Improving data integrity and fault tolerance in IoT networks with Blockchain: on the search for suitable consensus mechanisms

Author: Michael Beekhuizen[1] , **Supervisor: Miray Ayşen**[1] , **Responsible Professor: Zekeriya Erkin**[1]

[1]Cyber Security Group
Department of Intelligent Systems
Delft University of Technology

Abstract

IoT devices have grown rapidly over the past few years. IoT devices are mostly connected to a central server that stores the data and handles end-to-end communication. Due to the increase of IoT devices, the latency with the server increases. Furthermore, when using a central server the data is at risk of being deleted or tampered with. To mitigate these issues blockchain could be integrated with the IoT devices to create a decentralized framework. This paper discusses how IoT integrated with blockchain can solve the problems with data integrity and fault tolerance in current IoT frameworks. Furthermore, different consensus mechanisms are compared and improvements are given to make the mechanisms suitable for IoT devices. The paper concludes by stating that G-PBFT, BFT-SMaRt and Tangle/Jointgraph are the most suitable consensus mechanisms for IoT devices with regard to computational power, throughput, latency and Byzantine fault tolerance. Moreover, two improvements with regard to reducing the latency and increasing the trust in G-PBFT are given.

1 Introduction

The Internet of Things (IoT) industry has grown rapidly over the past few years. By 2025, it is estimated that the industry consists of 50 billion devices [1]. Most of these devices are connected to one central server that stores and handles the data and thus the framework is centralized [1]. Due to this growth of IoT devices, there is an increasing load on these servers. "The central server is no longer efficient enough for handling large amount of data as well as end to end communications" [1, p. 2], therefore the latency with the centralized framework will increase [2]. Another problem is that centralized servers have the risk that data can be deleted or tampered with [2], [1]. This will reduce the availability and the integrity of the data in the network. In case of such an event or malfunction of the server itself, the entire network is at risk of being paralyzed [3]. Thus the server can become a single point of failure in the framework [4].

To solve the aforementioned issues, decentralization can be an option. Another option, adding multiple servers or replicas, is not always the best solution. The server owner has an extra overhead of creating, maintaining and updating these replicas [5]. For server owners, this can be a large, difficult and expensive task. Furthermore, when the data is updated the communication overhead is high [6]. For applications where a lot of data is added or changed this can become an issue. Moreover, the cloud paradigm favours availability over data consistency to provide higher scalability [5].

Blockchain is a framework that can help to decentralize the IoT network. Firstly, it uses a point-to-point (P2P) network to connect the devices with each other [7]. Because of this P2P network the devices can communicate directly and are not dependent on the server. This is reducing the latency in the network because the network load is split among the nodes. Secondly, the risk of deleting or tampering with the data is resolved in blockchain by using consensus mechanisms and immutable data [8]. Byzantine fault tolerance consensus mechanisms can deal with malicious nodes and reject false or malicious information, so the integrity of the data is protected. Thirdly, every full node in the blockchain has a copy of the ledger. In this way, when data is leaked or gets corrupted, it can detect this and resolve it. So this improves the resilience of the network and data [2]. Furthermore, blockchain ensures that no single authority can tamper or delete the data. Besides this, blockchain also ensures high availability of data and communication. This is not always the case with centralized servers according to Ali *et al.* [2]. So blockchain will give at least the same level of availability but with higher data integrity and lower latency.

This research gives a literature review and tries to find a way to improve data integrity and fault tolerance via decentralization with blockchain because blockchain can achieve a similar or better data availability and fault tolerance than centralized frameworks with an equal or smaller latency with achieving immutable data. The main question to answer in this paper is: "How can blockchain-based IoT frameworks solve the problem of fault tolerance in current IoT frameworks with regard to computational power, scalability and fault tolerance?". The fault tolerance in this context is about the degree to which the network can still function correctly. That means with preferably no data manipulation/corruption, low communication latency and high data availability. The network can function incorrectly because of malicious data changes, corrupted data or malfunctioning of the server. Be-

fore blockchain can be integrated there are a number of challenges to solve. One of these challenges is to find a suitable consensus mechanism for IoT devices. Consensus mechanisms are algorithms that make sure only one node can publish a block and that other nodes must validate that block. Moreover, it ensures that no one single node can change information without other nodes agreeing on it, so it can tolerate a number of malicious nodes in the network.

In this paper, currently implemented and proposed consensus mechanisms are compared with regard to computational power, scalability, throughput, latency and percentage of Byzantine fault tolerance. Furthermore, improvements are given such that the frameworks are better suitable for IoT devices in terms of computational power, latency and throughput.

The paper is structured as follows: Section 2 discusses the related work and Section 3 describes the methodology. Section 4 describes the fundamentals for understanding this paper. It discusses IoT, blockchain, consensus mechanisms in general and the integration of blockchain and IoT. Next in Section 5, eleven currently used or proposed consensus mechanisms are explained and a comparison between all of them is made. Section 6, gives the result of the comparison and improvements are given. The next section, Responsible Research (7), discusses the ethical implications of this research. The paper ends with a discussion and the conclusions and future work in Sections 8 and 9, respectively.

2 Related Work

In previous years, research was going on how to integrate blockchain with IoT. In this section, some of the work is discussed and explained what is still missing.

In 2018, Reyna *et al.* [8], published a paper where the challenges and opportunities of integrating blockchain with IoT are discussed. The paper discusses main challenges such as storage capacity and scalability. It concludes by stating that there must be more research on how to overcome challenges like storage capacity, scalability, security and privacy. This is because of the possible critical systems IoT networks can become. Furthermore, it is concluded that consensus mechanisms will play a major role in this. Bhushan *et al.* [1], discussed in their paper that blockchain has numerous advantages that can improve IoT networks but it still faces some challenges before IoT can use it. These challenges exist of scalability issues, constrained IoT devices, blockchain infrastructure and more. A paper written by Makhdoom *et al.* [9, p. 10], stated that the main challenges include: "IoT-centric TX and block validation rules, IoT-oriented consensus protocol, fast TX confirmation for real-time IoT systems, scalability, and secure device integration to the Blockchain."

One of the challenges before integrating blockchain with IoT is to have a consensus algorithm that is IoT oriented. Bodkhe *et al.* [10], did a survey on consensus mechanisms for resource constrained devices like IoT. In the paper they stated that: "the consensus protocols developed for IoT are not power effective." [10, p. 27] and scalability and throughput are still a problem with currently developed consensus mechanisms.

3 Methodology

As mentioned earlier in this paper, this research compares different consensus mechanisms for blockchain and gives ideas to improve the mechanisms so they are better suitable for IoT. To find currently implemented consensus mechanisms, papers were found and read that discuss different mechanisms such as: [11], [12], [13] and [10]. For finding proposed mechanisms, Google Scholar and Web of Science were used with the following search terms: "consensus mechanism blockchain", "blockchain consensus IoT" and "lightweight consensus mechanism blockchain". The selected mechanisms will be compared on five criteria which are:

- Byzantine fault tolerance
- Scalability in the number of nodes
- Throughput or transactions per second (TPS)
- Latency or block confirmation time (BCT)
- Computational power

The first, third and fourth are quantitative criteria and the second and last one are qualitative criteria. The scalability is assigned to three different values: low, medium or high. It is hard to give an exact number for scalability, therefore this mechanism is used. If it gets assigned a low value it means that it is not scalable or it can function properly up to 50 nodes in a network. Medium is assigned if a mechanism is able to work up to 150 nodes. High is assigned if it can work with more than 150 nodes. The computational power criteria will also have criteria from low, medium to high. The high value corresponds to a computational expensive mechanism like PoW.

4 Background

In the following sections background information is discussed so that the reader is familiar with topics such as Internet of Things, Blockchain and consensus mechanisms.

4.1 Internet of Things

Internet of Things or IoT can be defined as a network of devices that are communicating with each other without human to machine or human to human interaction [14]. According to a paper written by Wortmann *et al.* [15], IoT can be used to add value to 'normal' things.

In [15], Wortmann *et al.* are simplifying the IoT stack into three components, the device referred to as Thing, the connectivity and the cloud. The cloud is needed because most of the IoT devices in the network are limited in computational power and storage capacity. The cloud is saving all the data and does computations with it. The cloud, however, is not always the best solution because it has some drawbacks.

Goyal *et al.* propose in [14] that all the drawbacks of this system can be categorized into the following categories: privacy, security, accountability, legal and general. One of the general issues has to do with the type of connection. All the IoT devices are sending information to one cloud service, therefore it is called a centralized network. A problem with this type of network is that if data is deleted or tampered with or the cloud server stops working due to a failure,

the whole network could stop working [1]. Another problem is that because IoT is growing very quickly there will be a lower bandwidth connection available to the cloud server. The server owner must increase the capacity and throughput of the server and the maintenance cost will increase [1]. To solve these problems there is research that focuses on making decentralized frameworks for IoT instead of the currently centralized frameworks.

4.2 Blockchain

Blockchain is a shared distributed ledger that is able to store transactions [2]. In [2], the most important features of blockchain are summarized: decentralization, immutability of data, auditability and fault tolerance. The blockchain network uses a peer-to-peer (P2P) network. This means that there is no central authority. Because every participant has the same shared ledger, or copy of the data, the data stored is transparent. The immutability of the data has to do with the structure. As the name blockchain suggests, data is stored in a block and these blocks are chained together [16]. A block consists of two main parts, a header and a data section. The header section contains different fields, depending on the implementation of the blockchain. The fields that are most common are: hash of the current block, hash of the previous block, a random nonce value and a timestamp [2]. The hash is calculated over the data and some of the header fields, this depends on the implementation. Because of this hash value and the linking to the previous block its hash value, data is tamper resistant. When changing data in a block, the hash is incorrect and the block that is linking to that block is also saying that it is incorrect. How longer the chain of blocks is, the more difficult it becomes to tamper with the data.

When talking about blockchain networks, there are two main types, permissionless and permissioned networks [17]. A permissionless network is a public network where everyone can join, read and write data to and from the blockchain. To prevent malicious users from writing data to the blockchain or tamper with it in any way, consensus mechanisms are used. More information about consensus mechanisms can be found in section 4.3. A permissioned network can be seen as a private network where everyone is known and authorized. The permissioned network type can be again split up into two different types: private blockchain and consortium blockchain [2]. A private blockchain is as the name suggests private and can be used inside an organization. A consortium blockchain on the other hand is used in between organizations where both parties may not trust each other fully. In a permissioned blockchain, not every user may have the same rights for reading and writing. An owner of the network could change these rights per user.

4.3 Consensus mechanisms

Consensus mechanisms are used to determine which node or user in the network is able to publish the next block. This is needed because especially in a permissionless network many users may want to publish a block and may not trust each other. "To make this work, blockchain technologies use consensus models to enable a group of mutually distrusting users to work together" [17, p. 18]. The consensus mechanism is also used to prevent a malicious user from taking over the blockchain or tamper with data [17]. This is more important in a permissionless network because the users are not trusted. Even in a permissioned network, where users are authorized, a consensus mechanism is used. In this case, a network owner may use a consensus mechanism that is faster but protects less against malicious users. This is a trade-off the network owner must make.

4.4 IoT and Blockchain

As discussed earlier, centralized IoT frameworks have drawbacks regarding fault tolerance, security, privacy and trust. To solve these issues blockchain technology can be used. The characteristics of blockchain discussed in section 4.2, are beneficial for IoT networks. When integrating blockchain with IoT we get [1]: Higher fault tolerance and reduce the bottleneck of the cloud server, information transparency, immutable data, use of smart contracts and enhanced security.

Unfortunately, there are a number of challenges to overcome before integrating Blockchain with IoT. IoT devices are mostly small devices that have small memory storage and low computational power. The PoW consensus mechanism is implemented in many blockchains, but IoT devices are not capable of dealing with this mechanism, because of the high computational power [7]. A principle of decentralization is that every node has a copy of the data, but for IoT devices this is problematic. In [7], the author states that Bitcoin requires a minimum of 200GB of storage and Ethereum even more with 1.5TB. This is too large for IoT devices to store. In the remainder of this paper, the focus is on the consensus mechanisms and how to adapt these for IoT devices.

5 Analysis

In the following section, different consensus mechanisms are explained. First of all, seven already implemented and used consensus mechanisms are explained and after that four proposed and evaluated consensus mechanisms are discussed.

5.1 Proof of Work

The most implemented and well-known consensus mechanism in blockchain is Proof of Work (PoW). Zoican *et al.* [11] explain in their paper how this mechanism works and why it is computationally expensive. The goal is that nodes in the network that generate a block compute a hash value based on the data of the block and a random nonce value, which can be changed. The computed hash value must be smaller than the hardness value of the PoW mechanism. This value changes periodically to maintain the same difficulty. This is computationally expensive because it has a complexity of $\mathcal{O}(16^k)$ where k is the number of leading zeroes. To check if a hash value is correct can be done in $\mathcal{O}(1)$ because only the number of leading zeroes needs to be checked. Due to the high computational power, this mechanism offers resistance to DDoS attacks and faulty blocks [2]. However, it is vulnerable to so-called 51% attacks [2]. In this attack, one user or one group of users has the majority of the computational power in the network and thus can control the mechanism.

5.2 Proof of Stake

To overcome the high computational power of PoW and its energy consumption, Proof of Stake(PoS) was developed [18]. In PoS, every node has a share in the network [17]. The network is selecting a node based on its stake in the network. This node is then the one that is able to publish a new block. There are multiple variants of this mechanism. What they all have in common is the "nothing at stake" problem [17]. When there are two forks, a user may use both chains and no chain will be the longest one. So there will be no single chain anymore. Besides this problem, an advantage is that it is using less computational power and less energy. Just like PoW, Proof of Stake suffers from a 51% problem. If a node has more than 50% of the stake then it gets selected with a probability of more than 50% and thus can control the network.

5.3 Proof of Elapsed Time

Proof of Elapsed Time (PoET) is an improved mechanism based on PoW [10].

This mechanism tries to reduce the computational work in PoW by creating a lottery. The node that wins the lottery creates the block [19]. The lottery is implemented by a timer that is counting down. The node where the timer is ended first is able to create the block. This timer is executed in a trusted environment like Intel SGX [19]. This is a disadvantage because to take part in the network every device needs to have a trusted environment. That is why this consensus mechanism is not suited for current IoT devices if they do not have a trusted environment.

5.4 RAFT

RAFT is a consensus mechanism that belongs to the family of crash-tolerant consensus protocols. RAFT has no resilience towards nodes that subverted but can withstand $n/2$ crashed nodes where n is the number of nodes in the network [13]. "It offers a solution for distributing the state machines over the different clusters of machines and ensures that all the transactions will be performed in a sequence." [10, p. 9]. In this mechanism, each round a leader is selected who will create a new block [19]. When the network suspects that the leader crashed or behaves in a malicious way, the network can decide to move to the next round and select a new leader. Due to this behaviour of only being a family of crash-tolerant consensus protocols, it is better to use this in a permissioned network where there is already some trust among the nodes.

5.5 PBFT

Practical Byzantine Fault Tolerance(PBFT) consensus mechanism is a mechanism that is part of the Byzantine Fault Tolerance(BFT) family [19]. The advantage of BFT mechanisms is that they are able to detect up to a certain amount of malicious nodes in a network. Therefore they are crash tolerant and Byzantine tolerant. The PBFT mechanism is able to work with less than 1/3 of the nodes being malicious or faulty [19]. "Permissioned consensus protocols rely on a semi-centralized consensus framework and a higher messaging overhead to provide immediate consensus finality and thus high transaction processing throughput." [20, p. 4]. The processing throughput of PBFT is estimated in [21] at 78 000 TPS. This is a desirable throughput for IoT frameworks but because the network has a large communication overhead of $\Theta(n^2)$, it is not scalable and can only be used in small networks [20].

5.6 BFT SMaRt

"BFT SMaRt is an open-source Java-based library implementing robust BFT state machine replication" [21, p. 1]. It is a library developed by Brazilian researchers and is said to achieve better performance than other already implemented SMR libraries[21]. Some researchers implemented a Byzantine fault-tolerant ordering service for Hyperledger with the use of this framework [22]. It could achieve a transaction speed of 10k per second with 0.5 second block conformation time. A major advantage of this framework is that BFT SMaRt is able to work with non-fixed networks [19]. This is different from PBFT which mainly works with a fixed size network.

5.7 DAG-based

Another solution proposed by a cryptocurrency IOTA is the framework Tangle [12]. This framework is DAG-based instead of the linear block linkage of blockchain. DAG stands for Directed Acyclic Graph. The structure of a block in Tangle is very similar. Instead of referring to the previous block its hash it is referring to two previous blocks their hash value. "Due to the unique design of tangle, it is a fast, infinitely scalable framework which makes it well-suited for IoT networks." [12, p. 13]. This framework has an estimated transaction speed of 1.5k per second and a confirmation time of around 10ms [19]. This is very fast and desirable for an IoT framework. A downside of this framework is that it is sort of centralized [12]. This is because it suffers from a 50% attack like PoW, to overcome this one trusted node in the network ensures that it has more than 50% of the power so there are no malicious attacks. Besides the Tangle framework which uses a DAG-based approach, Jointgraph and Hashgraph are also using this approach. Jointgraph is an improved version of Hashgraph and has a higher performance. Hashgraph and Jointgraph both work with less than 1/3 of all nodes being malicious. Hashgraph has a transaction speed per second of 10k and a confirmation time of 5 seconds [19]. In [23], Jointgraph is evaluated and outperforms Hashgraph at TPS and BCT. Both of these frameworks are highly scalable but Jointgraph outperforms Hashgraph. At 150 nodes in the experiment, Jointgraph did 60 events per second compared to 3 events per second in Hashgraph [23].

5.8 Proposed solution

In a paper called "Consensus Mechanism of IoT Based on Blockchain Technology"[24], a new consensus mechanism is explained that is suitable for lightweight IoT frameworks. It is using the Diffie-Hellman algorithm for key negotiation between the blockchain nodes. After a transaction is received, it will verify the signature of the data and will look with machine learning for abnormal data. This new framework tries to minimize the computational overhead of PoW and the communication overhead of PBFT by using a Verifiable Random

Function (VRF) [24]. It is using a round consensus mechanism and in each round the VRF is used to determine if the node is allowed to broadcast a block. When it is allowed it broadcasts the block and other nodes can verify it with a public key. VRF cannot preserve one main chain and thus forks are possible. To resolve this issue the longest chain rule is applied [24]. In the paper, the experimental results show that it has a throughput of 600-800 transactions per second and a maximum latency of 5 seconds.

5.9 G-PBFT

G-PBFT is an improved version of the existing PBFT mechanism. This mechanism tries to improve the scalability and the communication overhead of the current PBFT mechanism. The G in G-PBFT stands for Geographic and according to the authors in [25, p. 1] it is "a new location-based and scalable consensus protocol designed for IoT-blockchain applications.". This mechanism relies on the fact that most IoT devices are fixed in one location and those fixed devices are less likely to be malicious than mobile devices. The authors in the paper reason that fixed IoT devices, in general, may have more computational power and these fixed devices are probably owned by companies and therefore less likely to be malicious. This mechanism is more scalable than the existing PBFT mechanism because only a small number of fixed nodes or endorsers are doing the consensus mechanism instead of all nodes. Furthermore, endorsers are removed from the group of consensus nodes when they behave wrongly. This is done with an era switch [25]. With an era switch also new nodes can join the group under some conditions like when one node was all the time in the same geographical location. In the paper, they compared the improved G-PBFT version with the existing PBFT version. The improved version had an average delay of 5.64 seconds and the existing version 251.47 seconds when averaging the delay with 4 to 202 nodes in the network. This is a good improvement and G-PBFT has a delay of around 5-6 seconds from 40 nodes onwards, while PBFT grows almost exponentially.

5.10 PoBT

PoBT stands for Proof of block and trade. In the paper written by Biswas *et al.* [26], they first propose a new framework for the whole mechanism. Because IoT devices are resource constrained and thus cannot run blockchain on it, they are connected to nodes. A node is a device that is able to perform blockchain operations like storage and consensus. Every device is connected to one specific node. In the network, there is one membership service provider (MSP) and a certificate authority (CA) to create keys and configuration information. When an IoT device sends a transaction to a node, it first verifies the transaction and looks for who the transaction is. If the destination device is in its own network, thus connected to the same node, it uses a local consensus mechanism. Otherwise, it forwards the block to another node that has the destination device connected to it. This node verifies it again. When everything is verified the block is sent to an orderer. This orderer elects the nodes who do the consensus. This is where the mechanism is significantly different from others because it uses the ratio of destination nodes of the transactions. So if the transactions were done by one node, there can be one node randomly chosen, except the source/destination node, to verify the block. When more than 50% of all the participants agree, then the block gets added. Because of this ratio, it is faster than the standard BFT in Fabric [26]. When there are 50 nodes doing 250 trades, Fabric will have consensus in 1.25 seconds and PoBT in 200 milliseconds. The more nodes and trades there are, the better PoBT performs in comparison to the normal BFT mechanism.

5.11 PoEWAL

In a paper written by Raghav *et al.* [27], they propose a new consensus mechanism called PoEWAL (Proof of Elapsed Work And Luck). This mechanism tries to minimize energy consumption and computational power. It is using PoW but instead of constantly updating the hardness value, the devices will solve the puzzle partially. Each device will try to generate a hash with the highest number of leading zeroes. When the time is up, all the devices share the block and the device that had the highest number of zeroes will publish it [27]. This is the elapsed work part of the mechanism. The luck has to do with the fact that forking can exist when two devices have the same number of zeroes. When this is the case the device with the lowest nonce value wins. If the nonce values are the same again then the winner is the one with the lowest block hash value. In this way, they have a probabilistic consensus mechanism with no forking possibilities. An important aspect to note is the framework of the network. IoT devices are divided into clusters and each cluster has a head node that has enough capacity to store the ledger and perform blockchain operations. Furthermore, the IoT devices themselves participate in the consensus mechanism and not only the cluster head nodes. In the paper, they did an evaluating of the framework and concluded that it performs better than other probabilistic consensus mechanisms in terms of consensus time in seconds and energy consumption in Joule.

6 Comparison and results

In the following sections, the aforementioned frameworks are compared on a couple of criteria to see if they are suitable for IoT devices. These criteria consist of scalability in the number of nodes, the number of tolerated Byzantine nodes, the throughput and confirmation/validation time. Finally, the computational power required is taken into account. These criteria are chosen because IoT devices are resource constrained devices and an IoT network is mostly not a trusted environment. Moreover, high scalability, a high number of transactions per second and low latency are preferable in an IoT network. After this, the result of the comparison is discussed and improvements are given.

6.1 Comparison

The first mechanism to start comparing is Proof of Work. As stated in 5.1, PoW is computationally expensive to perform. Nodes need to mine for blocks and guess the nonce value to calculate the hash value. This is not suitable for the IoT devices due to the computational constraint they have. In terms

of throughput and confirmation time, 7 TPS and 10 min respectively, PoW is also not suitable for IoT [19]. IoT networks need a high throughput and a small confirmation time due to the enormous amount of data the sensors in the network can send. In terms of scalability, it is high because a lot of nodes can enter a blockchain network with PoW like Bitcoin without issues like exponential confirmation time increase or throughput decrease [28].

When looking at Proof of Stake, it already performs better than PoW in terms of computational power, throughput and confirmation time. Unfortunately, it still requires a lot of computational power and the throughput and BCT are still on the low side [29]. Both of these mechanisms can tolerate any number of malicious users if the sum of their computational power, or stake in the case of PoS, is less than 51% of the network its total computational power or stake [2].

Proof of Elapsed Time has a couple of advantages over PoW and PoS. It has a higher throughput and lower confirmation time of 2.3k TPS and less than 1 sec, respectively [19]. Whereas PoW and PoS are highly scalable in the number of nodes, PoET is less scalable, but requires less computational power than the previous two mechanisms. In terms of computational power, TPS and BCT, this mechanism could be suitable for IoT devices, but a disadvantage is that it needs specific hardware to work. Moreover, this mechanism cannot tolerate many malicious users, only $\Theta(\frac{\log \log n}{\log n})$ nodes need to be compromised for hijacking the system [30].

In comparison to PoET, RAFT cannot tolerate any Byzantine node in the network. RAFT is only crash tolerant and not Byzantine fault tolerant. In terms of computational power, scalability, TPS and BCT it outperforms the above-mentioned consensus mechanisms. It has a high scalability with a TPS of 7k-40k and a latency of less than 1 second [19]. The power required for RAFT is low to medium, but overall this mechanism could be suited for IoT when used in the right environment. Because it is only crash tolerant, all the nodes in the network should be trusted.

The next mechanism to compare is PBFT. This mechanism is a good candidate for IoT frameworks. It uses minimal computational power, has a high throughput of 78k TPS and a latency under 1 second [19]. Moreover, it can tolerate Byzantine nodes when there are more than 2/3 honest nodes in the network. Unfortunately, there is one big limitation of this mechanism and that is the scalability. The scalability in the number of nodes is low for this mechanism, because it has a complexity of $\theta(n^2)$ with regard to network overhead. BFT-SMaRt is an improvement of the previous PBFT implementation. It improved the scalability because the network does not have to be a fixed size [19]. Furthermore, the throughput and latency stay the same depending on the framework. It could reach 10k TPS in Fabric and around 80k TPS in Symbiont [19].

In comparison to all the other aforementioned mechanisms, the DAG-based solutions are different in terms of structure. Every block is referring to two older nodes that are not validated yet. This way of adding ensures that there can be parallel validation [12]. In terms of the comparison criteria the mechanisms scores high. It is highly scalable, achieves a TPS of 1.5k TPS with a BCT of 10ms [19]. Jointgraph has in comparison with Tangle a higher throughput but a higher latency. Both of these mechanisms can be used for IoT devices because the computational power is low. A disadvantage is that if there is no transaction, the transaction that is waited to be validated is not validated and thus the latency increases.

A proposed solution in [24] uses VRF to elect the block production node. This ensures a fast election. In the evaluation, it achieves a TPS of 600-800 and a latency of 5 seconds. The goal of this mechanism was to reduce the computational overhead of PoW and the network overhead of PBFT. Because the network overhead is less than PBFT, the scalability of this proposed mechanism is higher than PBFT and thus is stated as medium. The computational requirement is low to medium because the nodes need to create a key agreement protocol and a machine-learning algorithm to detect and reject outliers of sensor data. Overall it seems like a mechanism that is suitable for a trusted IoT network where there are not many devices (more than 300-400).

A more scalable version of the PBFT mechanism is G-PBFT. This mechanism achieves higher scalability due to the lower number of nodes that are executing the consensus. Due to this lower number of nodes, the latency decreases. In the evaluation, it resulted in a decrease of 97.6% when doing consensus with 202 nodes [25]. PBFT its latency was over 250 seconds while G-PBFT its latency was around 6 seconds. The other characteristics of PBFT still apply to this mechanism.

The PoBT mechanism uses the same principles as BFT mechanisms but with fewer nodes. In [26], they evaluated the mechanism against Fabric and PoBT is more scalable and uses less bandwidth. The TPS is not evaluated in this framework but the latency was less than one second.

The last mechanism PoEWAL is relying on partially solving the problem of PoW. Due to this, the computational power would be medium. In the evaluation in [27], it has a BCT of 0.75 seconds with 50 nodes. Unfortunately, also in this evaluation, the TPS was not stated, therefore not a proper recommendation of this and the previous mechanism can be given.

6.2 Result and improvements

In the following sections, the result of the comparison and improvements for some mechanisms is given. The next section discusses the comparison.

When looking at Table 1, some mechanisms are more suitable for IoT devices than others. The least suitable mechanism is PoW and after that PoS. This is the case because the computational power required is high for IoT devices and the TPS and latency is too low and high respectively. Furthermore, PoS is also not very suitable for IoT devices, because it is based on a currency to have a stake in the system. This is not the case in every IoT framework. The most suitable mechanism for IoT apart from the scalability criteria is PBFT. The mechanism can tolerate up to 1/3 of malicious nodes in the network, has a high TPS of 78k and a low BCT of less than one second. Lastly, the computational power required is low, and therefore more suitable for IoT devices. When taking the scalability criteria again into account, the improved version of PBFT: BFT-SMaRt and G-PBFT are suitable for IoT devices. G-PBFT is more scalable than BFT-SMaRt but

Table 1: Comparison of the eleven different consensus mechanisms on Byzantine tolerance, scalabilty in number of nodes, TPS, BCT and computational power.

Consensus mechanism	Byzantine tol.	Scalable # nodes*	TPS	BCT	Computational power*	IoT suitable
PoW [28]	51% power	High	7	10 min	High	No
PoS [29]	51% stake	High	125-256	2-10 min	Medium-High	No
PoET [19]	log log n / log n	Medium	2.3k	less 1 sec	Medium	Maybe
Raft [19]	0	High	7k-400k	less 1 sec	Low-Medium	Maybe
PBFT [19]	0.33	Low	78k	less 1 sec	Low	Yes
BFT-SMaRt [21]	0.33	Medium	10k	less 1 sec	Low	Yes
Tangle [19]	?	High	1.5k	10ms	Low	Yes
Jointgraph [23]	0.33	High	10k [a]	5 sec [a]	Low	Yes
Proposed solution [24]	?	Medium [b]	600-800	5 sec	Low-Medium	Maybe
G-PBFT [25]	0.33	High	10k [a]	5-6 sec	Low	Yes
PoBT [26]	?	High	?	in ms	Low	Maybe
PoEWAL [27]	50%?	High [b]	1k [a]	1 sec [a]	Medium	Maybe

[?] Question mark means value not known
[a] Value is derived from evaluating of another algorithm which was outperformed by the mechanism (This value can be seen as lower bound)
[b] Value is derived from evaluation in the corresponding paper
[*] Low, medium and high values are explained in section 3

has a higher latency than BFT-SMaRt. In terms of computational power, both mechanisms have low computational requirements. The two DAG-based solutions, Tangle and Jointgraph are also very promising solutions when looking at the comparison criteria. Both have a low computational requirement and high scalability in the number of nodes. Jointgraph has a higher throughput but slower BCT in comparison with Tangle. Tangle may suffer from the problem when there are no transactions, blocks are not validated and the latency could be large [29]. The latency and throughput in Tangle rely on the actual number of transactions the network is doing. In Jointgraph, where the gossip protocol is used, there is no such problem [23]. Jointgraph suffers from another problem and that is that it has a weak central node called a supervisor. When this supervisor node is down or acting maliciously, the network cannot reach consensus. To fix the problem in Jointgraph, the normal nodes could switch to the Hashgraph mechanism which was the basis Jointgraph was built on. For PoBT and PoEWAL, not enough information is available to make a correct comparison. What can be concluded is that the scalability, BCT and computational power have 'right' values to be suitable for IoT devices. Therefore no recommendation can be given for these mechanisms but ideas from these mechanisms can be used to improve the BCT and scalability.

In the following section observations of the mechanisms are given and some improvements to make the mechanisms more suitable for IoT devices. As stated in the previous section, PoW is not suitable for IoT devices due to its high computational power requirement, low throughput and BCT. Moreover, PBFT has high throughput and low BCT but poor scalability. When looking at alternative mechanisms two important observations can be made: mechanisms try to lower the computational requirement by finding another way of solving a cryptographic puzzle or solving it partially like in PoEWAL and the mechanisms try to reduce the communication overhead by creating a subset of nodes.

When looking at BFT variants there could be an improvement in terms of scalability and latency. G-PBFT is already a highly scalable mechanism that could be suitable for IoT devices. An improvement would be to try to reduce the BCT of this mechanism because it is now between 5 and 6 seconds. When this mechanism is used between a couple of fabrics or companies where the different locations do not lie on the same continent, this latency could increase due to the distance and possible increase of hops in the network [31]. To minimize the risk of increasing the latency, the mechanism its election process for nodes could be improved. In the current algorithm, a node is able to join the group of execution nodes if the device is at a fixed location for at least 72 hours. An improvement would be if a new node must be added to the group that they randomly select m nodes and select the nodes with the closest distance to the current average group location of the group. In this way, the distance to the current group is reduced and the risk of increasing the latency is reduced. When creating a whole new group this rule could also be applied but some caution needs to be taken into account. If there are multiple locations, the algorithm must not only select nodes from one location, even if this minimize the average distance. This is done to still create a bit of trust among the devices. Furthermore, for a malicious user, it could also be harder to

infect more than 1/3 of the nodes in the network if they are not all at the same location or in a small range.

To improve the scalability of the current PBFT mechanism, the network overhead of $\Theta(n^2)$ needs to be reduced, where n is the total number of nodes in the network. One way of achieving this is to reduce the number of nodes that participate in the consensus like G-PBFT or PoBT. In G-PBFT a group of nodes agree on a node participating when the node can verify its location over the past hours/days is fixed. PoBT has a single node that select nodes that need to do the consensus. As an improvement for the scalability of PBFT, there needs to be a fixed set of nodes that participate in the consensus. G-PBFT is using 40 nodes to do the consensus with, this will give a latency of 5-6.5 seconds. When using 32 nodes the latency is between 3.5-5 seconds. The mechanism needs to have a way of selecting suitable and honest nodes to do the consensus with. An idea is to use signatures and certificates as a proof of trustworthiness. The group of endorsers can then discuss if a particular node is allowed to be added to the system. When a node, say A, is acting maliciously, the node is removed from the consensus protocol and its reputation will be influenced negatively. If node B wants to join the group and has a certificate of A that B is trustful, then it is less likely that B is allowed to join. In this way, the group of nodes doing the consensus stays more trustworthy and due to this higher trust, 32 nodes could be used instead of 40. This will reduce the latency on average by $\frac{\frac{5+6.5}{2} - \frac{3.5+5}{2}}{\frac{3.5+5}{2}} * 100\% \approx 26.8\%$.

7 Responsible Research

In the following section, ethical aspects of the research and the reproducibility of the research methods are discussed.

In this research, consensus mechanisms for blockchain with IoT are discussed. An ethical implication of blockchain itself is the environmental footprint. The Proof of Work algorithm uses a lot of computational power and because it is a non-cooperative game, everyone is using a lot of resources to be the first one to have a suitable hash value. This is taking a lot of energy and blockchain even has the same carbon footprint as Azerbaijan [32]. This paper focuses on consensus mechanisms that are suitable for IoT devices and thus requires not much computational power. This will reduce the energy used and thus not increase the carbon footprint drastically. Moreover, society will also be influenced economically. When a decentralized approach is implemented, the server owner gets less traffic and can scale down. This means that the server owner gets, on the positive side, a lower maintenance cost. On the negative side, it will potentially get less income because fewer users pay now for the service.

Different consensus mechanisms are compared in this research. Moreover, a table is created with the values for all the comparison criteria: scalability in the number of nodes, number of Byzantine fault tolerance, throughput and confirmation time and computational power. To make sure the comparison can be reproduced, data is gathered from different scientific papers, where an evaluation of these mechanisms was performed. This data was used for the criteria: number of Byzantine fault tolerance, throughput and confirmation time, because these are exact numbers. The scalability and the computational power are more abstract criteria, where assigning a number to it is hard. To still make these values reproducible the reasoning behind it is explained and some reflection on the outcome are given. Lastly, all the references for the papers are given. In this way the data, evaluation and explanation of the mechanisms can be found to reproduce the comparison table.

8 Discussion

When comparing different consensus mechanisms five different criteria were used. Three were quantitative and two qualitative. To improve the comparison the computational power could be measured in another unit like average seconds to complete the consensus. Then all the mechanisms should be tested on the same device. Another point of improvement would be to evaluate PoBT and PoEWAL. When this is done there are accurate measurements of TPS and BCT for these mechanisms and the comparison is more complete. Based on the comparison, there can be concluded that G-PBFT, BFT-SMaRt and Tangle/Jointgraph are possible suitable consensus mechanisms for IoT. Although there were a couple of improvements that could be made, which were discussed in the analysis, these mechanisms were the best. One could argue that this subset of consensus mechanisms is far from complete to conclude this. This subset was created by analysing survey papers on consensus mechanisms in general and for IoT and chose the ones that already were the most suitable or promising from those papers. Furthermore, this list was extended by searching for proposed consensus mechanisms for resource-constrained devices.

9 Conclusions and Future Work

In this paper, the main research question is: "How can blockchain-based IoT frameworks solve the problem of fault tolerance in current IoT frameworks with regard to computational power, scalability and fault tolerance?". Blockchain-based IoT frameworks can solve the fault tolerance problem in current IoT frameworks by making use of a point-to-point (P2P) network and a suitable consensus mechanism. Another question in this research was, what are suitable consensus mechanisms for IoT and which possible improvements could be made? After doing the analysis of eleven currently implemented or proposed consensus mechanisms and doing a comparison among them, there were a couple of suitable candidates. These suitable consensus mechanisms are: G-PBFT, BFT-SMaRt and Tangle/Jointgraph. These mechanisms have the most suitable characteristics for IoT devices: high scalability in the number of nodes, Byzantine fault tolerant, high number of transactions per second, low latency and low computational power requirement. Although these mechanisms were the most suitable a number of improvements could be made. In G-PBFT the geographical location of a node is distributed among the network to create a certain amount of trust. When devices are active at a large distance from each other, the latency in the network could increase due to the distance and the number of extra hops. To reduce this latency, the algorithm could take the geographical location into account and select nodes that are the closest to each other.

Some caution needs to be taking into account, one does not want to have all devices very close to each other because then a malicious user could easily manipulate devices. When these nodes are spread out, it is much harder to do this. Another improvement is the trust in the network when dealing with G-PBFT. Now, nodes get trusted when they are at least 72 hours at the same location. This is a weak constraint because if a malicious user is long at the same location it could still be accepted. To overcome this issue, nodes can become a sort of society. Every node can have a digital certificate that is signed by another node. When a node has a digital certificate that was signed by a suspected malicious node, this node is less likely to be accepted.

In the future, it would be better for comparison if all frameworks could be compared in the same environment to get more and reliable data. Furthermore, it would be an idea to implement the possible improvements and take them into consideration when comparing the frameworks.

References

[1] B. Bhushan, C. Sahoo, P. Sinha, and A. Khamparia, "Unification of blockchain and internet of things (biot): requirements, working model, challenges and future directions," *Wireless Networks*, vol. 27, no. 1, pp. 55–90, 2021. [Online]. Available: https://dx.doi.org/10.1007/s11276-020-02445-6

[2] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, "Applications of blockchains in the internet of things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1676–1717, 2019. [Online]. Available: https://ieeexplore.ieee.org/ielx7/9739/8727625/08580364.pdf?tp=&arnumber=8580364&isnumber=8727625&ref=

[3] I. Mistry, S. Tanwar, S. Tyagi, and N. Kumar, "Blockchain for 5g-enabled iot for industrial automation: A systematic review, solutions, and challenges," *Mechanical Systems and Signal Processing*, vol. 135, p. 106382, 2020. [Online]. Available: https://dx.doi.org/10.1016/j.ymssp.2019.106382

[4] A. J. Dadhania and H. B. Patel, "Access control mechanism in internet of things using blockchain technology: A review," in *Proceedings of the 3rd International Conference on Intelligent Sustainable Systems, ICISS 2020*, 2020, Conference Proceedings, pp. 45–50. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85100812355&doi=10.1109%2fICISS49785.2020.9316126&partnerID=40&md5=19fa96b7226875d1eed4c0aee2a91ae4https://ieeexplore.ieee.org/document/9316126/

[5] M. B. Gudadhe and A. J. Agrawal, "Performance analysis survey of data replication strategies in cloud environment," in *Proceedings of the 2017 International Conference on Big Data Research*, 2017, pp. 38–43.

[6] S. U. R. Malik, S. U. Khan, S. J. Ewen, N. Tziritas, J. Kolodziej, A. Y. Zomaya, S. A. Madani, N. Min-Allah, L. Wang, and C.-Z. Xu, "Performance analysis of data intensive cloud systems based on data management and replication: a survey," *Distributed and Parallel Databases*, vol. 34, no. 2, pp. 179–215, 2016.

[7] L. Lao, Z. Li, S. Hou, B. Xiao, S. Guo, and Y. Yang, "A survey of iot applications in blockchain systems: Architecture, consensus, and traffic modeling," *ACM Comput. Surv.*, vol. 53, no. 1, p. Article 18, 2020. [Online]. Available: https://doi-org.tudelft.idm.oclc.org/10.1145/3372136

[8] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with iot. challenges and opportunities," *Future Generation Computer Systems*, vol. 88, pp. 173–190, 2018. [Online]. Available: https://dx.doi.org/10.1016/j.future.2018.05.046

[9] I. Makhdoom, M. Abolhasan, and W. Ni, "Blockchain for iot: The challenges and a way forward," in *ICETE 2018-Proceedings of the 15th International Joint Conference on e-Business and Telecommunications*, 2018, Conference Proceedings.

[10] U. Bodkhe, D. Mehta, S. Tanwar, P. Bhattacharya, P. K. Singh, and W.-C. Hong, "A survey on decentralized consensus mechanisms for cyber physical systems," *IEEE Access*, vol. 8, pp. 54 371–54 401, 2020. [Online]. Available: https://dx.doi.org/10.1109/access.2020.2981415

[11] S. Zoican, M. Vochin, R. Zoican, and D. Galatchi, "Blockchain and consensus algorithms in internet of things," in *2018 International Symposium on Electronics and Telecommunications (ISETC)*, 2018, Conference Proceedings, pp. 1–4.

[12] M. Salimitari, M. Chatterjee, and Y. P. Fallah, "A survey on consensus methods in blockchain for resource-constrained iot networks," *Internet of Things*, vol. 11, p. 100212, 2020. [Online]. Available: https://dx.doi.org/10.1016/j.iot.2020.100212

[13] C. Cachin and M. Vukolić, "Blockchain consensus protocols in the wild," *arXiv preprint arXiv:1707.01873*, 2017.

[14] P. Goyal, A. K. Sahoo, T. K. Sharma, and P. K. Singh, "Internet of things: Applications, security and privacy: A survey," *Materials Today: Proceedings*, vol. 34, pp. 752–759, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S221478532033385X

[15] F. Wortmann and K. Flüchter, "Internet of things," *Business & Information Systems Engineering*, vol. 57, no. 3, pp. 221–224, 2015. [Online]. Available: https://dx.doi.org/10.1007/s12599-015-0383-3

[16] R. Thakore, R. Vaghashiya, C. Patel, and N. Doshi, "Blockchain - based iot: A survey," *Procedia Computer Science*, vol. 155, pp. 704–709, 2019. [Online]. Available: https://dx.doi.org/10.1016/j.procs.2019.08.101

[17] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain technology overview," National Institute of Standards and Technology, Report, 2018. [Online]. Available: https://dx.doi.org/10.6028/nist.ir.8202

[18] K. J. O. Dwyer and D. Malone, "Bitcoin mining and its energy footprint," in *25th IET Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014)*, 2014, Conference Proceedings, pp. 280–285.

[19] S. Brotsis, K. Limniotis, G. Bendiab, N. Kolokotronis, and S. Shiaeles, "On the suitability of blockchain platforms for iot applications: Architectures, security, privacy, and performance," *Computer Networks*, vol. 191, p. 108005, 2021. [Online]. Available: https://dx.doi.org/10.1016/j.comnet.2021.108005

[20] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22 328–22 370, 2019. [Online]. Available: https://dx.doi.org/10.1109/access.2019.2896108

[21] A. Bessani, J. Sousa, and E. E. P. Alchieri, "State machine replication for the masses with bft-smart," in *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2014, Conference Proceedings, pp. 355–362.

[22] J. Sousa, A. Bessani, and M. Vukolic, "A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform," in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2018, Conference Proceedings, pp. 51–58.

[23] F. Xiang, W. Huaimin, S. Peichang, O. Xue, and Z. Xunhui, "Jointgraph: A dag-based efficient consensus algorithm for consortium blockchains," *Software: Practice and Experience*, 2019. [Online]. Available: https://dx.doi.org/10.1002/spe.2748

[24] Y. Wu, L. Song, L. Liu, J. Li, X. Li, and L. Zhou, "Consensus mechanism of iot based on blockchain technology," *Shock and Vibration*, vol. 2020, p. 8846429, 2020. [Online]. Available: https://doi.org/10.1155/2020/8846429

[25] L. Lao, X. Dai, B. Xiao, and S. Guo, "G-pbft: A location-based and scalable consensus protocol for iot-blockchain applications," in *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2020, Conference Proceedings, pp. 664–673.

[26] S. Biswas, K. Sharif, F. Li, S. Maharjan, S. P. Mohanty, and Y. Wang, "Pobt: A lightweight consensus algorithm for scalable iot business blockchain," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2343–2355, 2020. [Online]. Available: https://dx.doi.org/10.1109/jiot.2019.2958077

[27] Raghav, N. Andola, S. Venkatesan, and S. Verma, "Poewal: A lightweight consensus mechanism for blockchain in iot," *Pervasive and Mobile Computing*, vol. 69, p. 101291, 2020. [Online]. Available: https://dx.doi.org/10.1016/j.pmcj.2020.101291

[28] M. Vukolić, *The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication*. Springer International Publishing, 2016, pp. 112–125. [Online]. Available: https://dx.doi.org/10.1007/978-3-319-39028-4_9

[29] B. Cao, Z. Zhang, D. Feng, S. Zhang, L. Zhang, M. Peng, and Y. Li, "Performance analysis and comparison of pow, pos and dag based blockchains," *Digital Communications and Networks*, vol. 6, no. 4, pp. 480–485, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352864819301476

[30] L. Chen, L. Xu, N. Shah, Z. Gao, Y. Lu, and W. Shi, *On Security Analysis of Proof-of-Elapsed-Time (PoET)*. Springer International Publishing, 2017, pp. 282–297. [Online]. Available: https://dx.doi.org/10.1007/978-3-319-69084-1_19

[31] Z. Imran. How server location affects latency? (2015, October 1). [Online]. Available: https://www.cloudways.com/blog/how-server-location-affects-latency/

[32] IB. 5 Blockchain Problems: Security, Privacy, Legal, Regulatory, and Ethical Issues. (2020, March 23). [Online]. Available: https://blocksdecoded.com/blockchain-issues-security-privacy-legal-regulatory-ethical/