robustness against increasing numbers of faulty nodes and confirming its scalability under growing network demands. This research aims to assist managers and organizations in adopting blockchain technology, particularly by integrating suitable consensus protocols, improving the resilience of blockchain, as well as its adoptive networks.**

Keywords—Blockchain, Resilience, Consensus protocols

I. INTRODUCTION

Blockchain technology has attracted growing interest from both academia and industry due to its decentralized architecture, cryptographic security, and capabilities for ensuring data integrity, traceability, and transparency. These characteristics make blockchain a promising solution for building resilient digital infrastructures that can operate reliably even under adverse or unpredictable conditions. Among its various applications, the supply chain domain stands out as a particularly relevant use case, where disruptions, such as natural disasters, human error, or cyber-attacks, can severely impact performance and continuity [1].

In this context, blockchain is increasingly seen as a tool to support supply chain resilience by enabling secure, transparent, and tamper-evident transaction records across distributed networks. Despite growing literature on the benefits of blockchain in supply chain systems [2-4], empirical research linking blockchain architecture—particularly consensus protocols—to resilience in real-world applications remains limited. This study focuses on blockchain resilience, with particular emphasis on the performance and fault tolerance of its underlying consensus protocols. We present supply chain resilience as a motivating real-world scenario, where the ability to endure and recover from disruptions is essential. Achieving such resilience increasingly relies on blockchain systems that uphold availability, consistency, and security, even under challenging conditions. Specifically, blockchain resilience is largely determined by the resilience of its consensus protocols, which control the network's ability to maintain consistent and reliable operation under adverse conditions.

This paper presents Resilience-based Byzantine Fault Tolerance (RBFT), a novel consensus protocol designed to strengthen the resilience of blockchain systems. In order to provide a comprehensive analysis, we will delve into the resilience of existing consensus protocols, namely AURA (Authority Round), Clique, PBFT (Practical Byzantine Fault Tolerance), and IBFT (Istanbul Byzantine Fault Tolerance), through the lens of the CAP (Consistency- Availability- Partial Fault Tolerance) theorem. Subsequently, we will identify and examine key metrics specific to blockchain-based systems, which play a vital role in evaluating the resilience of these protocols, namely latency, finality, fork management, message complexity, and byzantine fault tolerance. The RBFT protocol operates through three distinct phases: proposal, validation, and decision. It emphasizes accommodating slow nodes and implementing weighted validation. By balancing the requirements of supply chain environments with security and resilience considerations, our proposed solution achieves better performance than existing protocols in terms of latency and throughput.

The paper is structured as follows: in Section II, we review and analyze related work. In Section III, we define the scope

Cite (APA): Fadi, O., Zkik, K., Bahaj, A., El Ghazi, A., Boulmalf, M. (2025). RBFT: Resilience-Oriented Blockchain Consensus Protocol. *Journal of Metaverse*. *5*(2), 132-144. **Doi:** 10.57019/jmv.1623586

of our research. Section IV outlines the methodology used, while Section V presents the proposed solution. In Section VI, we provide a theoretical results analysis. Section VII represents simulation results and discussion, and in Section VIII, we highlight both theoretical and practical implications. Finally, we conclude the paper in Section IX.

II. Background

A. Supply Chain Resilience

The supply chain comprises a network of organizations and individuals responsible for moving products, funds, information, and services. Many companies today rely on global networks to conduct their business and share information through interconnected physical devices [5]. However, the interdependence of these entities makes the supply chain increasingly vulnerable to cyberattacks. Cyber-attacks, namely code injection, certificate theft, DOS, and phishing attacks [6], can have far-reaching consequences, as external threats targeting one entity's information systems within the supply chain can spread and compromise other entities' data [7-10]. These risks can have tangible consequences, such as delays in delivery, costly recovery processes, and a decline in service quality. A successful attack on production schedules or supply chains can have a ripple effect throughout the entire global supply chain, resulting in a loss of customer trust, damaged brand reputation, and negative financial implications for the organization [11, 12]. Additionally, the paramount risk lies in the time needed for recovery from a system failure, underscoring the vital significance of supply chain security, which encompasses both preventing attacks and minimizing disruption time.

Supply Chain Resilience (SCR) is the ability of a company to withstand and recover from disruptions while maintaining high performance. It involves adapting and quickly returning to normal or improved operations after failures. Indeed, researchers are continuously exploring strategies to enhance supply chain resilience. The contribution in [13] emphasized hybrid flexibility/redundancy approaches, placing particular emphasis on comprehending supply chain structures. These proposed approaches also involved reducing uncertainty through business process engineering, fostering collaborative partnerships, and integrating operational capabilities to achieve supply chain transparency and control. Furthermore, the authors in [14] stated that investing in cutting-edge technology and data analytics is emphasized to enhance supply chain visibility and responsiveness, allowing for proactive decision-making. Moreover, the contribution highlights the need for periodic risk assessments and the development of contingency plans to be better prepared for unforeseen challenges. However, these approaches are not sufficient to ensure supply chain resilience due to scalability, transparency, and trust issues between supply chain entities, prompting the need to adopt blockchain technology.

Blockchain technology has emerged as a promising solution to address limitations in supply chain networks, offering a decentralized infrastructure that enhances data integrity, traceability, and collaborative transparency. Its core features: immutability, real-time tracking, and secure data sharing—contribute to faster recovery from disruptions, improved risk detection, and reduced information asymmetry [15]. The integration of blockchain with Industry 4.0 technologies has also been shown to enable digitally connected, resilient supply chains capable of withstanding future crises. Recent literature emphasizes blockchain's role in strengthening supply chain resilience (SCR) through key mechanisms such as visibility, collaboration, integration, and risk management. For instance, Bayramova et al. identify these as critical factors during disruption phases when transparency and coordination are vital [16]. In the context of the Saudi construction industry, Azmi et al. highlight blockchain's potential to overcome inefficiencies and foster trust across fragmented supply chain actors—both essential for achieving SCR [17].

Despite these contributions, much of the focus in SCR literature remains on blockchain's structural features (e.g., smart contracts [18, 19], traceability systems), rather than its foun- dational mechanism: the consensus protocol. Yet, as these protocols govern how decentralized networks maintain agreement and validate transactions, they are pivotal to ensuring the consistency, availability, and fault tolerance that resilient supply chains depend on, especially under adversarial or high- load conditions. Therefore, understanding the resilience of blockchain-enabled supply chains necessitates a closer exam- ination of the consensus mechanisms that underpin them.

B. Blockchain Technology For Resilience

Blockchain Technology (BT) operates in a decentralized architecture, allowing for quicker decision-making and improved data transfer efficiency. Its security features are also noteworthy, with blockchain-based networks being able to trace data and ensure that it cannot be tampered with by malicious entities, thereby preserving users' privacy [20].

Moreover, the implementation of blockchain technology in supply chains not only enhances security but also offers several other advantages. These include the reduction of transaction time and costs, assisting sustainability and scalability of the supply chain [21], and notably enhancing visibility across the entire supply chain through readily accessible open ledgers [22].

A comprehensive study of blockchain protocols is essential to fully understand their impact on resilience.

Since any network is susceptible to system failure, it should possess the ability to recover to its initial state. Two types of response mechanisms can manage network recovery: Crash Fault Tolerance (CFT) and Byzantine Fault Tolerance (BFT) mechanisms. CFT mechanisms prevent system failure when nodes crash or go offline. In contrast, BFT mechanisms aim to establish consensus even in the presence of faulty or "Byzantine" nodes that may attempt to disrupt the network by sending conflicting information or withholding information altogether.

TABLE I: CONSENSUS PROTOCOL COMPARISON

Blockchain Consensus Protocol	Key Characteristics	Suitability for Supply Chain Resilience	References
Practical Fault Tolerance (PBFT)	• 33% Byzantine fault-tolerant network. • Operates within 3 phases (Pre-prepare, Prepare, Com- mit). • View change mechanisms in case of unresponsiveness of the node. • High throughput.	• Suitable for private and con- sortium blockchains (SC en- vironment). • Maintains the liveness and safety of the network. • Promotes agility of SC pro- cesses with view-changing mechanisms.	[30] [28] [31]
Proof of Authority (PoA)	• Assigns authority to valida- tor nodes. • Enables rotational mining for fairness. • Adversarial Nodes below 50% for consensus.	• Suitable for private and consortium blockchains (SC environment). • Provides fairness and trust- worthiness	[32] [33]
Aura (Authority Round)	• Implemented in Parity. • Uses a set of authorities for block approval. • Resolved forks	• Provides a fair consensus mechanism but can face challenges when Byzantine nodes are too numerous	[34] [35] [30] [36]
Clique	• Implemented in Geth. • Authorities take turns as leaders in block proposals. • Leader prioritized using the GHOST protocol.	• Allows authority nodes to create blocks but aims for finality through leader prior- itization.	[33] [37] [36]
Istanbul Byzantine Fault Tolerance (IBFT)	• Ethereum standard-based protocol. • Consensus with less than 1/3 Byzantine nodes. • Instant finality	Suitable for private and con- sortium blockchains, ensures instant finality.	[29] [38]

BFT mechanisms align seamlessly with decentralized networks, ensuring system stability and meeting the resilience requirements of blockchain systems [23]. To understand how BFT consensus protocols can enhance resilience, a thorough analysis of their functions and features is necessary.

C. Blockchain consensus protocols

The use of consensus protocols among blockchain nodes establishes peer-to-peer connections, effectively minimizing fraud risks and lowering network costs [24]. Moreover, these consensus protocols encompass rules for nodes to either accept or reject new transactions/blocks, namely Proof of Work (PoW) [25], Proof of Stake (PoS) [26], Practical Byzantine Fault Tolerance (PBFT), and Proof of Authority (PoA) [27].

Given that the supply chain is a relativelt controlled environment with few participants, our assessment focuses on the resilience of private blockchain protocols. PBFT and PoA prove to be well-suited for private and consortium blockchains, primarily because of their minimal complexity and reduced energy consumption [28]. Consequently, this study excludes PoW and PoS, owing to their significant resource and energy demands. [29].

Building on the work in [39], we will focus on the PoA (Proof of Authority) protocol and its variations Aura (Authority Round) and Clique, PBFT (Practical Byzantine Fault Tolerance) and IBFT (Istanbul Byzantine Fault Tolerance) as they are suitable for supply chain environments such as private and consortium blockchains (see table I).

PBFT: The Practical Byzantine Fault Tolerance (PBFT) protocol is based on the Byzantine generals' concept, which states that as long as the number of Byzantine nodes does not exceed 1/3 of the total network, the network will function properly. The PBFT protocol provides fault tolerance, allowing the network to maintain liveness and data safety [30]. Additionally, the PBFT protocol includes a view change mechanism that enables the selection of a new primary node after the block of the previous primary node is successfully added.

POA: To overcome the limitations of both PoW and PoS, the Proof of Authority (PoA) consensus is introduced. As a new type of BFT consensus, PoA assigns authority to certain nodes to validate transactions and blocks. These validator nodes, chosen at random, can broadcast new blocks. By using the PoA mechanism, a rotational mining process is enabled, which ensures the fairness of the network and increases its trustworthiness [32]. In the case of forks, the validators should always append new blocks to the longest chain. To ensure that only honest nodes reach consensus, the number of adversarial nodes must not exceed 50% of the network's nodes. PoA is divided into two types: PoA-based Parity, which introduces Aura (Authority Round), and PoA-based Ethereum Geth, which introduces Istanbul Byzantine Fault Tolerance (IBFT) and Clique [33].

AURA: AURA (Authority Round) is a Proof of Authority (PoA) algorithm implemented in Parity [35]. Two versions of Aura are introduced: the parity-based Aura and the round-based Aura. In the parity-based Aura, a set of authorities, assumed to be honest nodes, are assigned to approve or deny the addition of new blocks to the chain [34]. Consensus is reached when the majority of authorities create new blocks and the first block reaches finality [30]. A leader is considered malicious if it sends different blocks to different authorities, exceeds the maximum number of blocks to create, or fails to send any blocks during its turn. The fork is resolved by voting out the nodes of the forked chain. However, if the number of Byzantine nodes, B, is greater than or equal to N1-N/2 (where

N1 is the number of nodes in the forked chain, and N is the total number of nodes in the network), the fork can't be avoided as there are not enough honest nodes to vote out the forked chain.

Clique: Clique, one of the built-in protocols of PoA, is implemented in Geth (GoLang-based Ethereum). In each step, a set of authorities N is assigned. Each authority takes turns as the leader of every N/2+1 block, and only N-(N/2+1) of the authorities are allowed to propose a block during each step. Unlike Aura, the Clique mechanism allows the authority nodes, including the leader node, to create new blocks, potentially causing forks. However, Clique introduces the GHOST protocol, which prioritizes the leader over other authorities. Upon majority voting on the validated block, the block is considered finalized.

IBFT: Istanbul Byzantine Fault Tolerance is an Ethereum standard-based protocol, inspired by PBFT. It is introduced as a solution for the resource-intensive nature of PoW-based blockchain [29] and is considered a better fit for private and consortium blockchains [38]. IBFT reaches consensus if fewer than 1/3 of the nodes are Byzantine. It operates through 3 steps: in a round change process, the validator broadcasts pre-preparation, preparation, and committing messages to validate the proposed block. However, only validator nodes can propose blocks, and the set of validators is dynamic and can be modified. Additionally, IBFT ensures instant finality by committing instantly validated blocks.

Multiple contributions in the literature discussed the various consensus protocols and their functions. Yet the protocol's impact on system resilience is rarely discussed.

III. Research Scope

Blockchain technology (BT) carries multiple benefits to supply chain (SC) environments. Many works have addressed the SC-BT integration and discussed its benefits [16]. To the best of our knowledge, most of the contributions theoreti- cally discuss the impact of SC-BC integration on resilience. However, there is a lack of focus on practical evaluations or empirical analyses of how resilience is enhanced. We first empirically evaluate resilience using the CAP (Consistency-Availability-Partial Tolerance) theorem. Then, we develop a set of metrics to evaluate the behavior of various consensus protocols and their impact on resilience. Firstly, consistency on the blockchain is achieved when transactions are finalized and when nodes reach consensus [30]. Secondly, the resilience of the blockchain also depends on the availability of data and services shortly after any disruption. In addition, blockchain resilience relies on partial tolerance to ensure service continuity despite malicious nodes. To evaluate the resilience of blockchain, we establish a comparative study of consensus protocols for private blockchain, recalling the consistency, availability, and partial tolerance of each pro- tocol. Furthermore, we propose empirical metrics to evalu- ate blockchain protocol resilience, namely latency, message rounds, protocol complexity, finality status, and fork status. After assessing blockchain resilience using the CAP theorem and identifying evaluation metrics, we introduce RBFT, a new consensus protocol specifically designed to enhance the resilience of blockchain systems. The protocol consists of three key concepts: weak coordinator, weighted validation, and waiting for late nodes via three phases: proposal, validation, and finalization. An in-depth description of the CAP analysis and metrics design will be provided in subsequent sections.

IV. Methodology

A. Framework

The CAP theorem, developed by Eric BREWER, is a widely used tool for evaluating the performance and limitations of distributed systems. It is based on three properties: consistency, availability, and partition tolerance. Consistency refers to all nodes in the system having access to the same data at the same time, resulting in the same output across all network nodes. Availability refers to nodes receiving responses at all times. Partial tolerance is the ability of the system to continue functioning despite the presence of faulty nodes. It is important to note that any distributed system can only guarantee two of the three properties outlined in the CAP theorem. The authors in [40] discussed the importance of partial tolerance for the functioning of a blockchain. Therefore, any decen- tralized system can be adapted to either be AP (Availability- Partial Tolerance) or CP (Consistency-Partial Tolerance) The resilience of the network has gained considerable attention from academia and companies. Network resilience is defined as the ability of the network to ensure the continuity of service despite experiencing disruptive events such as cyber attacks. Many studies focus on identifying appropriate technologies to promote resilience. Our work focuses on the impact of blockchain adoption on resilience. In this paper, we start by analyzing the concept of resilience using the CAP theorem to evaluate different blockchain protocols as illustrated in Figure I.

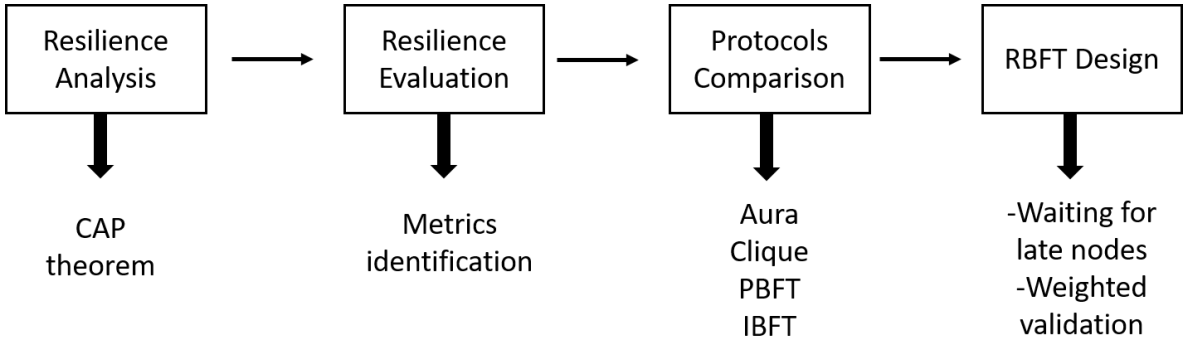

Figure I. Key Steps In RBFT Design

Furthermore, an empirical study is performed by deploying a set of metrics, namely fork management (Fm), latency (L), message complexity (Mc), finality (F), and finally, Byzan- tine fault tolerance (Bn). Our research focuses on private blockchains, which is why we have chosen the most appro- priate protocols (AURA, Clique, PBFT, and IBFT) for private and consortium blockchain networks. Additionally, protocol comparison is conducted using the proposed metrics. We then designed a new resilience-oriented consensus protocol (RBFT) to enhance the blockchain networks' resilience.

B. Resilience from a CAP perspective

In this paper, our goal is to evaluate the resilience of our SC-BC ecosystem based on the CAP theorem. To do so, an analysis of consistency, availability, and partial tolerance is necessary to determine the resilience of the network. The resilience can be assessed from a consistency perspective, specifically, the consistency of an ecosystem reflects its ability to function in an organized and conform manner. The resilience of the network heavily relies on its ability to be prepared for dealing with unexpected events [3]. A resilient network requires consistency during both the pre-attack and post-attack phases.

The work in [41] states that the consistency of a blockchain- based network is increased when the probability of creating forks is decreased exponentially. Additionally, the contribution of [42] states that a blockchain is no longer consistent if an attacker manages to withhold a block for too long, ultimately causing forks. In the work of [41], it is stated that the num- ber of operations reaching consensus from all honest nodes is called convergence opportunities. Therefore, a blockchain network cannot achieve total consistency unless the number of convergence opportunities exceeds the number of faulty nodes that deny all convergence opportunities. In short, the resilience of SC-BC relies on its consistency during both the pre-attack and post-attack phases.

According to the contribution in [3], the post-attack phase resilience includes the network's response to disruptions, as well as its recovery and growth following disruptive events. The continuity of service and the availability of blockchain parties in the face of disruptive events are key factors in the system's resilience. Therefore, we consider availability to be a crucial aspect in interpreting resilience. Additionally, the last component of the CAP theorem is partial tolerance, which is a fundamental characteristic of blockchain-based networks [30]. Consensus protocols are based on Byzantine fault tolerance, ensuring that the system functions despite the occurrence of malicious behavior. As a key component of the CAP theorem, partial tolerance supports resilience during the survival phase (post- attack phase). In summary, blockchain resilience can be com- prehensively analyzed and evaluated through the three pillars of the CAP theorem: consistency, availability, and partial fault tolerance. Each of these elements contributes to the network's resilience in the face of adversarial conditions and its ability to maintain secure, uninterrupted service.

V. Proposed Solution

A. CAP-enabled resilience metrics

Consensus protocols are commonly evaluated in the literature based on latency and throughput. However, to enable a more comprehensive assessment of blockchain resilience, we identify and introduce additional metrics that extend beyond performance. These include Fork Management, Message Complexity, Finality, Latency, and Byzantine Fault Tolerance for theoretical analysis (Section VI), as well as Latency and Throughput for simulation analysis (Section VII). We measure the protocol's latency (L) as the time required to propose, validate, and finalize a block in a consensus-based network. Throughput represents the number of transactions processed per second, serving as a key indicator of performance.

TABLE II: Protocols Performence Comparison

	AURA	**Clique**	**PBFT**	**IBFT**
Latency	2max{E(N/2) + 1}	max{E(N/2) + 1}	3max{E(2N/3) + 1}	3max{E(2N/3) + 1}
Msg Complexity	2	1	3	3
Fork Management	Drop nodes	Resolved (GHOST)	Not allowed	Not allowed
Byzantine Fault Tolerance	Up to 50%	Up to 50%	Up to 33%	Up to 33%
Finality	Majority of nodes	Majority of nodes	After 3 msg rounds	Instant

To evaluate fault tolerance, we introduce the Byzantine Fault Tolerance metric (Bn), which quantifies the maximum number of Byzantine nodes a protocol can withstand while continuing to operate correctly. This metric reflects the protocol's ability to ensure service continuity in the presence of faulty or malicious actors. Ensuring the availability of parties and execution of transactions is another fundamental requirement for blockchain networks. As noted in [30], a blockchain is considered available when proposed transactions are successfully committed i.e., when validated transactions reach finalization and are subsequently executed. To capture this, we define the Finality metric (F), which measures the number of votes needed to finalize a validated block. This metric provides insight into the speed at which transactions are confirmed and the reliability of the network in maintaining continuous service [43].

Another key aspect is message complexity, which reflects the communication overhead and computational demands of the consensus protocol. This is closely related to the sustainability and scalability of the system. To quantify this, we define the Message Complexity metric (M), indicating the number of message rounds required to complete block validation. Together, the metrics L and M are particularly useful for evaluating network resilience, as they measure both liveness and efficiency under normal and disruptive conditions. Finally, fork management is critical for

maintaining network consistency. Forks can result not only from protocol upgrades or disagreements but also from adversarial actions such as selfish mining or eclipse attacks. Effective fork management is therefore essential for ensuring both the availability and consistency of the blockchain. To assess this, we introduce the Fork Management metric (Fm), which indicates whether and how a given consensus protocol handles the occurrence of forks [44].

B. Resilience in Blockchain Protocols: A Comparative Study

In this section, we present a comparative analysis of four prominent consensus protocols: the two main types of Proof-of-Authority (PoA) protocols, namely Aura and Clique, alongside PBFT (Practical Byzantine Fault Tolerance) and IBFT (Istanbul Byzantine Fault Tolerance). The characteristics of these protocols are summarized in Table II. A key parameter for evaluating these protocols is message complexity, which di- rectly influences the protocol's communication overhead and, by extension, its scalability and sustainability. Both PBFT and IBFT employ a three-phase communication process involving a pre-prepare, prepare, and commit phase. This structure provides strong consistency and Byzantine fault tolerance but introduces significant message overhead. In comparison, the Aura protocol reduces this overhead by operating with only two message rounds, thus improving communication efficiency. The Clique protocol further optimizes this by re- quiring just a single message round, which makes it the most efficient among the protocols in terms of message exchange. This characteristic contributes positively to overall network sustainability, particularly in high-load or resource-constrained environments. To assess and model latency, we define li as the latency experienced by node i, representing the time it takes for a block proposed by this node to be propagated through the network and validated by a majority of participants. The overall network latency l reflects the total time required for block finalization under a given consensus protocol. For the purpose of simulation and deeper analysis, we introduce Ek as the ordered set of latencies corresponding to the first k nodes involved in proposing and validating a block. This formulation enables us to compute and compare finalization latency across protocols, offering valuable insight into their responsiveness, efficiency, and resilience under various network conditions.

$$E^k = \{l_k,\ k \in [1, \mathrm{N}]\} \qquad (1)$$

Explicitly, both PBFT and IBFT require a latency of **3 max E[N/2] + 1** to propose and validate a block, as the consensus process involves three distinct phases: pre-preparation, preparation, and commit. Here, *N* denotes the total number of authorities in the network. In comparison, the AURA protocol exhibits a latency of approximately **2 max E[2N/3] + 1**, while Clique achieves improved responsiveness with an estimated latency of **max E[2N/3] + 1**.

Concerning the fork management metric, both PBFT and IBFT are inherently designed to prevent the formation of forked chains by halting consensus when inconsistencies arise. The AURA protocol mitigates forks by rejecting blocks from newly formed branches; however, this approach can compromise network availability due to delayed or discarded proposals. In contrast, Clique adopts the GHOST (Greedy Heaviest-Observed Sub-Tree) rule as an alternative fork resolution strategy, prioritizing blocks proposed by leader authorities over those from non-leader authorities during fork events. Regarding fault tolerance, AURA and Clique can withstand up to **50%** of faulty nodes, whereas PBFT and IBFT maintain their safety guarantees only under the condition that fewer than **33%** of nodes are faulty, in line with classical Byzantine fault tolerance limits. As for block finality, both AURA and Clique reach finality when a majority of nodes vote to finalize a validated block. In contrast, PBFT-based protocols achieve finality after the completion of three message rounds, while IBFT ensures instant finality once a block has been validated and agreed upon by the required quorum.

C. Development of a Resilience-Oriented Blockchain Protocol

In this section, we present our newly designed protocol, in which ensuring the system resilience is its topmost priority. The idea of developing RBFT (Resilience-based Byzantine Fault Tolerance) protocol stems from the requirement of designing a blockchain protocol to improve supply chain resilience.

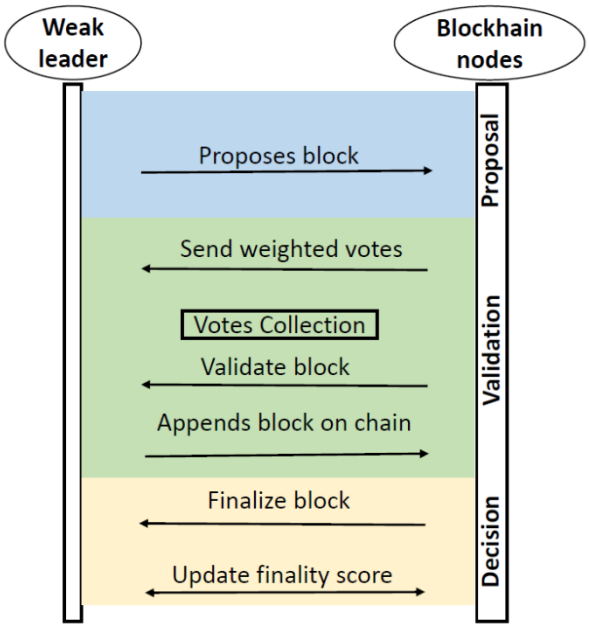

FIGURE II. RBFT PHASES

RBFT functions through three different types of phases as illustrated in Figure II:

- Proposal phase: In this phase, the coordinator is prioritized over regular nodes, proposing a new block to the blockchain.
- Validation phase: During this phase, the proposed block needs to be verified by a majority of the network and hence validated. R-BFT performs weighted validation.
- Decision phase: In this phase, the validated blocks are finalized, thereby the contained transactions are executed.

Being the first blockchain consensus protocol enhancing network resilience, RBFT accommodates delays caused by blockchain attacks such as double spending and selfish mining. A new round is initiated when an authority node proposes a block. After validating the blocks from the current round, the protocol revisits and retrieves any invalidated blocks from the previous two rounds, ensuring this occurs before the timeout period ends. Moreover, RBFT adopts a weighted voting mechanism for block validation. The data used is explained in Algorithm I.

ALGORITHM I. DATA USED IN DIFFERENT ALGORITHMS

Data:
$sealers \subseteq Ids$, the set of sealers;
$c_i =< B_i, P_i >$, the local blockchain at node p_i is a directed acyclic graph of blocks B_i and pointers P_i;
n: number of nodes in the network i.e.$n = |N|$;
b: a block record with field;
$parent$: the block preceding block b in the chain, initially $\perp$;
t: number of byzantine nodes (assuming that $t < n/3$);
r_j : round j;
$Timeout$: the time for a round to be complete;
$coord_i$: the weak leader node i;
p_i : node i;
v_i: the value proposed by p_i;
w_i : the value proposed by coordinator node;
$x_i(p_i)$: the score of p_i ;
$x_i\ (p_i) = 1$;
$vote_j$: the number of votes of an agreement to append the value proposed in r_j,$vote_j = 0$;
aux_i: The message broadcasted by p_i;
$Proposals_i$: list of proposed values from node p_i (values are the hash of the bloc b_i proposed by p_i);
$validated_i$: list of values sent from nodes of the network reaching agreement from the majority of the nodes;
$decided_i$: list of the decided values (finalized blocks) after the validation;
$count_i$: The count of how many time node i has proposed a block and it got finalized.

Proposal phase

Algorithm II presents the proposal phase, which gives priority to a dynamic set of authority nodes over the regular nodes. This concept refers to the decentralized distribution of authoritative responsibilities among multiple nodes in the network, rather than relying on a single authority to maintain the network's operations. The authority set of nodes changing periodically enhances network flexibility and reduces the risk of service disruptions, whether due to an authority node's failure, crash, or attack. The rotation of authoritative responsibilities among nodes allows for contin- uous network functionality, even in the presence of adver- sarial elements. This design helps ensure the robustness and resilience of the network, promoting stability and reliability in its operations. Furthermore, the authority nodes broadcast the proposed blocks to the entire network. Once a round is completed, the role of the proposing node shifts to a different authority node, which then broadcasts a new proposed value to be added to the proposals list.

ALGORITHM II. PROPOSAL PHASE

Procedure `propose()`$_i$
  $r_j = j; j = 0$;
  while *true* **do**
    $r_j = r_j + 1$;
    if $coord_i$ *proposes* **then**
      $p_i = coord_i$;
      $aux_i = Broadcast[r_j](w_i)$;
    end
    else if p_i *proposes* **then**
      $p_i = coord_i$;
      $aux_i = Broadcast[r_j](v_i)$;
    end
    proposals.append(aux_i)
  end

Validation phase

As presented in Algorithm III, R-BFT uses a weighted validation approach. By definition, a block must be validated by a majority of nodes to be appended to the chain. However, disruptions can delay the collection of majority votes, thus hindering the validation process. To mitigate this issue, RBFT assigns two types of votes to the authority nodes (voters). Specifically, some nodes have a higher vote value than others.

ALGORITHM III. VALIDATION PHASE

Procedure `validate()`$_i$
  Wait until ($\exists i$,proposals[i]$\neq \perp$ and timeout expired);
  while $\exists i$,*proposals[i]*$\neq \perp$ *and timeout expired* **do**
    for *i in range (1,n)* **do**
      if $x(p_i) = 2$ *and* p_i *agrees with* aux_i **then**
        $vote_j = vote_j + 2$;
      end
      else if $x(p_i) = 1$ *and* p_i *agrees with* aux_i **then**
        $vote_j = vote_j + 1$;
      end
      if $vote_j > \lfloor \sqrt{n-t} \rfloor$ **then**
        validated.append(aux_i) ;
      end
      wait until $\frac{(n-t)}{2} + 1$ propose a value(bloc) and $\frac{(n-t)}{2} + 1$ agree with aux_i;
      decide(aux_i);
      if aux_i*decided in round* r_j *and timeout not expired* **then**
        Wait until ($vote_{j-1} > \lfloor \sqrt{n-t} \rfloor$ and $vote_{j-2} > \lfloor \sqrt{n-t} \rfloor$);
      end
      else if *decided in* r_{j-1} *and* r_{j-2} **then**
        halt;
      end
    end
  end
  $j = j + 1$

Each node's vote value is determined by the number of blocks it has finalized during previous rounds determines each node's vote value. In this way, nodes that finalize a greater number of blocks gain eligibility for a higher voting weight, thereby accelerating the validation process. In order to ensure fast finalization without compromising security, R-BFT does not require a full majority of all nodes to validate a block. Instead, it defines a dynamic threshold based on the expression

$\sqrt{n} - t$, where n denotes the total number of participating nodes and t represents the maximum number of Byzantine (faulty) nodes. The intuition is that since t<n/3, there are at least (n − t) honest nodes in the network. Rather than requiring confirmation from all honest nodes, RBFT selects a threshold proportional to the square root of the number of honest nodes, which balances reliability and latency.

Formally, the validation condition can be written as: the block is validated if and only if votej > $\sqrt{(n - t)}$, where votej denotes the number of votes on a proposed block. This square root threshold ensures that the number of required votes grows sublinearly with the network size, thereby improving scalability. At the same time, it preserves security because the threshold remains significantly higher than the possible cumulative weight of Byzantine nodes, making it unlikely for faulty participants to compromise the system. Moreover, by not requiring participation from all honest nodes, the protocol tolerates delays or failures from a subset of them while still guaranteeing timely block validation. Thus, RBFT achieves a validation latency bounded by the time required to collect $\sqrt{(n - t)}$ weighted votes.

Decision phase

During the decision phase, the validated blocks are committed, leading to the execution of the block's transactions. To maintain a fairly distributed network, finality is reached only once the majority of the network has verified and confirmed the validated block. In essence, a node's vote depends on how many blocks it has finalized, the more finalized blocks a node has in previous rounds, the higher its finality voting score. To avoid outliers problems, we added meank(counti) - stdi(counti). This integration of the standard deviation allows RBFT to prevent nodes with abnormally high finalized block counts; often old nodes whose past performance may no longer reflect their current reliability from dominating the voting process. By correcting the threshold based on both the average and the dispersion of counts, the protocol ensures that only active and consistently reliable nodes contribute with high voting scores during validation.

Algorithm IV represents the block finality process. The goal is to assign finality voting scores to the nodes using $mean_k(count_i)$ as the threshold over which we prioritize the nodes with higher $count_i$.

ALGORITHM IV. DECISION PHASE

Procedure decide()$_i$
  decided.append(aux_i);
  $count_i = count_i + 1$;
  if $count_i >= mean_k(count_i) - std_i(count_i)$ **then**
    $x(p_i) = 2$;
  end
  else
    $x(p_i) = 1$;
  end

We provide a summary of RBFT of latency estimation in Figure IV. In the following section, simulations of protocols are conducted to measure the latency and throughput of the following protocols: AURA, Clique, IBFT, PBFT, and our proposed protocol, RBFT.

VI. THEORETICAL RESULTS ANALYSIS

While blockchain resilience has been extensively discussed in theoretical terms, there remains a significant need for empirical investigation to validate and refine these concepts. In this work, we argue that blockchain resilience fundamentally depends on the robustness of its underlying consensus protocol. Specifically, we examine resilience through the lens of the CAP theorem, emphasizing three core properties: consistency, availability, and partial tolerance.

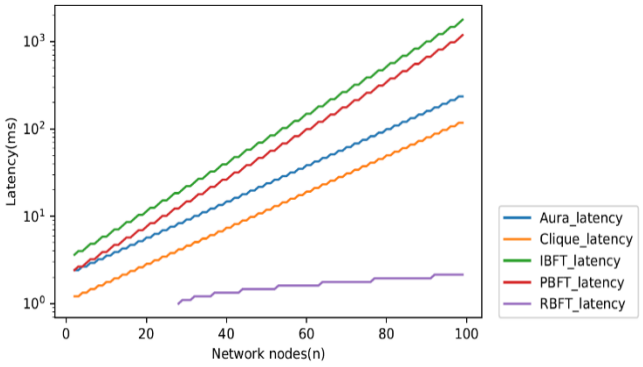

FIGURE III. LATENCY COMPARISON: RBFT VS OTHER PROTOCOLS

Consistency ensures that all honest nodes in the network agree on the same state, even in the presence of faults. According to Kiffer et al. [41], resilience is preserved when the number of operations accepted by honest nodes outweighs attempts by faulty nodes to disrupt consensus. Therefore, a blockchain system's ability to maintain consistency before and after an attack is a crucial aspect of its resilience.

Moreover, availability refers to the system's capacity to provide uninterrupted service, even during adverse conditions. As noted by Dubey et al. [3], post-attack resilience includes the network's ability to respond to disruptions and to recover and continue growing afterward. Ensuring that transactions are processed and that network participants remain reachable during such disruptions is essential to sustaining trust and functionality. Furthermore, partial tolerance, often realized through Byzantine Fault Tolerance (BFT), allows the system to function correctly even when a portion of nodes behave maliciously or fail. This property is especially important during the survival phase, immediately following a disruptive event, as it ensures that the consensus mechanism remains operational and reliable. Resilience in blockchain networks has received limited attention in empirical studies. To address this gap, we propose a set of metrics specifically designed to evaluate the resilience of existing consensus protocols. Based on insights from our comparative analysis, we further introduce a novel consensus protocol aimed at enhancing both scalability and resilience in adaptive blockchain environments. Given our focus on the supply chain environment, our study has specifically focused on private and consortium blockchain protocols. More specifically, we have examined Aura, Clique, IBFT, and PBFT, taking into consideration the defined evaluation metrics, namely latency, message complexity, fork

management, fault tolerance, and finality status. Subsequently, we introduced a novel protocol, RBFT, which is developed with an emphasis on resilience and is structured into three phases: proposal, validation, and decision (see Section V). As depicted in Table III and Figure IV, the latency in RBFT is estimated at **max E$\sqrt{(n-t)}$**, noting that *n* is the number of network nodes and *t* is the number of faulty nodes. Furthermore, noting that $t = n/3$, RBFT exhibits tolerance towards malicious and late nodes during disruptive events while maintaining block processing, thereby proving its resilience. RBFT incorporates a late-node waiting strategy, which enhances both consistency and availability by allowing delayed nodes to participate in the consensus process, unlike protocols such as AURA and Clique, which may exclude such nodes, potentially compromising availability. Unlike PBFT and IBFT, which rely on three rounds of communication for block validation, RBFT reaches consensus with just a single message round, significantly reducing communication overhead. Additionally, RBFT ensures that the majority condition is satisfied before a block is finalized, thereby verifying transactions and minimizing the risk of reversal, in contrast to protocols like AURA, which may suffer from temporary forks due to their leader rotation strategy. These features collectively distinguish RBFT as a more time-efficient, resilient, and scalable alternative to existing consensus mechanisms. As illustrated in Figure III, the RBFT protocol demonstrates significantly lower latency compared to existing consensus protocols, making it both time-efficient and scalable. This reduction in latency, particularly under increasing network sizes, highlights RBFT's ability to sustain performance, thereby reinforcing its scalability and resilience.

RBFT Briefing and Mathematical Demonstration

Assumptions:

- RBFT uses weighted voting (some nodes have higher weights).
- Consensus is achieved when votes exceed $\lfloor \sqrt{n-t} \rfloor$.
- Nodes are ranked by the number of finalized blocks.
- Only one message round is required for consensus.
- Latency per node is modeled by a function E.

Latency Function:

Let E_i be the latency for node i. Define:

$$E^{(k)} = \max\{l_1, l_2, \ldots, l_k\}$$

This represents the worst-case delay among the k fastest nodes.

Finalization Threshold:

A block is finalized when the sum of weighted votes reaches (Algorithm 3, Sections V-C2 and V-C3):

$$\lfloor \sqrt{n-t} \rfloor$$

Latency Bound:

Since voting is parallel, the protocol latency satisfies (Algorithm 3, Sections V-C2):

$$\text{Latency} \leq \max\left\{E^{\lfloor \sqrt{n-t} \rfloor}\right\}$$

FIGURE IV. A SUMMARY OF RBFT LATENCY ESTIMATION

TABLE III: RBFT VS OTHER EXISTING PROTOCOLS

	AURA	Clique	PBFT	IBFT	RBFT
Latency	2 max{$E^{N/2+1}$}	max{$E^{N/2+1}$}	3 max{$E^{2N/3+1}$}	3 max{$E^{2N/3+1}$}	max{$E^{\sqrt{n-t}}$}
Msg Complexity	2	1	3	3	1
Fork Management	Drop nodes	Resolved (GHOST)	Not allowed	Not allowed	Resolved
Byzantine Fault Tolerance	Up to 50%	Up to 50%	Up to 33%	Up to 33%	Up to 33%
Finality	Majority of nodes	Majority of nodes	After 3 msg rounds	Instant	Majority of nodes

VII. SIMULATION RESULTS & DISCUSSION

To implement and evaluate the consensus protocols discussed in the previous section (AURA, Clique, IBFT, PBFT, and our proposed RBFT), we utilized an open-source blockchain simulator available at [45]. This simulator provides a modular and extensible framework for modeling consensus mechanisms in permissioned blockchain networks,

allowing the configuration of network parameters such as block size, number of nodes, message delays, and fault tolerance thresh- olds. It enabled us to simulate realistic network conditions and measure key performance indicators, notably latency and throughput, under varying adversarial scenarios, block sizes, and network sizes.

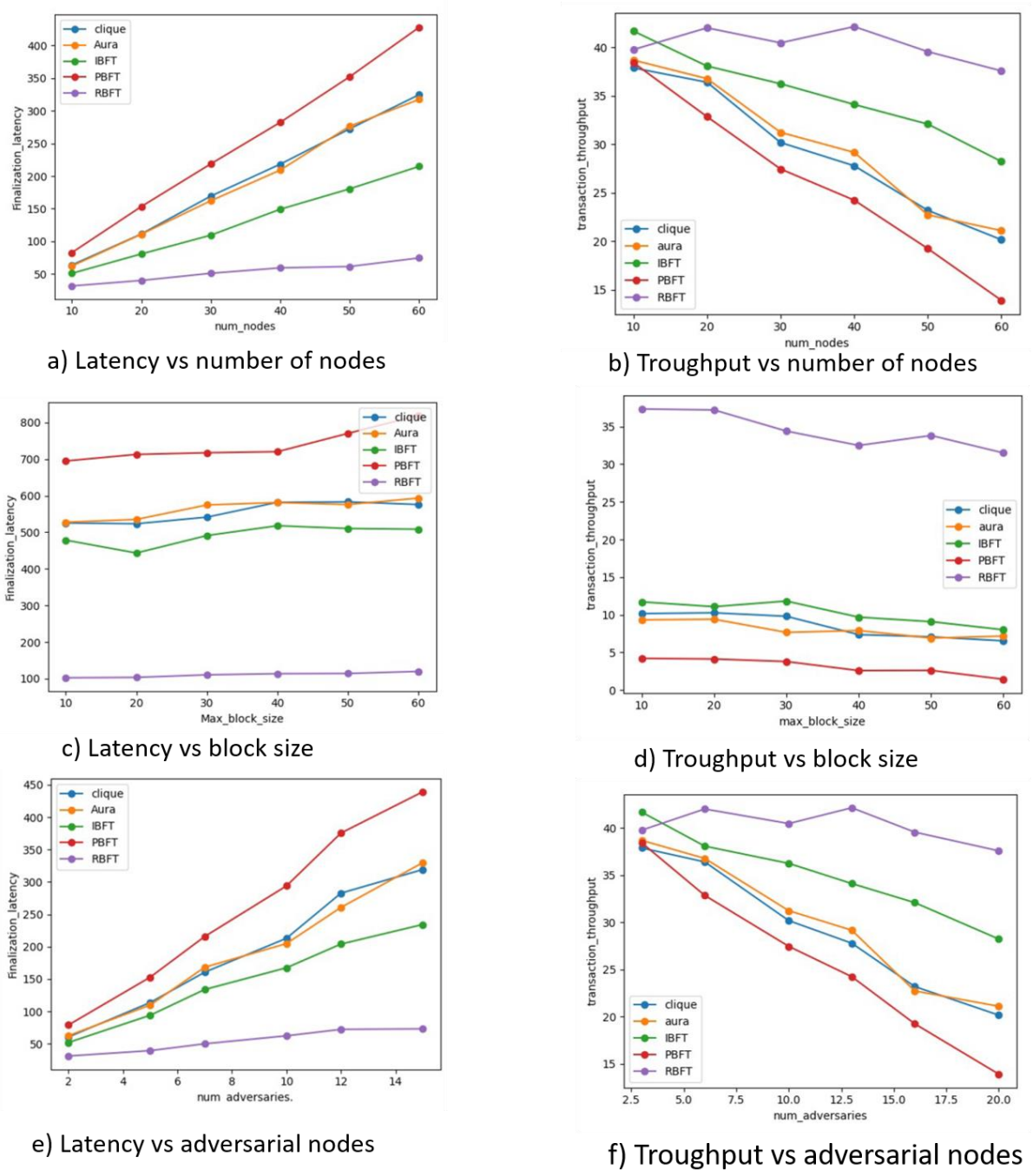

FIGURE V: LATENCY AND TROUGHPUT PERFORMANCE OF PROTOCOLS

In our evaluation, we assessed each consensus protocol based on latency and throughput, alongside key resilience-related characteristics including fault tolerance, message complexity, fork management, and finality (see Section II).

Here, latency refers specifically to the finalization latency, defined as the time from a block's proposal by a node to its validation and finalization on the blockchain. This metric captures the cumulative delay introduced during block proposal, network- wide validation, and finalization. Throughput measures the number of transactions processed per unit of time under each consensus protocol. In our experiments, we gradually increased the number of nodes while keeping the block size fixed at 10 transactions. The results show that PBFT experiences a significant increase in latency as the network grows, indicating limited scalability. AURA and Clique also exhibit noticeable increases in latency with the addition of more nodes. IBFT shows a more moderate latency trend (see Figure V-a). In contrast, our proposed protocol, RBFT, maintains a consistently lower latency across all network sizes, demonstrating its improved scalability and communication efficiency.

When evaluating throughput, RBFT again outperforms the other protocols, maintaining a higher rate of transaction processing as the network scales. In comparison, baseline protocols (PBFT, IBFT, AURA, and Clique) show a decrease in throughput as the number of nodes increases (see Figure V-b). These findings not only highlight RBFT's superior scalability but also suggest its inherent resilience to DoS-like attacks, such as node saturation or network overload. By maintaining low latency and high throughput under increasing system load, RBFT demonstrates its capacity to continue operating effectively even in conditions that mimic resource exhaustion or targeted disruption, which are typical characteristics of a Denial-of-Service attack.

To assess the resilience of each protocol under adversarial conditions, we introduced faulty nodes into the network while keeping the block size fixed at 10 transactions and ensuring that the proportion of adversarial participants remained below one-third of the total nodes. As the number of faulty nodes increased, we observed a marked increase in latency for PBFT, IBFT, AURA, and Clique, indicating their sensitivity to both the presence and density of adversarial behavior. In contrast, RBFT maintained significantly lower latency under the same conditions, demonstrating strong robustness in the presence of Byzantine faults (see Figure V-e).. RBFT also achieves high throughput in adversarial settings, further highlighting its ability to tolerate and perform under malicious interference (see Figure V- f).

RBFT also demonstrates strong resilience under varying block sizes. While increasing block size typically leads to higher latency in conventional protocols, RBFT efficiently manages this overhead, showing only a slight increase in finalization delay (see Figure V-c). Additionally, RBFT maintains high transaction throughput across different block sizes, reflecting its scalability and stable performance (see Figure V-d). These results demonstrate that RBFT can sustain high performance even as block sizes increase. This robustness is especially relevant in the face of transaction flooding attacks, where adversaries attempt to overwhelm the network by injecting large volumes of transactions. RBFT's ability to maintain low latency and high throughput under such conditions highlights its resilience to flooding attacks and confirms its robustness in dynamic and adversarial settings.

These results highlight RBFT's ability to achieve a balance between high transaction throughput and strong resilience against adversarial behavior, making it a promising solution for scalable and secure blockchain deployments. Its performance advantages are rooted in a set of innovative design features, including a pseudo-leader voting mechanism, weighted validation processes, and an adaptive strategy for accommodating delayed or late nodes. This latter feature not only supports network consistency under variable conditions but also mitigates the impact of selfish mining attacks by reducing the effectiveness of block withholding strategies. Together, these elements contribute to RBFT's resilience, enabling it to maintain efficiency and consistency even under challenging network conditions and potential Byzantine threats.

VIII. Theoretical And Practical Implications

This study provides significant insights for both research and business communities, contributing to the advancement of blockchain technology in the context of supply chain resilience [46]. Although the effectiveness of adopting blockchain for supply chain resilience has been theoretically proven, limited attention has been paid to the specific features of blockchain that impact it. This study presents an accurate and targeted approach to adopting blockchain within the supply chain industry, aimed at improving resilience at a more proximate level.

The proposed RBFT protocol, designed to be resilience-oriented, strengthens network resilience and maintains availability even after disruptive events that cause latency. The pro- tocol enables the network to withstand, adapt to, and recover from potential disruptions to achieve high performance. This study provides the research community with a foundational approach to exploring the potential of blockchain in supply chain systems, particularly from a resilience perspective. Fur- thermore, this work encourages researchers and blockchain users to investigate the adaptability of blockchain protocols by defining parameters to evaluate blockchain performance and resilience. In addition, it is intended to initiate the development of new blockchain protocols customized to different operating environments. The potential of blockchain technology to revolutionize sup- ply chain management by providing increased transparency, traceability, efficiency, security, and compliance is significant. This can ultimately help businesses reduce costs, improve customer satisfaction, and gain a competitive advantage. From a business perspective, the proposed solution is designed to assist supply chain partners and business entities in adopting a secure blockchain-based protocol that enhances resilience. The RBFT protocol can help cybersecurity professionals establish trust and confidence in collaborations by offering a practical guide to select the appropriate blockchain model for their different network needs [47]. Furthermore, it ensures system survival after abnormal events in the blockchain context, preventing service degradation that could directly impact an organization's brand and financial performance. Overall, the proposed approach empowers today's blockchain users to fos- ter resilience in their businesses, supporting the transformation of their business models to achieve higher revenue and a value-added environment.

IX. Conclusion

Blockchain resilience refers to the system's ability to with- stand, recover from, and adapt to disruptions or unexpected events, all while maintaining secure and efficient operations. This includes resilience to network failures, malicious be- haviors, and performance degradation. Blockchain technology has shown promise in strengthening the resilience of critical systems, such as supply chains and distributed infrastructure. In this study, we examined blockchain resilience through the lens of the CAP theorem, focusing on the three foundational properties of consistency, availability, and partial (Byzantine) fault tolerance. We conducted a comparative evaluation of widely used consensus protocols: Aura, Clique, PBFT, and IBFT using a

set of blockchain-specific metrics: Latency, Message Complexity, Fork Management, Byzantine Fault Tol- erance, and Finality. This analysis combined both theoretical assessment and simulation-based experimentation to provide a well-rounded evaluation. Based on our findings, we introduced RBFT, a novel consensus protocol explicitly designed to enhance both efficiency and resilience. RBFT incorporates a weighted validation mechanism and a late-node waiting strategy, which together reduce communication overhead and improve inclusivity without sacrificing fault tolerance. Em- pirical results from our simulations demonstrate that RBFT consistently outperforms existing protocols in terms of latency and throughput, across varying network sizes, block sizes, and proportions of Byzantine nodes. These results validate RBFT's potential to support scalable and robust blockchain applications, especially in dynamic and adversarial environ- ments. As future work, we plan to extend this work by exploring additional resilience metrics, simulating real-world attack scenarios to assess RBFT resilience and integrating AI- based anomaly detection to further automate and enhance the consensus validation process.

ACKNOWLEDGEMENT

Not applicable

FUNDING

This research did not receive any outside funding or support. The authors report no involvement in the research by the sponsor that could have influenced the outcome of this work.

AUTHORS` CONTRIBUTIONS

G.M.: Conceptualization, Writing-Original draft preparation, Writing- Reviewing and Editing;

M.M.: Conceptualization, Writing-Original draft preparation, Writing- Reviewing and Editing, Supervision

CONFLICT OF INTEREST

Authors declare that they have no conflict of interest

DATA AVAILABILITY

The data supporting the findings of this study are available upon request from the authors.

ETHICAL STATEMENT

This article followed the principles of scientific research and publication ethics. This study did not involve human or animal subjects and did not require additional ethics committee approval.

DECLARATION OF AI USAGE

No generative AI tools were used for content creation in this manuscript (e.g., drafting, rewriting, or generating ideas).

REFERENCES

[1] Dubey, R., Gunasekaran, A., Childe, S. J., Fosso Wamba, S., Roubaud, D., & Foropon, C. (2021). Empirical investigation of data analytics capability and organizational flexibility as complements to supply chain resilience. *International Journal of Production Research, 59*(1), 110–128.

[2] Moosavi, J., Naeni, L. M., Fathollahi-Fard, A. M., & Fiore, U. (2021). Blockchain in supply chain management: A review, bibliometric, and network analysis. *Environmental Science and Pollution Research*, 1–15.

[3] Dubey, R., Gunasekaran, A., Bryde, D. J., Dwivedi, Y. K., & Papadopoulos, T. (2020). Blockchain technology for enhancing swift-trust, collaboration and resilience within a humanitarian supply chain setting. *International Journal of Production Research, 58*(11), 3381–3398.

[4] Queiroz, M. M., Telles, R., & Bonilla, S. H. (2020). Blockchain and supply chain management integration: A systematic review of the literature. *Supply Chain Management: An International Journal, 25*(2), 241–254.

[5] Tu, M. (2018). An exploratory study of internet of things (IoT) adoption intention in logistics and supply chain management: A mixed research approach. *The International Journal of Logistics Management, 29*(1), 131–151.

[6] Parker, S., Wu, Z., & Christofides, P. D. (2023). Cybersecurity in process control, operations, and supply chain. *Computers & Chemical Engineering*, 108169.

[7] Simon, J., & Omar, A. (2020). Cybersecurity investments in the supply chain: Coordination and a strategic attacker. *European Journal of Operational Research, 282*(1), 161–171.

[8] Gordon, L. A., Loeb, M. P., Lucyshyn, W., & Zhou, L. (2015). The impact of information sharing on cybersecurity underinvestment: A real options perspective. *Journal of Accounting and Public Policy, 34*(5), 509–519.

[9] Huang, C. D., Hu, Q., & Behara, R. S. (2008). An economic analysis of the optimal information security investment in the case of a risk-averse firm. *International Journal of Production Economics, 114*(2), 793–804.

[10] Wu, Y., Feng, G., Wang, N., & Liang, H. (2015). Game of information security investment: Impact of attack types and network vulnerability. *Expert Systems with Applications, 42*(15–16), 6132–6146.

[11] Vanov, D. (2018). Revealing interfaces of supply chain resilience and sustainability: A simulation study. *International Journal of Production Research, 56*(10), 3507–3523.

[12] Ali, I., & Gölgeci, I. (2019). Where is supply chain resilience research heading? A systematic and co-occurrence analysis. *International Journal of Physical Distribution & Logistics Management*.

[13] Ponomarov, S. Y., & Holcomb, M. C. (2009). Understanding the concept of supply chain resilience. *The International Journal of Logistics Management*.

[14] Tukamuhabwa, B. R., Stevenson, M., Busby, J., & Zorzini, M. (2015). Supply chain resilience: Definition, review and theoretical foundations for further study. *International Journal of Production Research, 53*(18), 5592–5623.

[15] Beck, J., Birkel, H., Spieske, A., & Gebhardt, M. (2023). Will the blockchain solve the supply chain resilience challenges? Insights from a systematic literature review. *Computers & Industrial Engineering, 185*, 109623.

[16] Bayramova, A., Edwards, D. J., & Roberts, C. (2021). The role of blockchain technology in augmenting supply chain resilience to cybercrime. *Buildings, 11*(7), 283.

[17] Azmi, N. A., Sweis, G., Sweis, R., & Sammour, F. (2022). Exploring implementation of blockchain for the supply chain resilience and sustainability of the construction industry in Saudi Arabia. *Sustainability, 14*(11), 6427.

[18] Fadi, O., Bahaj, A., Zkik, K., El Ghazi, A., Ghogho, M., & Boulmalf, M. (2025). Smart contract anomaly detection: The contrastive learning paradigm. Computer Networks, 111121.

[19] Zkik, K., Sebbar, A., Fadi, O., Mustapha, O., & Belhadi, A. (2023). A graph neural network approach for detecting smart contract anomalies in collaborative economy platforms based on blockchain technology. In 2023 9th International Conference on Control, Decision and Information Technologies (CoDIT) (pp. 1285–1290). IEEE.

[20] Sengupta, J., Ruj, S., & Bit, S. D. (2020). A comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT. *Journal of Network and Computer Applications, 149*, 102481.

[21] Novo, O. (2018). Blockchain meets IoT: An architecture for scalable access management in IoT. *IEEE Internet of Things Journal, 5*(2), 1184–1195.

[22] Aggarwal, S., Chaudhary, R., Aujla, G. S., Kumar, N., Choo, K.-K. R., & Zomaya, A. Y. (2019). Blockchain for smart communities: Applications, challenges and opportunities. *Journal of Network and Computer Applications, 144*, 13–48.

[23] Berger, C., & Reiser, H. P. (2018). Webbft: Byzantine fault tolerance for resilient interactive web applications. In *Distributed Applications and Interoperable Systems: 18th IFIP WG 6.1 International Conference, DAIS 2018* (pp. 1–17). Springer.

[24] An, A. C., Diem, P. T. X., Van Toi, T., & Binh, L. D. Q. (2019). Building a product origins tracking system based on blockchain and POA consensus protocol. In *2019 International Conference on Advanced Computing and Applications (ACOMP)* (pp. 27–33). IEEE.

[25] Lepore, C., Ceria, M., Visconti, A., Rao, U. P., Shah, K. A., & Zanolini, L. (2020). A survey on blockchain consensus with a performance comparison of PoW, PoS and Pure PoS. *Mathematics, 8*(10), 1782.

[26] Saleh, F. (2021). Blockchain without waste: Proof-of-stake. *The Review of Financial Studies, 34*(3), 1156–1190.

[27] Hjalmarsson, F. P., Hreiðarsson, G. K., Hamdaqa, M., & Hjálmtýsson, G. (2018). Blockchain-based e-voting system. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)* (pp. 983–986). IEEE Computer Society.

[28] Li, W., Feng, C., Zhang, L., Xu, H., Cao, B., & Imran, M. A. (2020). A scalable multi-layer PBFT consensus for blockchain. *IEEE Transactions on Parallel and Distributed Systems, 32*(5), 1146–1160.

[29] Honnavalli, P. B., Cholin, A. S., Pai, A., & Anekal, A. D. (2020). A study on recent trends of consensus algorithms for private blockchain network. In *International Congress on Blockchain and Applications* (pp. 31–41). Springer.

[30] De Angelis, S., Aniello, L., Baldoni, R., Lombardi, F., Margheri, A., & Sassone, V. (2018). PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain. *arXiv preprint arXiv:1901.07160*.

[31] Sukhwani, H., Martínez, J. M., Chang, X., Trivedi, K. S., & Rindos, A. (2017). Performance modeling of PBFT consensus process for permissioned blockchain network (Hyperledger Fabric). In *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)* (pp. 253–255). IEEE.

[32] Xiang, H., Ren, Z., Zhou, Z., Wang, N., & Jin, H. (2020). Alphablock: An evaluation framework for blockchain consensus protocols. arXiv preprint arXiv:2007.13289.

[33] Ekparinya, P., Gramoli, V., & Jourjon, G. (2019). The attack of the clones against proof-of-authority. arXiv preprint arXiv:1902.10244.

[34] Openethereum. (2024). Parity Ethereum. https://github.com/openethereum/parity-ethereum.

[35] Rouhani, S., & Deters, R. (2017). Performance analysis of Ethereum transactions in private blockchain. In 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS) (pp. 70–74). IEEE.

[36] Islam, M. M., Merlec, M. M., & In, H. P. (2022). A comparative analysis of proof-of-authority consensus algorithms: Aura vs Clique. In 2022 IEEE International Conference on Services Computing (SCC) (pp. 327–332). IEEE.

[37] Christyono, B. B. A., Widjaja, M., & Wicaksana, A. (2021). Go-Ethereum for electronic voting system using Clique as proof-of-authority. TELKOMNIKA (Telecommunication Computing Electronics and Control, 19(5), 1565–1572.

[38] Saltini, R., & Hyland-Wood, D. (2019). Correctness analysis of IBFT. arXiv preprint arXiv:1901.07160.

[39] Mylrea, M., & Gourisetti, S. N. G. (2018). Blockchain: Next generation supply chain security for energy infrastructure and NERC critical infrastructure protection (CIP) compliance. In Resilience Week.

[40] Raikwar, M., Gligoroski, D., & Velinov, G. (2020). Trends in development of databases and blockchain. In 2020 Seventh International Conference on Software Defined Systems (SDS) (pp. 177–182). IEEE.

[41] Kiffer, L., Rajaraman, R., & Shelat, A. (2018). A better method to analyze blockchain consistency. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (pp. 729–744).

[42] Longo, R., Podda, A. S., & Saia, R. (2020). Analysis of a consensus protocol for extending consistent subchains on the Bitcoin blockchain. Computation, 8(3), 67.

[43] Cachin, C., & Vukolić, M. (2017). Blockchain consensus protocols in the wild. arXiv preprint arXiv:1707.01873.

[44] Oyinloye, D. P., Teh, J. S., Jamil, N., & Alawida, M. (2021). Blockchain consensus: An overview of alternative protocols. Symmetry, 13(8), 1363.

[45] Applied Protocol Research. (2024). Blockchain simulator. https://github.com/appliedprotocolresearch/blockchain-simulator

[46] Zkik, K., Sebbar, A., Nejjari, N., Lahlou, S., Fadi, O., & Oudani, M. (2023). Secure model for records traceability in airline supply chain based on blockchain and machine learning. In Digital Transformation and Industry 4.0 for Sustainable Supply Chain Performance (pp. 141–159). Springer.

[47] Fadi, O., Lahlou, S., Bahaj, A., Zkik, K., El Ghazi, A., & Boulmalf, M. (2025). An integrated framework for securing IoT networks with blockchain and AI. In Empowering IoT with Big Data Analytics (pp. 199–211). Elsevier.