---

**Article:**
Alrubei, S., Ball, E. orcid.org/0000-0002-6283-5949, Rigelsford, J. et al. (1 more author) (2020) Latency and performance analyses of real-world wireless IoT-blockchain application. IEEE Sensors Journal, 20 (13). pp. 7372-7383. ISSN 1530-437X

https://doi.org/10.1109/JSEN.2020.2979031

---

# Latency and Performance Analyses of Real-World Wireless IoT-Blockchain Application

S. Alrubei, E. Ball, J. Rigelsford, and C. Willis, *University of Sheffield, Department of Electronic and Electrical Engineering.*

***Abstract*—The Internet of Things (IoT) is increasingly being utilized, by both businesses and individuals, for many applications. This utilization means increases in the smart devices that are connected to the Internet of Things, which will significantly increase the challenges related to devices' interconnectivity and management, data and user privacy, and network, data, and device security. At the same time, blockchain approaches provide a decentralized, immutable, and peer-to-peer ledger technology that could be the right answer to these challenges. Significant challenges, however, accompany the integration of blockchain into the Internet of Things, since IoT smart devices may suffer from resource and power constraints and blockchain is associated with scalability and delay issues. In this paper, a practical incorporation of blockchain into the Internet of Things is demonstrated using Ethereum Proof of Authority (PoA). This provides performance analyses, which include measurement of the transaction arrival time, the system end-to-end latency for different network implementations over cellular and Wi-Fi, and the average power consumption. This includes the study of the effect of network bandwidth on the stability and synchronization of all nodes on the blockchain network.**

***Index Terms*— Blockchain, Ethereum, IoT, Latency, Performance Analyses, Transaction Arrival Time, Ultrasonic Sensor.**

## I. Introduction

THE Internet of Things (IoT) in the age of technological revolution promises to be something new and different that will affect our daily lives. According to [1], by 2022, there will be around 29 billion devices connected to the IoT. These devices will be utilized in many applications, such as in healthcare, smart manufacturing, and smart cities. Many of these devices have limited computational power and storage capacity, yet they are generating large amounts of data. This makes them difficult to secure, vulnerable, and easy for intruders to target. Consequently, many security and privacy issues affect these devices [2].

The extensive production of vast amounts of data poses significant challenges, which can frustrate efforts to address the security and privacy of these devices and data. The first challenge is related to the distributed nature of IoT systems, which means that each connected device is a possible entry point and can be exploited by an intruder to launch an attack [3]. Typically, IoT systems trust a central entity, such as a cloud service provider, for data processing, security, and system management. This could introduce the risk of a single point of failure. IoT systems are utilized in applications such as vehicular networks, where real-time processing forms an integral part, and this requires system availability all the time [4]. This makes it vital to resolve the issues surrounding the use of a central entity for better system performance. Authentication of devices and users and data integrity represent another significant challenge [5]. IoT devices can currently exchange data for resources such as power; IoT systems also have the ability to collect data from many sensors and use them for making timely decisions [4]. This necessitates the preservation of the integrity of these data to ensure system safety and accuracy in decision-making processes.

Traditional security measures implemented within IoT are built around trusted centralized architectures [6]. This means that such solutions will suffer from limited scalability, high cost, and a single point of failure. Conversely, self-managed, decentralized, trustless architectures provide scalable, redundant, potentially autonomous, and secure solutions for IoT systems. One of the most notable trustless and decentralized architectures is the blockchain technology.

Blockchain has existed for a long time: in 1991, the authors of [7] proposed a solution based on cryptographically hashing a chain of items to timestamp documents. Nevertheless, it was not until 2008 that blockchain was reintroduced in a popular form through Bitcoin [8]. Since then, blockchain has attracted a lot of attention, especially in the financial world. Many other areas, however, have recently been exploring the prospects associated with this technology; these areas include IoT. Blockchain provides a robust and decentralized platform for trustful interactions and information exchange. Since IoT is a distributed, dynamic, and heterogonous system, it will greatly benefit from the decentralized, self-managed blockchain [6].

Blockchain and IoT are potentially an ideal fit, where blockchain can offer a solution to the challenges within IoT, such as data integrity, device authentication and authorizations, and system availability. Immense effort, however, is required to integrate the two technologies. This

This paragraph of the first footnote will contain the date on which you submitted your paper for review.

S. Alrubei is with the University of Sheffield, Western Bank, Sheffield, S10 2TN, UK (e-mail: salrubei1@ Sheffield.ac.uk).

E. Ball, is with the University of Sheffield, Western Bank, Sheffield, S10 2TN, UK (e-mail: e.a.ball@ Sheffield.ac.uk).

J. Rigelsford is with the University of Sheffield, Western Bank, Sheffield, S10 2TN, UK (e-mail: j.m.rigelsford@ Sheffield.ac.uk).

C. Willis is with the University of Sheffield, Western Bank, Sheffield, S10 2TN, UK (e-mail: callum.willis@ Sheffield.ac.uk).

is because IoT devices may be limited in power and storage; they also produce vast amounts of raw data that need to be processed in a suitable environment. At the same time, blockchain still suffers from some issues, such as scalability. Based on this, there is a need to study and evaluate the performance of blockchain-IoT application using a real-world use case. According to the authors of [9], who provide a comprehensive systematic literature review and analysis of blockchain solutions for IoT, most studies have not measured the complete transaction time from submission until the transaction is committed in the blockchain network. The authors of [9] also state that, for better performance analyses, 'the performance of the whole proof of concept (PoC) should be analysed from end to end, from the transaction being submitted until the transaction being included and committed'.

### A. Contribution

The following are the major contributions of the proposed research work in this paper:

- Practical implementation of an IoT-blockchain application for flood monitoring and detection using Ethereum Proof of Authority (PoA) [10].
- Utilization of Smart Contract to coordinate and automate the execution of decisions within IoT realm.
- A performance analysis is provided, which includes the measurement of the transaction arrival time and the system end-to-end latency for different IoT-blockchain network implementations over 3G cellular and Wi-Fi.
- A comprehensive study of the network stability and node synchronization for both network implementations for different transaction submission scenarios.
- IoT device's energy consumption measurements for both implementations (over Wi-Fi and over cellular networks).

### B. Organization

The rest of the paper is organized as follows. Section II presents some blockchain background, and the related work is discussed in section III. Section IV presents the system analysis; this is followed by our practical implementation, which is described in section V. Details of our results are in section VI, and followed by the discussion in section VII. Finally, we conclude our paper in section VIII.

## II. Blockchain

Blockchain can be defined as a chain of blocks containing records of transactions with necessary data; this makes it an immutable, peer-to-peer, decentralized technology. Blockchain offers great benefits to different applications, including IoT, due to its characteristics and the advantages it can confer on an application. These advantages include decentralization, immutability, security in the form of reliable identification and an authentication mechanism in the form of public encryption keys, and cost-effectiveness through eliminating the costs associated with architectures that rely on a central entity [11][12]. In the following subsections, we discuss some of the well-known consensus algorithms and some of blockchain's platforms.

### A. Consensus Algorithms

Proof of Work (PoW) was implemented within blockchain in bitcoin platforms [8]. It is permissionless and allows for building a secure and public platform. Nodes have the freedom to joining and leave the network as needed. The process of generating blocks requires nodes to compete with one another to solve a cryptographic puzzle. PoW is a secure algorithm as long as honest nodes form the majority of the network, but the computation power required for PoW is increasing; this results in higher energy consumption [13].

Proof of Stake (PoS) was introduced as a possible replacement for PoW due to its lower use of energy [14]. A mining process is conducted based on currency ownership: the higher the stake a node has in the currency, the greater its chance to mine the next block. In PoS, no computation power is needed to find the hash. Nevertheless, this constitutes a consensus disadvantage to nodes that do not have a high stake in the currency, which will result in rich nodes becoming richer. It is also vulnerable to 'Nothing at Stake' attack, where nodes could mine multiple blocks, resulting in different forks [15].

The Proof of Authority (PoA) consensus protocol belongs to the family of Byzantine fault-tolerant algorithms [10]. This protocol is mainly used in permissioned networks; it is a simple protocol, which does not entail any extensive computation work, such as finding the nonce to mine blocks. The network relies on trusted nodes, called authorities, to mine and propagate blocks.

### B. Blockchain Platforms

Bitcoin is a digital currency based on the blockchain technology introduced by [8] in 2008. Bitcoin implements the PoW consensus algorithm to mine blocks and ensure the security of the network. It records all transactions and makes them available to the public in an immutable and decentralized distributed ledger. Bitcoin, with its use of the PoW protocol, requires a lot of energy and computation power, and this makes bitcoin undesirable and difficult to implement in the IoT realm.

IOTA is a cryptocurrency that is intended for the IoT industry and uses the tangle protocol [16]. The transactions issued by the nodes in the tangle constitute the site set of the tangle graph, which is the ledger that stores transactions. Each transaction must approve two previous transactions. Direct approval requires the node that issued the transaction to do some work in the form of solving a cryptographic puzzle in order to accomplish the approval.

Ethereum is an open blockchain platform that allows users to deploy their distributed applications (dApps) [17]. Ethereum implements its own Ethereum Virtual Machine (EVM). Ethereum is an easy platform to deploy on many architectures, including ARM-based Linux systems. Compared to other platforms, Ethereum PoA is a suitable implementation within IoT because it consumes less power. The only drawback is that it is a permissioned protocol, which requires nodes to be authorized before they can join the network.

TABLE I
COMPARATIVE ANALYSES BETWEEN OUR WORK AND THE RELATED WORK

| Paper | C1 | C2 | C3 | C4 | C5 | C6 |
|---|---|---|---|---|---|---|
| [18] and 19] | No - Proposed an architecture and uses simulation for validation | Platform like Bitcoin but without PoW (more of PoA) | Qualitative evaluation and Simulation Results | No | No | Simulated results of energy consumption of the smart home miner - which is a PC(not the end IoT device) |
| [20] | No - only simulation using Bitcoin simulator in NS-3 | Bitcoin PoW in a sub-blockchain architecture | Yes- simulated performance analyses that includes; block sizes and block generation intervals, and evaluating the effect of varying the number of IoT devices and their locations | No | No | No |
| [21] | Yes - prototype (3 nodes and a smartphone) | Ethereum | No | No | No | No |
| [22] | Yes- Proposed Blockchain Platform for Industrial Internet of Things (BPIIoT), and validate it with practical implementation | Ethereum | No-only evaluation without measurements | No | No | No |
| [23] | No | Planning to use Ethereum | No | No | No | No |
| [24] | No | Not clear | Simulation Results | No | No | No |
| [25] | No | Proposed their own consensus algorithm | Evaluation and Simulation Results | No | No | No |
| [26] | No | Ethereum | Evaluation and Simulation Results | No | No | No |
| This paper | Yes- We deployed 16 IoT nodes around the city of Sheffield, UK. | Ethereum PoA – deployed private network. | Yes- we provided measurements of system latency including block propagation and importing, energy consumption, and node synchronizations and network stability | Yes, we also provided a module that predicts the latency based on the number of nodes | Yes | Yes – using pragmatic IoT devices. |

**Notes**: C1: **Practical Deployment, C2: Blockchain Platform, C3: Performance Analyses, C4: End-End System Latency Measurements, C5: Study of Network Stability and Nodes Synchronization, C6: Energy Consumption Measurements**

## III. RELATED WORK

The authors of [18], [19] introduce an architecture for blockchain implementation within an IoT application, namely smart homes, for access control purposes. The architecture relies on a central entity, which is a local home miner, to mine blocks and implement the access control policy. This architecture ensures the confidentiality of data through predefined policy. The introduction of the centralized miner, however, introduces the risk of a single point of failure and makes the architecture more of a centralized one.

The authors of [20] provide a simulated performance analysis of blockchain PoW implantation in IoT, which includes transaction throughput, average traffic on the network, and stall blocks. The authors propose a hybrid IoT sub-blockchain architecture based on a set of rules. The sub-blockchains use PoW as a consensus algorithm and Byzantine fault-tolerant (BFT) protocols for interconnectivity between sub-blockchains. While this work provides some performance analyses in a simulated environment, it does not provide any performance analyses of practical implementation.

Another solution based on blockchain for the IoT is the one provided by [21]. The authors in this work provide a proof of concept on how to use Ethereum blockchain to manage IoT devices through the implementation of smart contracts. The system enables the control of devices based on policy stored in the smart contract. As a concept, this is a good example showing the benefits of using smart contracts and blockchain to control IoT devices.

Blockchain Platform for Industrial IoT (BPIIoT), proposed by the authors of [22], is a platform that is based on Ethereum blockchain and consists of a single-board computer, connectivity to the cloud and the blockchain network, and an interface to control sensors and actuators and to collect data. The main aim of this platform is to facilitate the decentralized communication and dealings between machines themselves or the communication between machines and humans. This provides the ability in the industrial setup to monitor the health status of machines, automate the diagnostics process, and ensure the availability of a secure and shared distributed ledger for transaction records. This platform is based on permissioned blockchain, which offers a trusted platform to ensure the safety of machines and the security of transactions.

The work in [23] proposes CitySense which leveraged blockchain technology to solve the problem surrounding the sensors' data storage and management within smart cities. Moreover, for software development the authors apply the adaptive and iterative SCRUM methodology. This is a proposal that relies on a central collection endpoint, which is against the decentralize concept of blockchain. The authors of [24] proposed SURVIVOR a blockchain based framework in a software-defined networking (SDN) architecture to provide a secure platform for energy trading between vehicle-to-grid (V2G) for charging of electric vehicles (EVs). Another work by [25] proposed a framework called BEST based on blockchain and SDN technologies for energy trading and charging of electric vehicles (EVs) in secure and safe environment. While both frameworks provide

good solutions to energy trading, they are still just proposals that need real-world implementation and validation. The work by [26] is based on using blockchain and SDN technologies to build an architecture of two parts that combined both features of centralization and distribution for smart cities implementation. This is another good example of blockchain based solution for smart cities that need real world implementation and validation.

In terms of performance analyses and providing complete measurement of transaction arrival time and end-to-end system latency, this information is not provided by any of the authors of the current related work. In this work, we provide a performance analyses of the system latency, network synchronization and stability, and energy consumption. Table I provides a comparative analysis between our work and related works.

## IV. System Analyses

The system under consideration is based on the Ethereum clique PoA [10]. This protocol allows predefined authorities (signers) to mine and propagate blocks to other nodes in the network. Once a block is received by other nodes, its transactions are immediately confirmed, resulting in a latency of 1 block, because the protocol has been built around the trust of the authorized nodes. This provides significant benefits, in terms of lowering the network latency and energy consumption, and is ideal for implementation as a client in an IoT realm. Ethereum has its own EVM, which allows for the deployment of dApps, such as smart contracts, stored on the blockchain and can be triggered and executed by transactions on each node [27]. Ethereum has two sets of accounts: accounts owned by private keys controlled by users, called externally owned accounts (EOAs), and contract accounts, which users can activate using their EOAs. It also has its own currency, called Ether, and its own crypto fuel, called 'gas'. In the following sections, the analysis of transaction arrival time on the Ethereum network is presented. Table II presents a list of the variables used and their meaning.

### A. System Characteristics

We consider an Ethereum blockchain network with block generation based on the block period (*BP*) of a fixed value. The system has the following characteristics:

- Multiple nodes are connected to one another in a peer-to-peer network via wireless links.
- Two different processes are the main traffic generators on this network: propagation of transactions and propagation of blocks through the network to all nodes; both are broadcast transmissions.

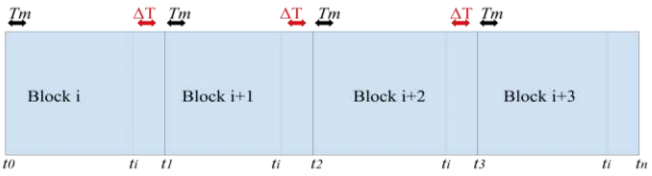



Fig. 1. Timing of Block Mining and Transactions Submission Ideal Time.

- We consider the case where the delay of the propagated transactions depends on the condition of the wireless network. This is called transaction propagation delay ($T_{pd}$).
- The size of transactions is assumed to be fixed, and only the gas charged by the miner for executing the transaction influences the block size.
- Nodes on the network are full nodes, where the full copy of the blockchain is stored locally and synched with the latest block in the network.
- The mining of blocks happens right at the start of the block period, at time $T_m$ (as shown in Fig. 1).
- Newly arrived transactions will not be mined until the next immediate block.
- Transactions are added to the block during the period $\Delta T$. Any transactions arriving during this period will not be considered for that block.
- The total number of transactions waiting in the transactions pool at any given time $t$ is $N_t$.
- Transactions are mined in batches; the maximum batch size is equal to the maximum block size, $B$.
- In Ethereum, the number of transactions that can be included in a block is based on the block gas limit ($B_g$) and the amount of gas consumed by each transaction ($Tg$).
- The interval between blocks is the block period (*BP*). After transactions are added to w block and mined, a miner will wait until the end of the *BP* to release the block to the network.
    - For every transaction, there is one *BP* service time.

TABLE II
DEFINITIONS OF VARIABLE USED

| Variables | Definition |
|---|---|
| $t_{0,1,2,3.....,n}$ | Times at which the miners release new blocks into the network. |
| $T$ | Time |
| Nv | Validators Nodes (store full copy of the blockchain and allowed to mine and propagate blocks). |
| $N_p$ | Participant Nodes( store full copy of the blockchain but are not allowed to mine and propagate blocks). |
| $BP = t_1\text{-}t_0$, $t_2\text{-}t_1, \ldots, t_{n+1}\text{-}t_n$ | Block period time (the minimum time between the release of new blocks). |
| *transaction arrival time TAT* | The time from transaction submission by a node until the transaction arrives on the network and can be seen by all nodes. |
| *Tx* | Transaction |
| $T_m$ | The time during which a miner mines the block. |
| $\Delta T$ | The time towards the end of a block; transactions arrive during this time will not be included in the next block. |
| $T_{pd}$ | The transaction propagation delay from transmission by a node until it arrives in a miner's transaction pool. |
| $t_i$ | The ideal time for transaction submission during the system steady state. |
| $S_{LP}$ | The period of time the sensor takes to measure the distance from the water level. |
| *transaction gas* $T_g$ | The amount of gas the transaction charges, to be executed or stored. |
| block gas limit $B_g$ | The maximum allowance of gas charges (the sum of all transactions' gas consumption). |
| ND | Number of Nodes |
| $N_t$ | The total number of transactions in a miner's transaction pool. |

### B. Synchronization Process

In IoT constrained devices there are two possible scenarios in terms of deploying blockchain clients. The first one is implementing a full node where a device has a full copy of the blockchain. In this protocol, devices are fully part of the network where they mine blocks, propagate blocks, send transactions, verify transactions and blocks. The second scenario is where IoT devices will act as a light node and keep track of a blockchain network and synchronize only the block headers, for example, the Ethereum Light client [28]. Nodes in this scenarios depends on how well they trust each other to access and check blocks and transactions.

In this work we will only consider the first scenario where devices are full nodes but could act differently in the network in terms of mining blocks and this will result in having two types of nodes. Nodes that keep full copy of the blockchain network locally and are able to mine blocks, validate them, and initiate and verify transactions and are called validators $N_v$. The second types are the nodes that keep a full copy of the blockchain network locally and are able initiate and verify transactions but not allowed to mine blocks and they are called participant nodes $N_p$. The length of the global chain at time $T$ can be describe as a $L(T)$. Since the validators $N_v$ are allowed to sign and propagate blocks then the length of the local copy is defined as $LN_v(T)$ where $LN_v(T) \geq L(T)$. On the other hand, the length of the local copy in $N_p$ should always be $LN_p(T) \leq L(T)$. The difference in the number of blocks between $N_p$ and the global chain can be calculated by the process $D(T) = L(T) - LN_p(T)$.

### C. Transaction Arrival Time During Steady State ($N_t \leq B$)

The probability of transaction arrival in the network is based on the Poisson process with arrival rate ( $\lambda$ ).

$$P(T \leq t) = 1 - e^{(-\lambda t)}$$

We let λ represent the rate at which blocks are added to the blockchain network; $\lambda = 1/BP$ blocks/sec, and we assume this rate for the remainder of this analysis. The time $t$ depends on the block period, the number of blocks ($n$) for which we need to wait before transactions arrive in the network, and the propagation delay ($T_{pd}$). If we assume that transaction submission at ($t_0$) (Fig. 1), then the probability for the transactions to arrive in the network after $n$ blocks is as follows:

$$P(n) = \begin{cases} 1 - e^{(-1/BP \times (n \times BP - T_{bp}))}, & T_{bp} < (n-1) \times BP \\ 0, & T_{bp} > (n-1) \times BP \end{cases} \quad (1)$$

This is true provided the transactions arrive before processing time $\Delta T$, that is to say ($T_{pd} < (BP - \Delta T)$); otherwise, $P(n) = 0$. Knowing the probability, we can calculate the transaction arrival time ($TAT$):

$$TAT = \frac{\ln(1 - P(n)}{-1/BP} + (T_{pd} - \sigma) \quad (2)$$

Where $\sigma$ is a variable that represents the system and smart contract processing time.

*Numerical Analysis*: If we assume that $\Delta T=0.2s$, we can calculate the probability of transaction arrival in block number $n$ for different values of $T_{pd}$ for $BP = [1, 2, 5, 10]$ seconds. In addition, using the values of P(2) and P(3) (i.e. arriving after two and three blocks) and assuming that $T_{pd}$ = 0.2s, we can calculate the transaction arrival time for $BP = [1,2,3,4,5,6,10,13,15,18,20]$ seconds. As can be seen in Fig. 2 and Fig. 3, it is clear that the blocks with shorter $BP$s (especially one and two seconds) are the most affected by $T_{pd}$; however, as the $BP$ increases, the $T_{pd}$ effect becomes negligible. This means that the longer block period (10s and above) should be implemented for better performance in networks with limited bandwidth.

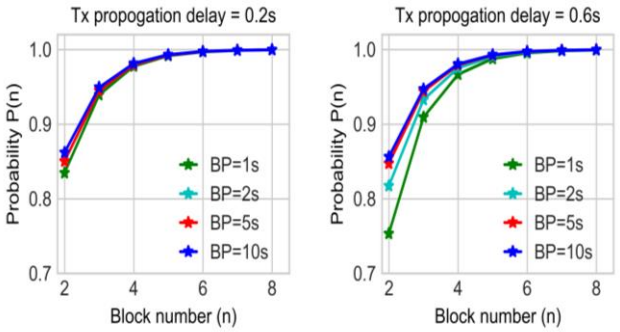



Fig. 2. Probability of transaction arrival after n blocks.

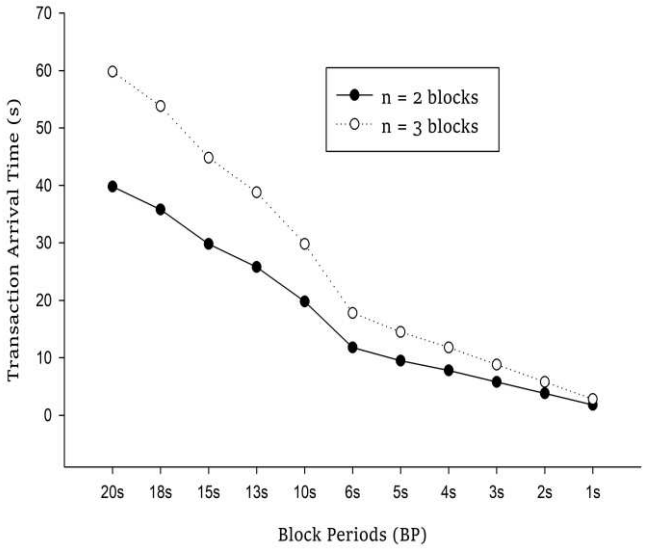



Fig. 3. Transactions Arrival Time for different Block Periods

### D. Transaction Arrival Time During the Busy Period ($N_t > B$)

During the busy period, where the system cannot accommodate all transactions waiting in the pool in one block, some transactions must wait in the pool for a number of block periods. We can define the maximum waiting time in the pool as *W*. If we assume that transactions are served on a first-come-first-served basis, then we can neglect $T_{pd}$, and *W* can be calculated as follows:

$$W = \left\lceil (Nt \times Tg)/Bg \right\rceil \times BP$$

In such cases, the probability of transaction arrival after *n* blocks is as follows:

$$P(n) = \begin{cases} 1 - e^{(-1/BP) \times \left( \left\lceil \frac{Nt \times Tg}{Bg} \right\rceil \times BP \right)}, & \left\lceil \frac{Nt \times Tg}{Bg} \right\rceil < (n-1) \times BP \\ 0, & \left\lceil \frac{Nt \times Tg}{Bg} \right\rceil > (n-1) \times BP \end{cases} \quad (3)$$

Knowing the probability, the transaction arrival time can again be calculated as follows:

$$TAT = \frac{\ln(1 - P(n))}{-1/BP} + \left( \left\lceil \frac{Nt \times Tg}{Bg} \right\rceil \times BP \right) + (T_{pd} - \sigma) \quad (4)$$

*Numerical Analysis*: If we assume that each node submits one transaction during the block period, *BP* = 20s and $T_g$ = 21,000. Using (3), we can calculate *P(n)*, and we calculated the transaction arrival time using (4). Fig. 4 and Fig. 5 both illustrate the effect of the number of nodes ND on the probability of arrival and how increasing the block gas limit can reduce the waiting time before transaction arrival in the network. It is clear from both figures that as ND increases, the total transactions will increase, resulting in increased waiting time for transactions in the pool. This waiting time can be reduced, however, by increasing the block gas limit.

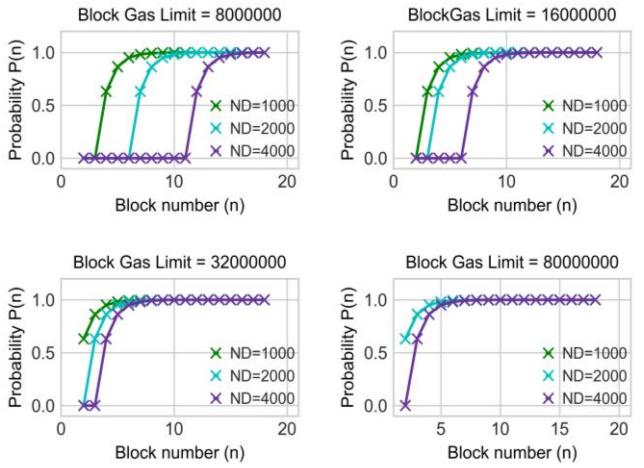



Fig. 4. Probability of transaction arrival during busy period.

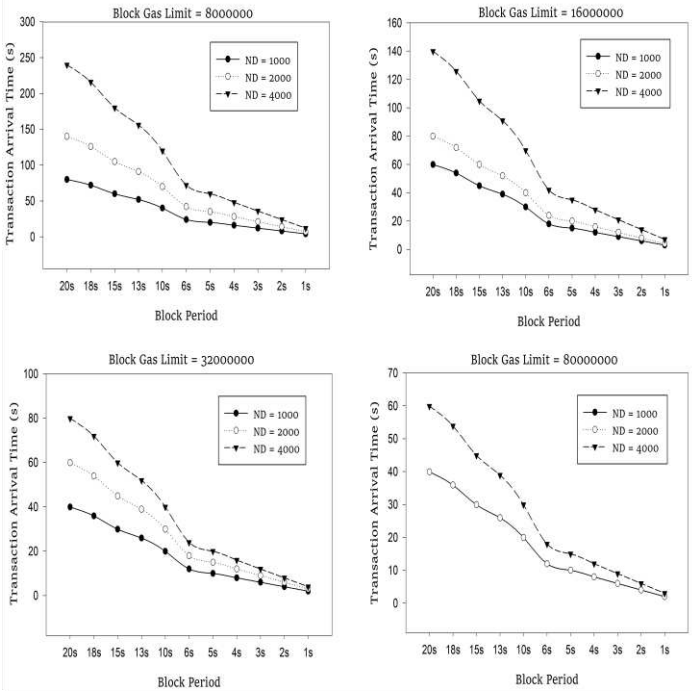



Fig. 5. Max Transaction Arrival Time.

## V. Practical Implementation

To perform the necessary measurements of the performance of Ethereum blockchain, a use case based on flood detection and control of a network was designed. The aim was to monitor a reservoir, tanks, or a river such that, in the case of a flood, a controlling pump could be automatically activated to discharge the water and prevent the flood from occurring.

### A. System Design

The system design includes the following:

- A network containing 16 nodes was created, with one node controlling the water pump.
- Nodes can communicate among themselves using wireless communication (Wi-Fi or cellular)
- Each node has an Ethereum Geth client (specifically, clique PoA) and has its own EOAs.
- A smart contract that includes the following functions was created:
  - A function to establish the initial value of the global positioning system (GPS) designated area and the threshold of the water level.
  - A function for extracting GPS longitude and latitude data to ensure the node is within the designated area.
  - A function to allow nodes to submit water level readings.
  - A voting algorithm, based on the majority function, which is only invoked by the node that controls the pump to calculate the number of flood detection nodes and trigger activation of the pump if the majority of nodes indicate that a flood is occurring.

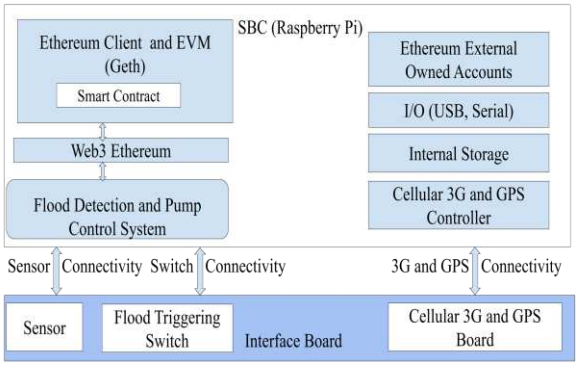



Fig. 6. System Design and Hardware

- Each node consists of the following hardware components:
  - Single-board computer (SBC) Raspberry Pi.
  - An ultrasonic sensor.
  - A cellular board in the form of Adafruit Fona 3G.
  - An interface board to facilitate communication between the SBC and the sensors and the Adafruit Fona 3g.

The diagram in Fig. 6 presents the different components of the system and their connectivity. The system was tested for different block periods over Wi-Fi and over a 3G cellular network.

### B. Test Setup

The system was tested in a controlled environment for flood detection and control, and it was successful. We continued testing, however, using a switch on the interface board to emulate flood detection, with nodes distributed around the city of Sheffield in the United Kingdom. This allowed us to focus the testing on aspects of blockchain. The test scenario includes the following:

- A peer-to-peer connection is achieved through the implementation of User Datagram Protocol (UDP) hole punching using a rendezvous server [29].
- Nodes were distributed around the city of Sheffield.
- The tests were conducted for *BP*s of 1, 2, 3, 4, 5, 10, 15, and 20 seconds.
- The transaction arrival time and the system end-to-end latency were measured.
- Python programs were developed for the purpose of monitoring the status of the network and reporting the timestamps of transaction submissions and the time of the consensus on the network and the change of status.
- The transaction submission time could occur at any time during the *BP*. Delaying transactions until as late in the *BP* as possible, however, can ensure that all events are detected and that the latency is reduced. Considering that the aim of our system was to monitor any changes in the environment (water levels), this was important. The system was tested for three different scenarios related to the transaction submission time for all *BP*s under consideration:
  - Transaction submission at the start of the *BP* ($t_0$).
  - Transaction submission randomly during the *BP*.
  - Transaction submission at the ideal time ($t_i$)

## VI. Results

For latency measurements, we used three different times to submit transactions to the smart contract: at $t_0$, randomly during the *BP*, and at $t_i$. The following sections present our latency measurement results as well as discussion and comments regarding these results.

### A. Ideal Time for Transaction Submission During the System Steady State

First, we calculated the ideal time for transaction submission. As can be seen in Fig. 1, the mining of *Block i* happens right at the start of the block period, at time $T_m$. We defined $t_i$ as the ideal submission time, which is the time towards the end of the *BP* and immediately before entering the critical period $\Delta T$. To calculate this ideal time, we had to identify the $\Delta T$ and calculate the sensor latching period ($S_{LP}$) and the transaction propagation delay ($T_{pd}$).

*Sensor Latching Period ($S_{LP}$)*: The $S_{LP}$ for different distances was measured. As the distance from the water level increases, the latching period will increase, forming a linear relationship. In our implemented case, the water level threshold was 10 cm; the average sensor latching period to measure this distance was 0.614ms.

*Transaction Propagation Delay ($T_{pd}$)*: Each node on the network submits transactions to the smart contract, and once they are accepted, they will be propagated to the other nodes on the network. The $T_{pd}$ was measured for both transmissions over the Wi-Fi network and the cellular network, and the results are shown in Table II. As can be seen from the table, propagation delay over the cellular network was higher than propagation delay over the Wi-Fi network. In this test, we used the Fona 3g board, which limits the connectivity to 3G.

*Critical Period $\Delta T$*: $\Delta T$ is the period during which miners fetch and add transactions to the new block. Based on our experiments and tests, we can conclude that the final ≈ 400 ms of the *BP* is the critical period, where any transaction arriving during this period has a very low probability of being included in the next block; instead, it will likely have to wait for the block after the next one.

From the above measurements of $\Delta T$, $S_{LP}$, and $T_{pd}$, the ideal time $t_i$ for transaction submission can be calculated as follows:

$$t_i = BP - \left(\Delta T + AverageS_{LP} + MaxT_{pd}\right) \quad (5)$$

TABLE III
TRANSACTIONS PROPAGATION DELAY (TPD)

| Over Wi-Fi | | | | Over Cellular (3G) | | | |
|---|---|---|---|---|---|---|---|
| Avg | Max | Min | STD | Avg | Max | Min | STD |
| 0.09 s | 0.2s | 0.064s | 0.32s | 1.8s | 3.4s | 0.6s | 0.7s |

For *BP* = 20s and water level = 10cm and testing over the Wi-Fi network, we obtained the following:

$t_i = 20 - (0.4 + 0.000614 + 0.2) \approx 19.39$ s

Using the measurement of $t_i$, we were able to monitor the water level during the *BP* until $t_i$, at which point we were able to submit the transactions. By doing this, we achieved the following:

- Reduce the overall system latency.
- Ensure that all flood events can be detected on time and without extra delay by continuously monitoring the water level because submitting transactions at the start of the block could have resulted in a flood incident occurring after the submission, which would have resulted in extra latency of up to 1 *BP*.

### B. Transaction Arrival Time (TAT)

The transaction arrival time in the network over both Wi-Fi and cellular networks was measured. The results were compared with the analysed values for all *BP*s under consideration. Fig. 7 shows both the measured and analysed (using equation (2)) transaction arrival times for transaction submission at $t_0$. We only measured transaction arrival time during the steady state because the system only has 16 nodes deployed. For the Wi-Fi results, all *BP*s were almost identical to the values obtained from the analyses. Conversely, the results of the test that was conducted over the 3G network demonstrates the effect of $T_{pd}$ (on average, it was 1.8s (see Table III)). This delay has a major effect on the arrival time, especially when shorter *BP*s are implemented (i.e. 1 second, 2 seconds, and occasionally 3 seconds). From Fig. 7, this becomes clear when the measured values are compared with the analysed values. Based on this, *BP*s of 1 second, 2 seconds, and 3 seconds are very difficult to implement over a 3G cellular network. This is also clearly illustrated in Fig. 8 and Fig. 9; both present the average transaction arrival time for all *BP*s for transaction submission at $t_0$, at random time, and at $t_i$ for both networks.

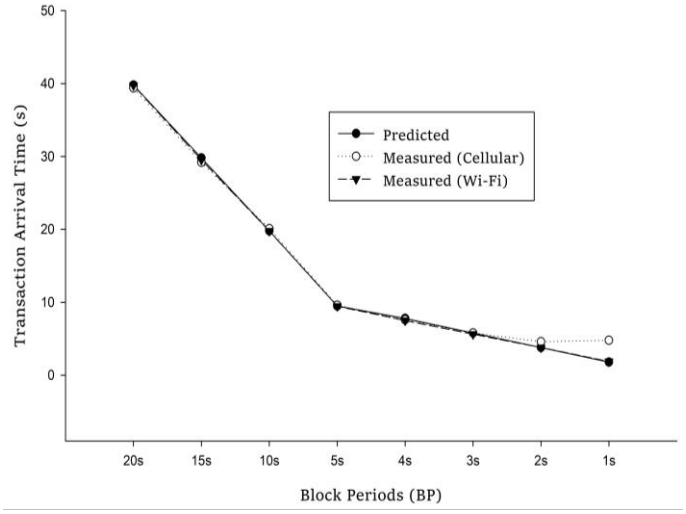



Fig. 7. Measured and Predicted Transaction Arrival Time

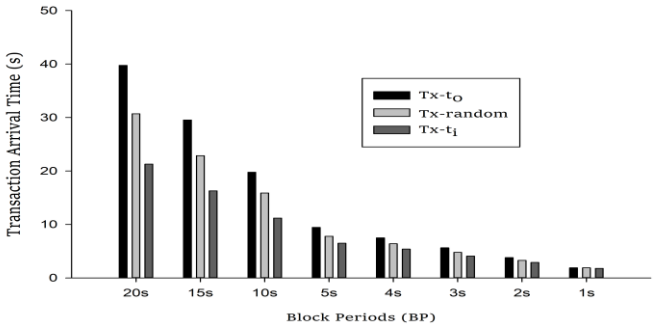



Fig. 8. Transaction Arrival Time over Wi-Fi Network

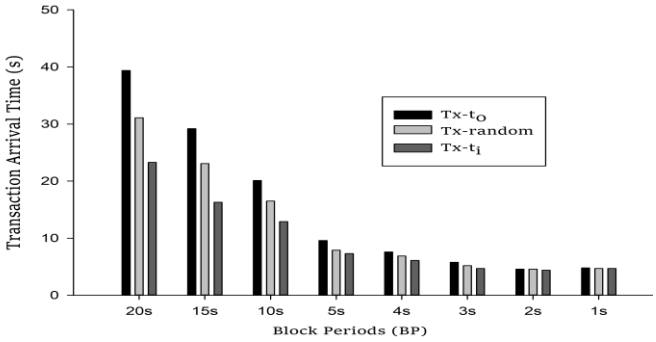
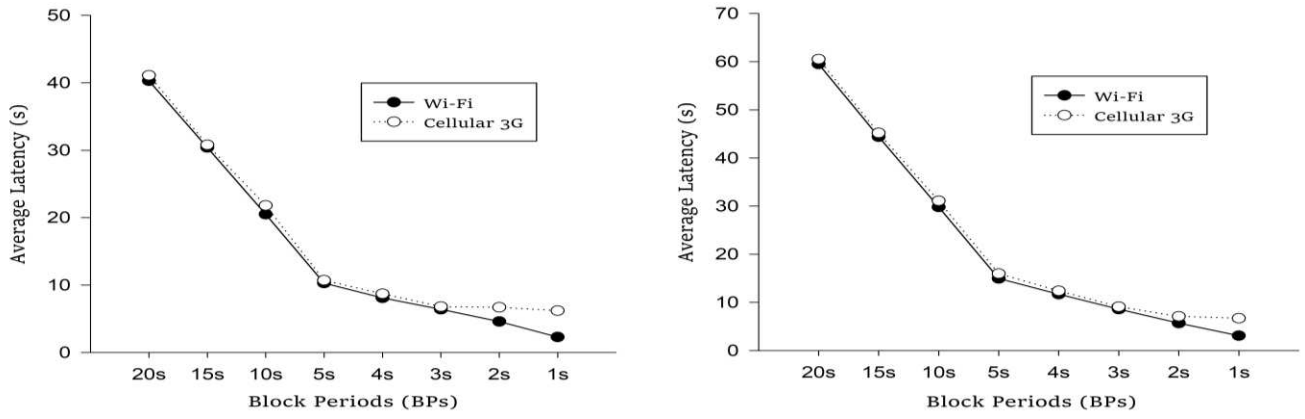



Fig. 9. Transaction Arrival Time over cellular Network

### C. End-to-End System Latency

Ethereum miners add transactions to a block based on the amount of gas the transaction charges. Transactions that charge higher gas have priority to be added first to the block and mined before others. To prevent this from affecting the processing of the system voting algorithm before water level readings are processed, we introduce the measure of submitting water level transactions during the even blocks and invoking the voting algorithm during the odd blocks. This step introduces an extra latency equal to 1 *BP*. The test was conducted over both cellular and Wi-Fi networks for comparison purposes and to determine the effect of using a network with limited bandwidth on the overall latency and network synchronization.

Fig. 10 shows the average latency for all *BP*s implemented. As discussed previously, it is again clear that *BP*s of 1 second, 2 seconds, and 3 seconds cannot be implemented when a 3G cellular network is used. These three block periods will not help with efforts to achieve less latency; in fact, they will simply disrupt the synchronization of the nodes, resulting in more nodes being out of sync with the network, and might cause the execution of the voting algorithm on obsolete water readings. Conversely, the implementation of all *BP*s over Wi-Fi was possible, except for the *BP* of 1 second, which occasionally could not be implemented. Unlike over the cellular network, *BP*s of 2 seconds and 3 seconds were possible to implement, and we achieved less latency. When implementing the *BP* of 1 second, however, there were occasions where the network stability was affected and implementation of a *BP* of 1 second caused the execution of the voting algorithm on water readings submitted by out-of-synch nodes.




a. Tx at $t_i$ b. Tx at $t_0$

Fig. 10. End-to-End System Latency over Wi-Fi and Cellular networks

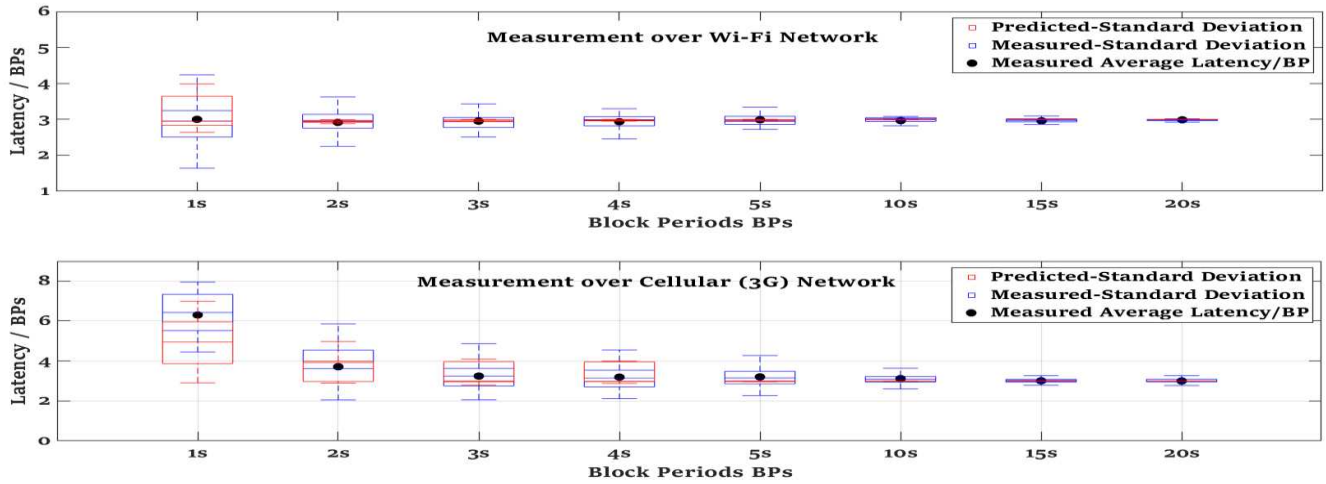


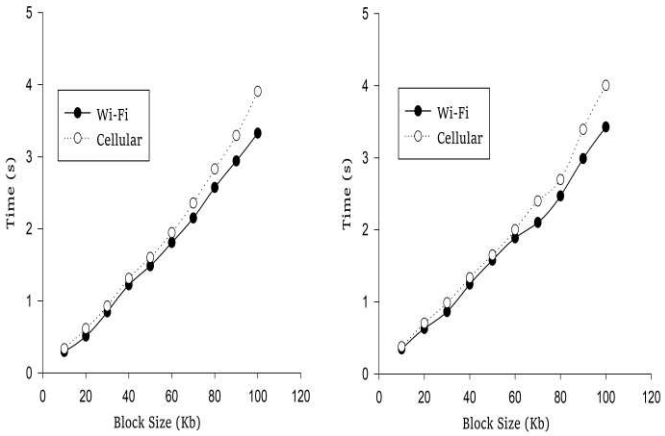

Fig. 11. End-to-End System Latency as Function of the BP (Tx at $t_0$)

### D. Latency as a Function of the Block Period

The network synchronization is the ability of the network to ensure that all water level readings are processed and confirmed by the blockchain network within an acceptable amount of time. This will provide the desired accuracy for the system to monitor and control the water pump. During the steady state of the system, where all arrived transactions are included in the next mined block, the transactions propagation delay has a great effect on the implementation of short *BPs*. Fig. 11 present the predicted and measured standard deviation of the end-to-end system latency as function of the *BP* over both Wi-Fi and cellular. As can be seen in the figure, synchronization and stability of the network were not achieved for all *BP* implementations, especially during testing over the cellular network. This is due to the bandwidth limitation and the increased transaction propagation delay, which sometimes exceeded the *BP*. The 1-second *BP* implementation recorded the highest standard deviation, which rendered the accuracy and the certainty of the voting algorithm poor. The standard deviation decreased, however, as the *BP* increased, making the network more stable, with almost perfect execution of our algorithm. The network has only 16 nodes, each submitting three transactions during each even-numbered block. Within IoT, tens of thousands of nodes could participate in such a network, and this would increase the wait time and the latency. This is one of the limitations of our study: it was not possible to implement thousands of nodes to conduct more synchronization testing. Nevertheless, using our analysis of the system provides a prediction module for the *TAT* during busy periods in the presence of thousands of nodes.

### E. Durations of some Block-Related Events

The durations of block importing, mining, and announcement are affected by the block size. As illustrated in Fig. 12, these durations increased as the block size increased, resulting in the need for more processing time and power to accomplish them. This can be a problem for IoT devices, which have limited computation power, and could also be a problem for the implementation of shorter BPs. The latter issue could result in synchronization problems because as the length of the global chain increases, the length of local copy in the IoT devices will become shorter. This is because nodes are not able to import another block before the release of the previous block; they are therefore not able to catch up with the global chain. This means that the freshness of the data and the current state of the blockchain will become uncertain.




a. Block Importing b. Block Announcement

Fig. 12. Delay when importing and announcing blocks by the nodes

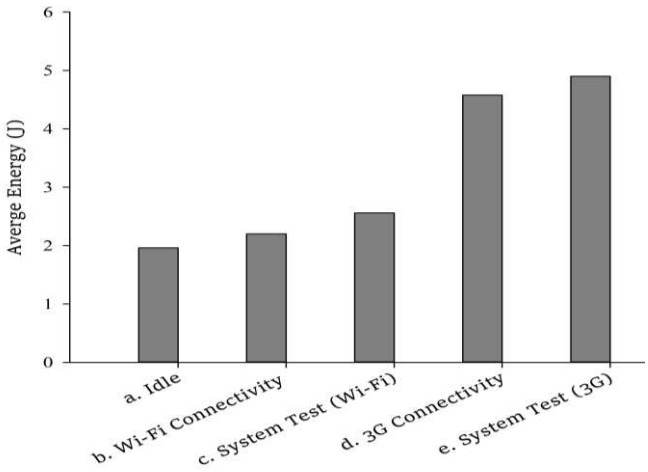



Fig. 13. Average Energy Consumption

### F. Energy Consumption

Ethereum PoA relies on trusted nodes to sign and propagate blocks, and this has a significant advantage in terms of power consumption because nodes do not have to perform any computational work. It is, however, important to characterize our system in terms of power consumption for deployment purposes, where the only source of power might be batteries. We used the Keysight 34450A 5 ½ Digital Multimeter to measure the average current draw by the Raspberry Pi.

First, we measured the average when the Raspberry Pi was idle and converted the average value to an average energy consumption. Subsequently, we measured the average energy for different cases, as shown in Fig. 14. We measured the energy consumption when running the full flood detection system over both Wi-Fi and cellular. During both tests, the node being tested was a fully functioning node. A fully functioning node submits at least two transactions each *BP*, signing and propagating blocks in turn and importing and adding blocks to its local copy. Each test was run for over 30 minutes with 10s as the *BP*, and over 190 blocks were generated and propagated in the network, with different sizes that ranged from 607 bytes to 100 Kbytes.

The results in Fig. 13 (c) indicate that there is a minimum increase in energy consumption of 0.36J (when testing our system over Wi-Fi) compared with Fig. 13 (a) (when the raspberry pi is in idle state). By contrast, the difference between the energy consumption of Fig. 13 (e) (when testing over 3G) and Fig. 13 (a) is more than double (2.95 J); this is due to the power drawn by the Fona 3G board. When all the measurements are analysed, the average energy consumption of running our flood detection system including the Ethereum Blockchain Geth client, regardless of the communication link, is a small amount of energy (around 0.3J). Knowing this result is crucial in selecting which method to use to power the nodes, especially in choosing the right batteries when deploying the system over cellular if no adequate power source is available.

## VII. Discussion

In this work we have studied the performance of IoT-blockchain by providing a system model that predicts the system end-to-end latency and the stability of the network and the nodes synchronization. We validate this study by practically implementing a real world IoT-blockchain application. Our measured results of the latency and network stability are in line with the numerical analyses Based on our tests and analyses, the implementation of BPs of 1, 2, and 3 seconds over a 3G cellular network is not recommended. On Wi-Fi, while it is possible to implement the 1-second BP, it carries a lot of risk in terms of synchronization and data freshness. However, in other application, such as tracking and traceability where the data will not be used to coordinate and automate the decisions, such BPs can be used.

Another aspect to consider when designing a blockchain network for an IoT application is the consensus algorithm. In our study we have chosen Ethereum PoA as our consensus since it does not require any computation works to solve a cryptographic puzzle, which resulted in less energy consumption. However, Ethereum PoA depends on the trusted nodes and their honesty in mining and propagating blocks. This renders it a more central network, which goes against the concept of decentralized blockchain. Many applications within IoT, however, require added security and privacy. Other consensus algorithms such as PoW provides a security consensus when implementing blockchain as public network but requires more energy to find the target hash for each block, and this makes it not an ideal fit with its current form within the IoT realm.

Another finding of our study, is that it is important to consider the size of the block when building IoT-blockchain network. Based on our study the events that are related to the block size such as announcing the block, and importing the block correspond to its size. As the block size increases the time the IoT devices takes to execute them increase as well.

For example, in our study the measured time for importing a block of size 20 kB was about 0.5 s over the Wi-Fi network and about 0.6s over the 3G network, and the time for importing a block of size 100 kB was about 3.3s over the Wi-Fi network and 3.9s over the 3G network. This means more energy consumption and could shorten the life of theses devices' batteries

By studying the implementation of blockchain networks over two different communication links, it is safe to say that Wi-Fi connectivity provides a reliable and fast link, nevertheless it is not available all the time for many IoT applications. In this study we showed the possibilities of implementing blockchain over 3G cellular network, however 4G and 5G networks are better in terms of latency. The authors of [30] who provided a comparison measurements between 3G and 4G that includes one way latency measurement showed that 4G is outperforming 3G in all measured parameters, for example the 4G throughput tests resulted in maximum of more than 28 Mpbs, while the 3g resulted up to 4.8 Mbps. However, 3G provides much larger coverage making this technology difficult to neglect just yet. The 5G technology bring a great potential for IoT-blockchain implementations. Some IoT applications require low latency and higher data rate, which are two strong advantages of 5G, which will help facilitate this integration. IoT, blockchain and 5G together have great potential, while 5G provides a low latency connectivity cover for IoT devices, blockchain can be integrated to eliminate centralized third-party entities and ensures the protection of user and transaction data. This will potentially be a good integration as each part strengthens the other.

In many IoT applications blockchain can provide great benefits, for example to resolve the issues surrounding the use of a central entity for better system performance by eliminating single point failure, and provide means for devices and users identification and authentication and preserve data integrity. These future IoT applications include tracking and traceability within both supply chain and healthcare systems. The latter one can benefit greatly from immutable system such blockchain to protect against medicine and drugs counterfeiting, to monitor the environmental conditions of pharmaceuticals including donated bloods. Also, within industrial Internet of Things blockchain can be utilized for better machine automation - especially ensuring decisions executed by machines are based on true data.

## VIII. Conclusion

In this work, we have provided an analysis of the transaction arrival time in the blockchain network. To validate our analyses, we implemented a real-world IoT-blockchain use case in the form of a flood monitoring and detection system. In our work, we have provided a performance analysis, which included measuring transaction arrival time from submission by the node until the transaction's arrival on the network and measuring the system end-to-end latency for different block periods over a cellular network and Wi-Fi. We studied the network stability and node synchronization for various *BP*s in different transaction submission scenarios. We have also provided a study with a measurement of the average energy consumption, and we have demonstrated that the average energy consumption of running our flood detection system including the Ethereum Blockchain Geth client, regardless of the communication link, is a small amount of energy (around 0.3J).

In this work we showed that blockchain can be integrated into IoT applications, and that Ethereum PoA can be used within IoT for permissioned implementation. We can also conclude that it is important to consider the application requirements, especially in terms of criticality. Also, it is important to consider the type of communication protocol in use and the number of nodes and their locations when deciding which block period and block gas limit to implement.

## References


[1] Ericsson, "Internet of Things forecast – Ericsson Mobility Report,"2019. [Online]. Available: https://www.ericsson.com/en/mobilityreport/internet-of-things-forecast, Accessed on: Jun. 28, 2019.

[2] J. Wurm, K. Hoang, O. Arias, A. R. Sadeghi, Y. Jin, "Security Analysis on Consumer and Industrial IoT Devices", 21st Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 519-524, January 2016.

[3] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," Computer (Long. Beach. Calif)., vol.50, no. 7, p. 80–84, 2017.

[4] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions,"Futur. Gener. Comput. Syst., vol. 29, no. 7, pp. 1645–1660, 2013.

[5] S. Sicari, A. Rizzardi, C. Cappiello, and D. Miorandi, "New Advances in the Internet of Things," vol. 715, no. December, 2018.

[6] N. Kshetri, "Can Blockchain Strengthen the Internet of Things?," Secur. IT, August, pp. 68–72, 2017.

[7] W. S. Stornetta and S. Haber, "How to Time-Stamp a Digital Document," J. Cryptol., vol. 3, no. 2, pp. 99–111, 1991.

[8] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system" in Tech. Rep., 2008.

[9] S. K. Lo et al., "Analysis of Blockchain Solutions for IoT: A Systematic Literature Review," in IEEE Access, vol. 7, pp. 58822-58835,2019. doi: 10.1109/ACCESS.2019.2914675.

[10] Ethereum, "Clique PoA protocol & Rinkeby PoA testnet." [Online]. Available: https://github.com/ethereum/EIPs/issues/225, Accessed on: Jun. 20, 2019.

[11] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with IoT. Challenges and opportunities," Futur. Gener. Comput. Syst., vol. 88, pp. 173–190, 2018.

[12] N. M. Kumar and P. K. Mallick, "Blockchain technology for security issues and challenges in IoT," Procedia Comput. Sci., vol. 132, pp. 1815–1823, 2018.

[13] D. Malone and K. J. O'Dwyer, "Bitcoin Mining and its Energy Footprint," 25th IET Irish Signals Syst. Conf. 2014 2014 China-irel. Int. Conf. Inf. Communities Technol. (ISSC 2014/CIICT 2014), pp. 280–285, 2014.

[14] S. King and S. Nadal, "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake," self-puplished paper, 2012.

[15] N. Houy, "It will cost you nothing to "kill" a Proof-of-Stake crypto-currency [v.0.1]," 8th Eur. Conf. Antennas Propagation, EuCAP 2014, pp. 741–742, 2014.

[16] S. Popov, "The Tangle," IOTA_Whitepaper. pdf, 2018.

[17] Ethereum, "What is Ethereum? — Ethereum Homestead 0.1 documentation." [Online]. Available: http://www.ethdocs.org/en/latest/introduction/what-is-

ethereum.html. Accessed on: Jan. 24, 2019.
[18] A. Dorri, S. S. Kanhere, R. Jurdak, P. Gauravaram, "Blockchain for iot security and privacy: The case study of a smart home", Pervasive Computing and Communications Workshops (PerCom Workshops) 2017 IEEE International Conference, pp. 618-623, 2017
[19] A. Dorri, S. S. Kanhere, R. Jurdak, Blockchain in internet of things: Challenges and solutions, 2016.
[20] G. Sagirlar, B. Carminati, E. Ferrari, J. D. Sheehan, and E. Ragnoli, "Hybrid-IoT: Hybrid Blockchain Architecture for Internet of Things - PoW Sub-blockchains," pp. 1–10, 2018.
[21] S. Huh, S. Cho, and S. Kim, "Managing IoT devices using blockchain platform," Int. Conf. Adv. Commun. Technol. ICACT, pp. 464–467, 2017.
[22] A. Bahga and V. K. Madisetti, "Blockchain Platform for Industrial Internet of Things," J. Softw. Eng. Appl., vol. 09, no. 10, pp. 533–546, 2016.
[23] S. Ibba, A. Pinna, M. Seu, and F. E. Pani, "CitySense: Blockchain-oriented Smart Cities," ACM Int. Conf. Proceeding Ser., vol. Part F1299, 2017.
[24] A. Jindal , G. S. Aujlab , and N. Kumar, "SURVIVOR: A blockchain based edge-as-a-service framework for secure energy trading in SDN-enabled vehicle-to-grid environment," Comput. Networks, vol. 153, no. 2019, pp. 3–17, 2019.
[25] R. Chaudhary, A. Jindal, G. S. Aujla, S. Aggarwal, N. Kumar, and K. K. R. Choo, "BEST: Blockchain-based secure energy trading in SDN-enabled intelligent transportation system," Comput. Secur., vol. 85, pp. 288–299, 2019.
[26] P. K. Sharma and J. H. Park, "Blockchain based hybrid network architecture for the smart city," Futur. Gener. Comput. Syst., vol. 86, no. 2018, pp. 650–655, 2018.
[27] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," IEEE Access, vol. 4, pp. 2292–2303, 2016.
[28] James Ray, "Light client protocol · ethereum/wiki Wiki." [Online]. Available: https://github.com/ethereum/wiki/wiki/Light-client-protocol. Accessed on: May, 13,2019.
[29] B. Ford and P. Srisuresh, "Peer-to-Peer Communication Across Network Address Translators." [Online]. Available: http://bford.info/pub/net/p2pnat/. Accessed on: Jul, 15,2019.
[30] F. Fresolone, R. Kloibhofer, A. Ralbovsky, P. Farkas, M. Rakus, and T. Palen, "Throughput and One-Way Latency Measurements in a 3G/4G Live-Network Hi-Mobility Uplink," 2016 39th Int. Conf. Telecommun. Signal Process. TSP 2016, pp. 44–49, 2016.