Review Article

iMedPub Journals
http://www.imedpub.com

American Journal of Computer Science and Information Technology

2020
Vol. 8 No. 4: 62

On the Scalability of a Blockchain Network based on the Practical Byzantine Fault Tolerance Consensus Method

Zeba Mahmood*

Department of Software Engineering, Kaunas Technology University, Kaunas, Lithuania

Corresponding author:
Mahmood Z, Department of Software Engineering, Kaunas Technology University, Kaunas, Lithuania

zeba.mehmood@gmail.com

Citation: Mahmood Z (2020) On the Scalability of a Blockchain Network based on the Practical Byzantine Fault Tolerance Consensus Method Vol.8 No.4: 62.

Abstract

In this work, an investigation is carried out on the architecture of the blockchain networks and the different consensus methods that the network uses to write the transactions on a blockchain. In particular, it focuses on the practical Byzantine Fault Tolerance (pBFT) consensus method. First, a bibliographic and systematic review of the networks using this type of consensus was carried out; then it evaluates the number of nodes and message replicas that a pBFT-based network must have for consensus to be possible. Such a review allow to understand that tolerance to Byzantine faults is associated with the number of nodes that replicate the message that is sent, taking into account that there is a way to calculate the minimum number of nodes that must exist to maintain consensus in these systems. In addition, such tolerance is a property of specific systems, that is, it is not exclusive to blockchain technology. Finally, the main blockchain networks that use the pBFT consensus method have been analyzed in detail, and we have evaluated the conditions under which these networks are working at the time of making this article.

Keywords: Blockchain; Practical byzantine fault tolerance; Distributed system; Consensus; P2P network

Received: October 9, 2020; **Accepted:** October 23, 2020; **Published:** October 30, 2020

Introduction

The blockchain is an innovative topic that has aroused the interest of a large part of the population around the world, given that, with its operation and architecture, it has become a source of inspiration for countless projects that have seen in this technology a potential Great for launching ideas beyond crypto currencies. Being blockchain an emerging digital technology offers a very fertile field for research; one of the most explored main aspects in this area has been decentralization and the methods by which the network reaches consensus [1].

In this work we want to investigate a particular aspect of this technology, we refer to the architecture of the blockchain networks and the different consensus methods that are used by the network to write the blocks of the chain. In particular, we will focus on the practical byzantine fault tolerance method (pBFT). First, we will make a bibliographic and systematic review of the networks that use this type of consensus, and then focus on the number of nodes that a pBFT-based network must have for consensus to be possible. It is often read that blockchain is a disruptive technology [2], so it is worth asking what features it has to make it worthy of this name. It can be said that the birth of this technology has been the conjunction in a single application of years of studies and attempts to present options to introduce changes in aspects such as decentralization or security in network operations that merit the exchange of information between two or more users: processes such as carrying out monetary transactions, handling and recording data, improvements in supply chain issues, including online games or other entertainment.

Although the blockchain became known as a result of Satoshi Nakamoto's publication called "Bitcoin: A Peer-to-Peer Electronic Cash System [3], there are publications made in the 80s and 90s that give the first decentralized architectural ideas historical evolution of the Blockchain. According to [4], Ralph C. Merkle made a publication about the development of Merkle's tree structure. This cryptographic structure uses an ingenious method to store several transactions in a secure and immutable way within the blocks of the chain utilizing nested cryptographic hash functions. The cryptographer David made a publication on secure digital money, which presents an idea of a cryptographic signature applied to electronic money [5]. He proposes the

approach about the creation of an anonymous communication network called DC-Net. Years later, he founded the company DigiCash, where he offers a system based on the ideas proposed through the publication of Ecash.

Haber and Stornetta published an article in, where they propose to work with a chain of cryptographically protected blocks so that nobody can make modifications [6]. A year later, they make modifications to their work to incorporate the Merkle tree and thus be able to add more documents to each block.

Nick Szabo, cryptographer, proposed the idea of creating smart contracts. According to Szabo, given the definition of a contract, which is considered legal agreement between the parties that describes their mutual responsibilities, their idea was to take this definition to the digital domain. A smart contract is the part of the code of a program with computer language, through which any property can be automatically transferred to a specific buyer and the funds to a seller, after an agreement between the interested parties and without the need to an intermediary [7].

In 1997, Adam back proposes the creation of a system that solves the problem of unwanted messages in emails and blogs, for this he developed a system called proof of work (Hashcash), which is currently used to stack a new block of transactions in blockchain networks that use this consensus method. This proposal is collected and improved in the publication [8].

This entire historical journey lays the foundations for the publication of Satoshi Nakamoto, along with the evolution in cryptographic protocols; cryptographic hash functions, which eventually led to the establishment of a completely decentralized network and currency, resistant to malicious attacks and which also overcomes the problem of the double expense.

Methodology

In this article, a review of the literature is carried out to consult, extract and gather relevant information on the topic of consensus on blockchain networks, mainly those based on the problem of the Byzantine generals. As part of this purpose, articles published in the most prominent journals have been consulted, including Scopus, Elsevier, IEEE, including gray literature (non-formal literature), to determine the current state of the subject studied.

The purpose of this review has been to detect, obtain, consult the bibliography and other materials that may be useful to understand the role of the Byzantine general's problem for consensus methods, these being considered a central issue for the evolution of such technology, where relevant and necessary information that concerns the research problem should be extracted and collected.

For the development of this work we have carried out a Multivocal review, which is a type of systematic literature review. Regarding the selection of this methodology, we have based on the ideas, who consider that the traditional methods for systematic reviews of literature are not adapted for the study of current topics from which useful information can be obtained in the form of gray literature, outside academic forums (for example, blog posts, videos and whitepaper) in addition to published literature (formal) [9].

Researchers underline the importance of using Multivocal review since in this way there is a wide range of relevant information, which following a review of systematic literature would not be addressed. In our case, considering the blockchain and everything related to this technology an emerging research topic, the use of this methodology was necessary [10].

In this work in addition, a search was made on the Web to obtain the detail of each one of the blockchain and its respective architectures. For this, first the whitepaper of each of the networks is reviewed for our study, which is commonly hosted on the websites of each project and, in the necessary cases; the portals were consulted external with relevant information [11].

Previous Works and Definitions

Problem of the byzantine generals

The pBFT consensus method is based on the classic problem of the Byzantine generals, which is an inconvenience to which any network of distributed computer systems is vulnerable [12]. This is a classic computing problem known as the Byzantine Generals Problem, which was developed to represent a circumstance where actors must agree on a strategy to avoid catastrophic system failure. And they must also achieve this goal knowing that among them, there may be unreliable or possibly traitorous actors [13].

The problem of the Byzantine Generals was described by Robert Shostak, within the framework of a project of the SRI International Computer Science laboratory [14]. This project was called SIFT and was backed by the same NASA aerospace agency. The case described essentially represents a problem of communication distributed between computers [15].

The historical origin of the name is as follows. The great Eastern Roman Empire, also known as the Byzantine Empire, has decided to capture a city, but there is strong internal resistance. The Byzantine army has surrounded the city, this army has many divisions, and each division has a general. The generals communicate with each other and also among all the lieutenants within their division only through messengers. All generals or commanders must agree on one of the following two action plans, exact coordinate time to attack at once or face fierce resistance, then withdraw immediately. The army cannot resist forever. If all generals and/or messengers are reliable, then the solution is simple. However, some of the messengers and even some generals can be traitors, are spies or enemies. There is a great possibility that they do not follow orders or convey the message wrongly.

The problem of the Byzantine generals applies to a distributed network. In fact, in the blockchain networks, it is even more complicated, since there is no real General. All participants or nodes are the same hierarchy. All participating nodes must

accept each message, which is transmitted between the nodes. If a group of nodes or the message they send is wrong, the network as a whole should not be affected and must withstand this attack [16].

Byzantine fault tolerance

Computer systems in general should be designed to handle components or procedures that may have a malfunction, especially those cases where contradictory information can be generated to the system and generate problems [17].

Byzantine failures can be considered the most disruptive, because they can occur for many reasons and manifest in any way, or even not manifest at all, so, when considering the model of failures that a system will tolerate, The Byzantine failure model is the most severe, blockchain-based systems must tolerate the most severe failure model possible, which is why the consensus-seeking algorithms tolerant to Byzantine failures have aroused increasing interest in recent years [18].

In addition, despite being a problem that is not current in computing, the interest in understanding and knowing it has been increasing today due to the emergence of Bitcoin who became the first known practical solution for this classic problem.

With regard to fault tolerance, it is important to take into account the implicit assertion that there is no relationship between the failures, that is, that the statistical probability that one component fails is independent of the failure in another component of the system [19].

To fulfill this property it is convenient to deploy the replicas on different platforms and locations, which depend on different sources of electricity supply and different network infrastructures and, if possible, using different versions of design and implementation the machines of replicated states as tolerant distributed systems to component failures. This last aspect is especially important in the case of Byzantine fault tolerant systems, since it reduces the probability that a defect or vulnerability in the design or implementation of a particular component compromises the entire system [20].

In addition to the assumed failure model, the fault tolerance of a distributed system is based on the assumptions that are based on its synchrony model and the cryptographic techniques used and applied to the messages. The synchrony refers to the timing aspects of the system in relation to the processing of the messages and their sending through the network, being therefore linked to the nature of the latter. While there are a considerable number of synchronization models, the three most relevant in the context of replicated state machines are the following:

In the asynchronous communication model, there are no bounded limits on the processing and delay times in the sending of messages [21].

In the synchronous communication model, there are bounded limits on processing times, delay in sending messages, and possible margin of error in the clocks that control processes [22].

The partially synchronous communication model is an intermediate model that considers that the system behaves asynchronously for most of the time, until a certain time called GST (Global Stabilization Time) in which the system behaves synchronously, setting limits on the processing and sending times of messages, during an interval or the execution of a certain protocol [23].

Likewise, as already mentioned, the operating scenario of a distributed system is conditioned by the cryptographic techniques used, which will define whether it operates on authenticated communication channels [24]. This is achieved using public key techniques that, through the use of asymmetric cryptography or the existence of shared secrets between each pair of processes, allow the messages sent to be signed, guaranteeing authentication, integrity and non-repudiation. In both cases the primitives that define the digital signature algorithm must be placed on collision-resistant hash functions.

PBFT Consensus Method Applied to the Blockchain Networks

The study of consensus in a Blockchain network has been much explored since it represents for many authors the critical point for this technology to last over time. We will then review some of these works, considering their relevance in the topic covered in this article. Researchers have made a systematic analysis of the development and application direction of BFT algorithms based on the hybrid failure model, the authors guide to design and select appropriate mechanisms and architecture of implementation [25]. The authors have analyzed the design options and applications of the TC assist mechanism (Reliable Components) and BFT algorithms based on the hybrid failure model. They also classify and compare different CT abstractions and their different implementation scenarios by proposing a flexible design framework. Whereas, by taking advantage of the assistance of the CT mechanism, they can ensure the secure processing of data outside the chain and build reliable blockchain systems. Besides, they claim that consensus algorithms based on this model can reach 50% tolerance to Byzantine failures and higher transaction throughput compared to PBFT [26].

Ramachandran et al. make a significant contribution to the blockchain community with a project called Trinity, which is a distributed publication-subscription agent with Byzantine fault tolerance and immutability based on blockchain [27].

The main findings of this research showed that Trinity consumes minimum computational resources, on the other hand, Trinity is the first case in which the components of blockchain technology have been merged with the publication-subscription messaging model, and the authors highlight the potential of blockchain and algorithms outside the world of crypto currencies. Padilha & Pedone, outside the blockchain environment, express that Byzantine Fault-Tolerant Services (BFT) generally have more significant latency, compared to simple client-server interactions and limited scalability, in the sense that adding replicas does not

They translate into higher performance [28]. The authors in their study carried out the construction of a scalable storage system that was tolerant to Byzantine failures which are based on the concept of mini-transactions; to carry out this they propose a new protocol of atomic compromise that tolerates malicious clients and servers.

On the other hand, studied in detail the deficiencies of the consensus methods applied to blockchain networks for consortiums, the authors present a Reputation algorithm based on Byzantine Fault Tolerance (RBFT), another the contribution of this work is that it shows a primary scheme based on reputation as well. As an experimental result of this study, it was obtained that the RBFT offers improvements over the PBFT [29].

Kotla et al. present Zyzzyva, a protocol that uses speculation to reduce the cost of BFT replication. In Zyzzyva, replicas respond to a client's request without first executing an expensive three-phase confirmation protocol to agree on the order to process the requests; instead they receive optimistically the rule proposed by a primary server, they handle the request and respond immediately to the client, this being the main contribution of their investigation, concluding that BFT overhead should not be considered an obstacle to employing BFT replication, even for many high-demand services [30].

Byzantine Fault-Tolerant Protocols (BFT) have been used in blockchains because of their high performance and rapid block acceptance although these protocols have weakness due to lack of scalability to support a large number of nodes in the network, according to [31]. The authors highlight recent improvements to the standard Byzantine Fault Tolerance Protocol (PBFT). The authors consider that evaluating the performance and reliability of the different BFT-based protocols in the context of blockchains will give users a better picture of the behavior and scalability of these protocols in different circumstances. With this purpose in their work they implemented and evaluated the performance of different protocols based on BFT for blockchains under normal conditions and cases in which Byzantine failures are found in the network, also, they have performed the calculation of the reliability of each protocol under the desired performance [32].

Network Growth in Terms of the Nodes and the Propagation of a Message

One of the most recurring concerns in designers of blockchain-based application architectures is about scalability, understood in this context as the ability of an application to maintain network growth without losing its basic features that make it work. In this regard, we ask ourselves, how to maintain the consensus of the network as the network grows. According to Castro & Liskov, we know that a system is resistant to Byzantine failures if it satisfies the equation

$$n \geq 3f+1 \quad (1)$$

Where n is the number of nodes of the network and f those nodes that can be faulty. That means that a minimum of 3f + 1 node is needed to ensure that there are enough no-faulty nodes.

Now, the propagation of a message on that network is given by the equation

$$m \geq 1+3f+3f\ (3f-f)+(3f+1)\ (3f+1)+3f-1 \quad (2)$$

Thus, we can establish a minimum level so that the message is replicated without failures. The explanation of the equation (2) can be found in the way a node, regardless it is faulty or not, makes the request for validating a message in the network.

From here we can see the table with the numbers that we have, according to the equations (1) and (2): **Table 1**.

Table 1: Faulty nodes conditionate the growth rate of a network.

Number of faulty nodes	Minimal nodes needed	Message replications
1	4	24
2	7	71
3	10	142
4	13	237
5	16	356
6	19	499
7	22	666
8	25	857
9	28	1072
10	31	1311

Finally, the Illustration shows us the growth rate of the network based on the number of nodes and the replications needed for maintaining the consensus with PBFT (**Figure 1**).

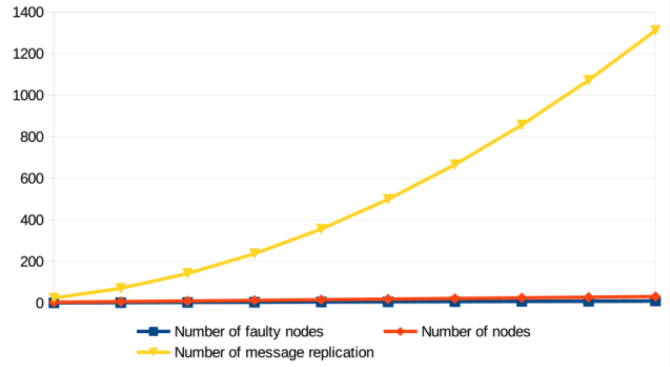

Figure 1: Growth rate of the nodes and message replications in a network with PBFT method.

Current Status of Networks using PBFT

Next, we present a number of blockchain networks using PBFT algorithms

Multichain is a platform where users can establish specific private blockchains that organizations can use for financial transactions, according to the Multi chain private blockchain white paper. The process of creating communication in Multi Chain occurs when the nodes in a blockchain connect. Multi Chain takes place when two blockchain nodes are connected. The identity of each node represents itself with an address with a list of permissions. Therefore, each node it represents sends a message to the other users. The P2P connection is canceled if they do not receive satisfactory results of the process [33].

Neo is an ecosystem-based on a blockchain on which smart contracts based on Ethereum are executed. Theoretically, Neo can process around 1000 or even reach 10,000 transactions per second. This Neo blockchain works as a registry and market for digital goods with smart contracts. It is a system that allows real-world assets to be digitized, allowing registration, deposit, transfer, negotiation, compensation, and liquidation through a decentralized and secure network. Neo can keep fully reliable records of transfers of digital assets associated with smart contracts. Any type of asset is capable of being digitized to be exchanged, bought, sold, distributed, or even modified following the rules agreed in said contract. The platform has the potential to be applied in areas such as crowd funding, stock trading, loans, loyalty programs, and private equity funds, supply chain financing, among others, and see Neo consensus mechanism [34].

Zilliqa is a blockchain platform based on a technology called sharding, which solves the problems of scalability, being the first implementation of sharding used in a public blockchain, which opens the possibility of a transaction rate per second that can match that of VISA, the largest payment processor in the world. The Zilliqa testnet can be accessed through a block explorer that allows users to see specific blocks, transactions, and addresses, and also know the transaction rate, the rate of the creation of blocks and the number of pairs, at through a user-friendly interface. A wallet application is also available for users to create test transactions. Initially, the test platform was launched in a "small scale" state, using less than 1000 nodes, which are instances of virtual computers on the Amazon Web Services (AWS) platform [35].

Hydra Chain is an open-source blockchain platform, developed by the brainbot technologies joint venture and the Ethereum project (Hc consensus). It is an extension of the Ethereum blockchain platform that provides support for creating private/allowed blockchain networks. As an extension of Ethereum, this blockchain is fully compatible with all API level and contract level protocols in Ethereum. There are several well-defined tools in Ethereum to create smart contracts and decentralized applications [36].

Symbiont Assembly is a blockchain platform to build networks in which multiple independent entities can share data and logic in real time. It is a decentralized database that replicates and executes the application logic in the form of smart contracts. This platform can be used to create financial instruments, such as loans and securities, in digital form from the beginning. The blockchain was designed to meet the standards of institutional finance in security, reliability, and performance.

Conclusion

After making a historical review of the emergence of the first blockchain and the practical byzantine fault tolerance consensus method, a systematic review of this consensus method has been carried out in this work, using a Multivocal review methodology to include references that are considered as gray literature, due to the novelty and emergence of this new technology. It was also seen, as a result of this investigation, how the Byzantine fault tolerance is a problem that concerns not only the blockchain environment but any distributed system. This review allowed us to find that the tolerance to Byzantine failures is linked to the number of nodes that replicate the message that is sent, taking into account that there is a way to calculate the minimum number of nodes that must exist to maintain consensus in these systems. We have analyzed in detail the main blockchain networks that use the pBFT consensus method, and evaluated the conditions under which these networks are working at the time of making this article. With the study of this consensus method in blockchain networks, we have found that it represents, for many authors, a critical point for the blockchain technology and until now, improvement has been made to increase the tolerance percentage up to 50% of faulty tolerance.

References

1. Mingxiao D, Xiaofeng M, Zhe Z, Xiangwei W, Qijun C (2017) IEEE International conference on systems, man, and cybernetics.
2. Limba T, Stankevičius A, Andrulevičius A (2019) Cryptocurrency as Disruptive Technology. Entrepreneurship and Sustainability 4: 2068-2080.
3. Nakamoto S, (2008) Bitcoin: a peer-to-peer electronic cash system.
4. Antonopoulos AM (2017) Mastering Bit coin: Programming the open block chain 'O' Reilly media.
5. Dolev D (1982) The Byzantine generals strike again. J Algorithms 3: 14-30.
6. Haber S, Stornetta WS (1990) How to time-stamp a digital document. InConference on the Theory and Application of Cryptography 437-455.
7. Chaum D (1983) Blind signatures for untraceable payments. In Adv crypt 23: 199-203.
8. Back A. (2002) Hashcash-a denial of service counter-measure.
9. Briner RB, Denyer D (2012) Systematic review and evidence synthesis as a practice and scholarship tool. Oxford handbook 112-129.
10. Garousi V, Felderer M, Mäntylä MV (2016) The need for multivocal literature reviews in software engineering: complementing systematic literature reviews with grey literature. In Proceedings of the 20th international conference on evaluation and assessment in software engineering 1-6.
11. Bass L, Clements P, Kazman R (2003) Software architecture in practice: Addison-wesley professional.
12. Driscoll K, Hall B, Paulitsch M (2004) The real byzantine generals: 23rd Digital Avionics Systems Conference.
13. Malkhi D, Reiter M (1997) Unreliable intrusion detection in distributed computations. In Proceedings 10th Computer Security Foundations Workshop 116-124.
14. Lamport L, Shostak R, Pease M (2017) The byzantine general's problem. ACM Trans Progam Lang and Syet 4: 382-401.
15. Lim HC (2019) Enterprises and Future Disruptive Technological Innovations: In future of information and communication conference 533-540.
16. Reischuk R (1985) A new solution for the Byzantine generals problem. Inform Control 64: 23-42.

17. Castro M, Liskov B (2002) Practical Byzantine fault tolerance and proactive recovery. ACM T Comput Syst 20: 398-461.

18. Buchman E (2016) Tendermint: Byzantine fault tolerance in the age of blockchains.

19. Chandra TD, Toueg S (1996) Unreliable failure detectors for reliable distributed systems. J ACM 43: 225-67.

20. Padilha R, Pedone F (2011) Scalable byzantine fault-tolerant storage. IEEE/IFIP 41st International conference on dependable systems and networks workshops 171-175.

21. Attiya H, Welch J (2004) Distributed computing: fundamentals, simulations, and advanced topics.

22. Fischer MJ, Lynch N. xaterson MS (1985) Impossibility of distributed consensus one faulty process. J ACM 32: 374-82.

23. Zilliqa A (2019) technical whitepaper: Version 1.

24. Coulouris G, Dollimore J, Kindberg T (2000) Concepts and distributed systems.

25. Zhang Q, Qi Z, Liu X, Sun T, Lei K (2018) Research and Application of BFT Algorithms Based on the Hybrid Fault Model. IEEE International conference on hot information-centric networking 114-120.

26. Jalalzai MM, Richard G, Busch C (2019) An experimental evaluation of BFT protocols for blockchains. In International Conference on Block chain 34-48.

27. Ramachandran GS, Wright KL, Zheng L, Navaney P (2019) A byzantine fault-tolerant distributed publish-subscribe system with immutable blockchain-based persistence. IEEE International conference on blockchain and crypto currency 227-235.

28. Krishna TL, Ragunathan T (2019) Performance evaluation of speculative semantics-based algorithm for read operations in distributed file system. Int J Comm Network Distr Syst 22: 275-293.

29. Driscoll K, Hall B, Sivencrona H, Zumsteg P (2003) Byzantine fault tolerance, from theory to reality. International conference on computer safety, reliability, and security 235-248.

30. Kotla R, Alvisi L, Dahlin M, Clement A, Wong E (2010) Speculative byzantine fault tolerance. ACM T Comput Syst 27: 1-39.

31. Jalalzai MM, Richard G, Busch C (2019) An experimental evaluation of BFT protocols for blockchains. In International Conference on Block chain 34-48.

32. Bracha G. (1987) Asynchronous Byzantine agreement protocols. Inf Comput 75: 130-43.

33. Lynch NA (1996) Distributed algorithms.

34. Lei K, Zhang Q, Xu L, Qi Z (2018) Reputation-based byzantine fault-tolerance for consortium blockchain. IEEE 24th International Conference on Parallel and Distributed Systems 604-611.

35. Zhang Y, Tang Z, Huang J, Ding Y, (2020) A Decentralized Model for Spatial Data Digital Rights Management. Int J Geo Inform 9: 84-85.

36. Xu M, Chen X, Kou G (2019) A systematic review of blockchain. Fin Innov 5: 27-20.