Iraqi Journal of Information and Communications Technology(IJICT)
Vol. 3, Issue 4, September 2020

ISSN:2222-758X

CONSENSUS ALGORITHMS BASED BLOCKCHAIN OF THINGS FOR DISTRIBUTED HEALTHCARE

Istabraq M. Al-Joboury [1], **Emad H. Al-Hemiary** [2]
[1,2] College of Information Engineering, Al-Nahrain University, Baghdad, Iraq
{estabriq-94, emad}@coie-nahrain.edu.iq [1,2]
Received:24/08/2020, Accepted:26/10/2020

***Abstract*- The Internet of Things (IoT) consists of smart Things with the evolution of ubiquitous computing. Fog Computing (FC) processes and analyzes data of these sensors near to users. However, the ever-increasing number of Things and the consequent explosion in data traffic has led to fail traditional solutions of centralized storage. Blockchain is a new technology developed as a shared ledger build around a peer-to-peer network to produce unchangeable blocks that contain multiple data. These blocks are linked to previous ones in a sequence called chain through hash functions. Participants in blockchain selects a leader through one of consensus algorithms who adds new blocks in the chain to prevent dishonest nodes from creating invalid blocks. In this paper, we propose IoT based blockchain architecture named blockchain of Things to store medical records in a distributed manner. The architecture is emulated on Fog server Linux-based using Node.js and Postman. Three consensus protocols, namely: Proof of Work (PoW), Proof of Stake (PoS), and Byzantine Fault Tolerance (PBFT) are emulated and communicated using WebSocket. The consensus algorithms are evaluated in terms of CPU utilization and memory usage. The results show that PoS compared to PoW and PBFT is more lightweight and beneficial to IoT application.**

keywords: **Blockchain, Internet of things, Consensus algorithms, PoW, PoS, PBFT.**

I. INTRODUCTION

Smart sensors (also called Things) can generate and collect data from environments, communicate with each other, and share information to increase the overall efficiency of applications. These sensors in the Internet of Things (IoT) with affective computing and networking capabilities can be controlled by anyone, anywhere, and anytime through the Internet. The main idea of IoT is to make a new digital world integrated into a physical world based on machine- to- machine or device- to- network to make life easier and more comfortable. IoT has three basic layers: the first layer called the Things layer is responsible for measuring values from sensors/actuators. While the gateway layer transmits data from Things layer to upper layer for further processing [1]. Cloud Computing (CC) handles data for analysis and provides a central storage and it can be considered as application layer. The number of Things is expected to reach to billions in the next few years, IoT may face new challenges like scalability, security and privacy issues. The classical solutions like CC may not be suitable with the expected huge volume of data because of the centralization structure. Fog Computing (FC) is a new concept developed to cope with the new challenges and the rapidly evolving in IoT. The main objective of FC is to provide the same services of traditional CC efficiently way and near to Things [2]. The rapid evaluation of IoT has opened up with new opportunities in the medical Things and especially for monitoring heath data of patients. Patients wear various medical Things that measure real-time data such as heartbeat. Gateway devices transmit data to healthcare providers for remote control and monitoring to reduce the time-consuming by doctors for diagnosing and analyzing diseases. However, healthcare providers concern about reliability and privacy of sensitive data of patients [3]. A blockchain is a new technology initially proposed by Satoshi Nakamoto, consists of a sequence of small databases called blocks linked by a cryptography and links between each two consecutive blocks called chain. Utilizing blockchain in IoT can provide immutable, proof of

tamper, security and efficiency of distributed storage. The key feature of blockchain is there no single point of failure like in CC or FC since data from Things can be shared directly among all other Things in peer- to- peer. The main difference between the traditional database and blockchain is that there is no central third party and data in blocks are immutable which means no one can delete or modify it. The integration between IoT and blockchain appears very useful because of new features can be added to IoT like local processing, no single point of failure, distributed sharing, and immutable data. Thus, can improve healthcare applications like enhances the transmission of patient's history and increases the security and privacy of health records [4]. Each node broadcasts the received data to all neighbor nodes connected to it, until all nodes in the network will receive the data. Data are stored locally in pool sometimes called memory pool (mempool). Nodes gather data from pool to create blocks and in order to add that block in the chain, the other nodes need to reach a consensus using one of various consensus algorithms due to avoid attacks and conflicts. These consensus algorithms can be divided into two types: lottery-based and voting- based, it will be discussed in details in the following sections [5]. The main aim of this paper is to propose blockchain with IoT named blockchain of Things to facilitate the analysis and monitoring of health data collected from medical Things and to keep track of history, measurements, and treatments of patients. Blockchain is implemented based Fog architecture in order to reduce latency of blocks transmission between Fog servers. Our proposed architecture utilizes a private blockchain to mitigate the cost and enhance the privacy of patients. Also, the architecture presents three scenarios with different most consensus algorithms used to show how to apply these algorithms in IoT for healthcare applications. The rest of this paper is organized as follows: Section II reviews the blockchain concept. Section III, covers three of consensus algorithms. Section IV, discusses WebSocket protocol. Section V, describes problem statement. Section VI, proposes blockchain of Things architecture. Section VII, shows the results of performance evolution. Finally, section VIII concludes this paper.

II. BLOCKCHAIN TECHNOLOGY

Blockchain is a concept of distributed ledger which represents a small- size database spreading across the whole network of devices and users. The main advantage of blockchain is that participants save a copy of full digital ledger after check if it is valid by using one of the consensus algorithms. Blockchain eliminates the concept of central record keeping. For example, when Alice wants to send a message to Bob, a message is created inside a block by Alice. If this block is valid by using verification process, then the block is added to blockchain, subsequently block is spreading across the network in peer-to-peer. Bob updates his blockchain to reflect changes. Finally, Bob receives the block from Alice [6]. Set of transactions are created and bundled together to create the candidate block after a specific time and stored in chronological order. Those transactions can be tracked by members without third party. Blocks are consecutively connected via cryptographic chain so that it is called blockchain. Blocks are added to the chain by nodes called miners where multiple nodes participate to verify that block by process called mining. The mining technique is designed for power- consumption and limited recourses [7]. Each block is linked to the previous block (also called parent block) through the hash value of parent block except the first block is sometimes referred to as a genesis block having no parent block. For instance, block I points to block i-1 through a single unique hash value of parent block called inverse reference as shown in Fig. 1 [8]. Since the genesis does not have previous block, the hash value of parent block is initialized to dummy values. If two blocks are received at

the same time, then forks will be created. In this situation, the longest valid chain is considered as the valid chain and the other chain is often referred to orphan blocks and then will be discarded. A block structure contains of both header and body. In details, a header consists of the following fields [9]

1) Version (4 bytes): there are three versions of blockchain, namely: 1.0 currency, 2.0 smart contract, and 3.0 Distributed Applications (DApps).
2) Hash of previous block (32 bytes): depends on hash function tools like SHA- 256 in Bitcoin or keccak- 256 in Ethereum. A Hash function maps any size data to fixed-size output (Bitcoin and Ethereum are platforms).
3) Merkle tree root hash (32 bytes): hash of transactions located inside the block so that malicious cannot be changed or modified unless the attacker changes the header.
4) Timestamp (4 bytes): represents the time of the creation of blocks in seconds. In Bitcoin, miners reject blocks with a timestamp for more than 2 hours.
5) Nonce (4 bytes): is any number starting from zero and increases to modify the header for the validation process.
6) Difficulty Target (4 bytes): threshold for validation process used by miners

While body f block consists of two fields as follows:

- Data Counter (1∼9 bytes): is number of transactions in a single block.
- Data List (up to 1 MB): arbitrary transactions depend on block size and size of a single transaction.

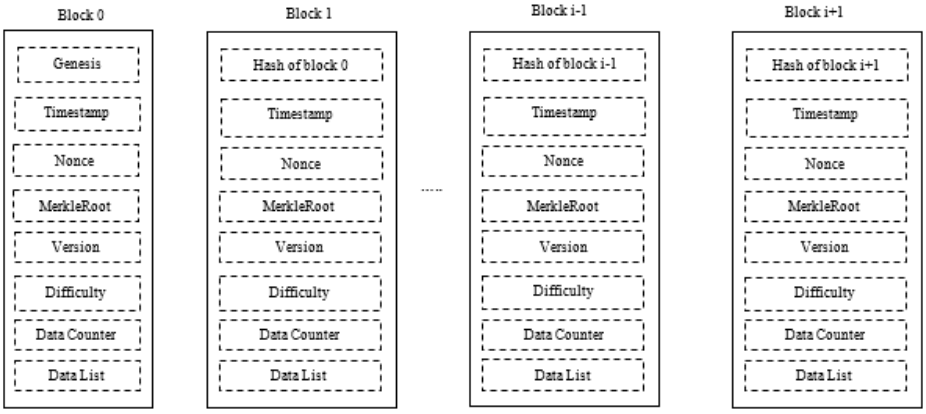

Figure 1: Blockchain structure

III. Consensus Algorithms

This section presents an overview of three of consensus algorithms to provide security and fairness. The goal of these algorithms is to validate, accept, and add blocks to the next index of chain by making all participant nodes reach a consensus/decision for the correct block. Without consensus, every node must have the same probability in the case of random selection to be picked as a creator of the block. However, attacker may join with a large number of nodes to increase the chance to be selected, then can control and manipulate the network. To avoid this problem, various consensus algorithms have emerged [10] and the comparison between them is described in Table I

A. *Proof of work*

Proof of Work (PoW) is the first lottery-based consensus algorithm employed by blockchain such as Bitcoin, Ethereum, and many of blockchain networks today. PoW has introduced long before blockchain to reduce spam emails attack [11] and Sybil attack in which participates perform a computational resource- intensive task. The main idea of PoW in the blockchain is to validate blocks through a process called work/mining which requires massive computing capacities. The responsibility of miners using PoW is to accept or reject blocks and to provide a validity check. Miners are able to add new blocks in the chain by solving computationally expensive task (sometimes called puzzle). All miners compete to find solution to a pre- defined puzzle, and the fastest one that solves the solution can create a new block and get rewarded [12]. Miners have to change an arbitrary number (called nonce) constantly until the solution is found and produce a valid block which requires a lot of computational power as in Equation 1

$$H(n\|H(b)) \leq M/D \tag{1}$$

Where H: is a hash function with variable numbers range $[0, M]$, D: is difficulty described in the next lines, and b: is contents of current block. The work steps of PoW are depicted in Fig. 2. The solution is to find the correct hash of header of certain block begins with a certain number of zeros (called difficulty). Then, the output is compared with target to determine its validity. Target is determined by blockchain and is automatically adjusted depends on the last two blocks, if they were being mined faster or slower than 10 minutes on average as in Equation 2.

$$Target = Actual\,Time\,between\,Blocks/Expected\,Amount\,of\,Time \tag{2}$$

Finally, others miners can verify the solution easily in order to reach a consensus. Thus, PoW- network improves security because the consumption of resources increases the cost of selection of miners [13].

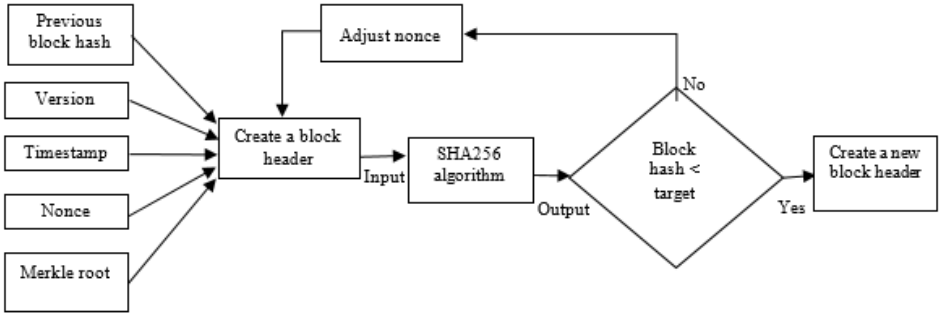

Figure 2: PoW flow

B. *Proof of stack*

Proof of Stake (PoS) is an energy- saving and lottery-based consensus algorithm replaces nonce in PoW with stake held to produce blocks quicker and more transactions per second. Miners (called validators in PoS) must prove the ownership of

the amount of stake (e.g. , currency). It suggests purchasing cryptocurrencies instead of buying devices with high properties like extremely high resource consumptions to win in the competition of creating blocks [11]. PoS divides nodes into two types: node does not own currency to held stake, it cannot be involved in process of selection validator. While, nodes own currency, they enter a race based on stake calculation using balances or deposits and the node has more stake is more likely to be selected as a validator as illustrated in Fig. 3. If the node is found guilty, then all the currency that is putted as a stake will be taken. However, this case is quite unfair because the richest node can always be selected to create blocks and control the network. To address this issue many solutions are proposed: one solution has been suggested to choose the next leader in random manner [14]. Others propose to find a solution to puzzle such as finding the lowest hash according this equation 3

$$SHA256\,(Previous\,hash\,block, Version, Timestamp, Merkle\,root, address\,of\,validator) < Target * Coin \quad (3)$$

Instead of adjusting the nonce for many times, the node needs to find hash with the specific amount of currency which requires less computational power. Another solution is to use coin- age based selection as in Peercoin platform as shown in Fig. 4. The older coins have a higher chance to be the validator of next block. For example, if the node puts 20 coins for 30 days, then the coin- age of the node will be 600. After the node creates a new block, the coin-age rests to zero [15].

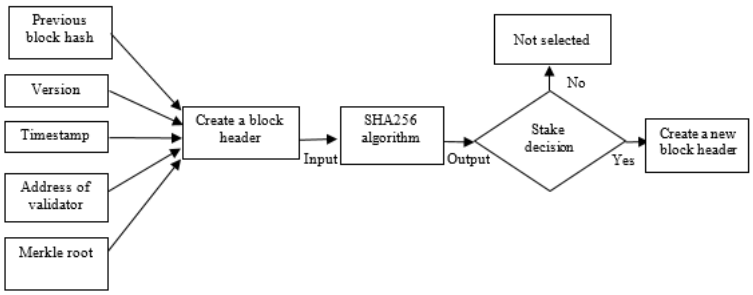

Figure 3: PoS flow

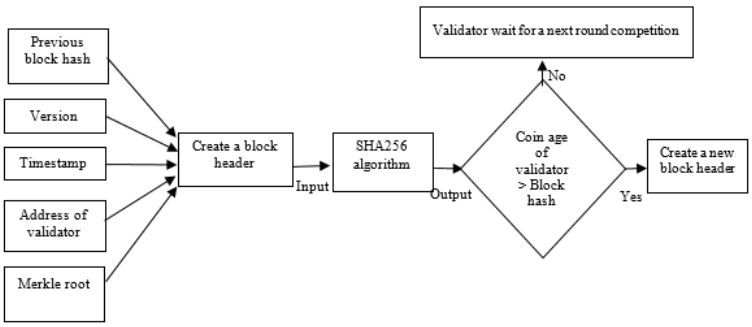

Figure 4: PoW flow (Coin- Age)

ISSN:2222-758X

C. Practical byzantine fault tolerance

Practical Byzantine Fault Tolerance (PBFT) is a replication algorithm proposed by Miguel Castro and Barbara Liskov as a solution of the Byzantine Generals Problem: when the Byzantine army is divided into different groups, each group ordered by a leader. These Byzantine groups want to attack the castle; however, the attack will only succeed if all groups make the same decision at the right time. To makes this happen, leaders have to reach a decision whatever to attack or not. Some of leaders are traitors and wish to reach to a disagreement. The primary purpose is to success the attack even in the presence of traitors [16]. PBFT is intended for private networks where node is identified and known to all other nodes in the network. In PBFT, all nodes have equal rights and there is no mining. It is a voting- based consensus means that nodes need to vote to reach a consensus and exchange their votes between each other in peer- to- peer. The network works according to PBFT a leader- based, there must be more than 2/3 valid nodes of the total nodes support the leader, while could be 1/3 malicious nodes. It has two type of nodes: the primary nodes are picked as a leader in round-robin (circular) order to create candidate blocks. The secondary nodes send vote to accept or reject on that candidate blocks. In each round, there are five phases as can clearly see the follow in Fig. 5 request, pre- prepared, prepared, commit, and reply. The pre-prepare phase begins by primary node broadcast pre- prepare message to all secondary nodes. Once node receives the pre-prepare message, it enters the prepare phase and broadcast prepare message to the rest nodes including the primary node. The two phased pre- prepare and prepare are used to order messages. Each node receives 2f prepare messages compared to the pre- prepare messages where f is the number of Byzantine nodes, then the nodes enter to commit phase and broadcast the commit message to other nodes. Each node must receive 2f+1 commit message that match the pre- prepare message. Both phased prepare and commit are for ensuring that all messages are ordered among all nodes [9]. This algorithm is considered as a communication- intensive because it places overhead of handshake messages between phases to maintain a healthy primary node. For instance, if there are five nodes: client, primary (leader) node, and three secondary nodes. The leader forwards the message from client to the three nodes. If the node 3 is crashed then one message pass all the phased to reach a consensus. At the end, client receive a reply message from the nodes to reach a consensus in that round as it can be seen in Fig. 5 [17].

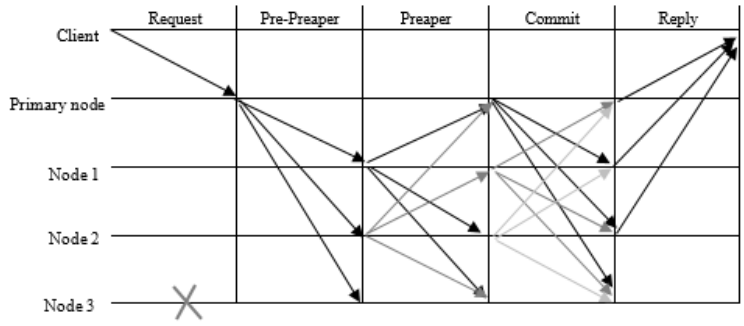

Figure 5: Process of PBFT

TABLE I
PROPERTIES OF POW [13]

Property	PoW	PoS	PBFT
Type	Probabilistic- finality	Probabilistic- finality	Deterministic- finality
Fault tolerance	<50% computing	<50% stake	<1/3 voting nodes
Power consumption	Large	Partial	Low
Platform	Bitcoin	Ethereum	Hyperledger
Application	Public	Public	Private
Node identity	permission- less	permission- less	permissioned
Transactions per second	7∼ 30	30∼ 100	100∼ 2000

IV. WEBSOCKET PROTOCOL

WebSocket protocol is an application OSI layer designed over TCP, especially for real- time web applications, and does not follow request/response model as in Hypertext Transfer Protocol (HTTP). It opens a bidirectional and full-duplex data transmission, once client sends message to server without closing the connection as long as both client and server are actives [18]. It reduces the communication overhead of HTTP that is open TCP socket each time client sends a request which requires high number of round- trips between client and server. The Fig. 6 emphasizes the handshake between client and server using WebSocket protocol. Firstly, the protocol has to establish a TCP connection with 3- way handshake to ensure reliability and scalability. Both client and server have now a direct access each other. Then, they can exchange WebSocket Upgrade Request and Response messages to open a session. Finally, the client and server can transmit data unless one of them closes the connection. WebSocket has a small header size range from (2 up to 14) bytes long prepended to the payload data [19].

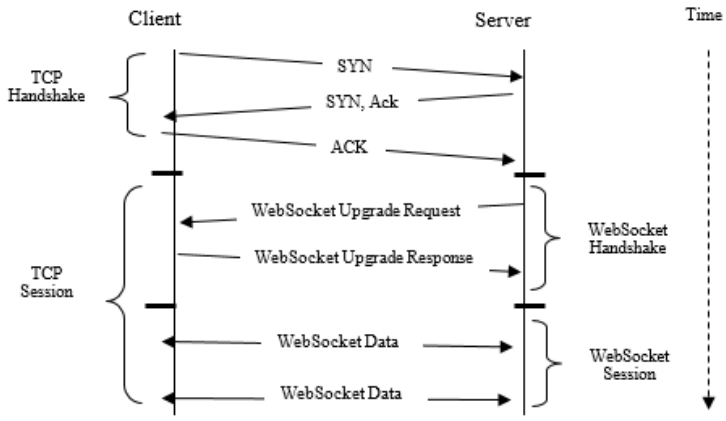

Figure 6: WebSocket handshake

V. PROBLEM STATEMENT

IoT connects Things around the world together to provide information of a common application, these Things are characterized by being smart, and can be controlled, automated, and monitoring from anywhere. However, Things have limited capabilities of computing, processing, and storage, also it is prone to attack. CC can provide high level of resources to IoT, but because of its physical location is far away from Things can result a high response time. Therefore, FC has

been proposed to bring all Cloud capabilities like processing and storage near to Things which causes a low latency and to offload IoT resource-restrained to Fog servers. In addition, FC has own limitations like it is a single point of failure that any abnormal behavior may lead to crash the whole system [20]. The researchers of the present paper on the one hand, intend to: integrate IoT with blockchain to strength the performance of IoT network in terms of efficiently, reliability, scalability, maintain high security standard, and to facilitate data transmission in peer- to- peer between Fog servers. On other hand, they intend to enable blockchain on FC as an attempt to support distributed Fog server with lower delay and become less susceptible to potential malicious attacks. By integrating the three technologies; IoT, blockchain, and Fog can complement each limitation of other by validating the accurate data using one of consensus algorithms. Blockchain can be very useful in healthcare applications because of inherent characteristics, particularly for exchanging and sharing of medical records and history. For instance, the author in [21] proposed a blockchain system named MedRec based on public Ethereum platform that creates electronic medical record. Patients can control their records. However, public ledger like Ethereum requires large number of fees compared to a small- size of IoT data and can anyone join the network with permission- less; while this paper utilizes a private blockchain with permissioned patients.

VI. Blockchain of Things Architectures

This section proposes three scenarios of IoT based blockchain architecture termed as blockchain of Things for data sharing healthcare application. Each scenario uses a different consensus algorithm, namely: PoW, PoS, PBFT. The proposed architecture based on FC to distribute and share medical records upon physicians with minimum delay. The proposed architecture consists of three layers: Things layer contains data generated from IoT wearable sensor on patient body. These data can be monitored by doctors and healthcare providers, then data are transferred to application layer by access point gateway. The application layer (called Fog layer) is responsible for storing data in blockchain located on Fog servers as shown in Fig. 7. The scenarios are emulated on Fog server 8 Core with 3 GB of dynamic memory and 8 GB of permanent storage. Fog server runs on Ubuntu server version 18.0.4 LTS Operating system (OS). VMware Workstation Pro version 15.5.2. used for virtualization to create multiple servers inside physical Fog server and Postman version 7.25.0 utilized for building APIs; both software are installed on Fog server.

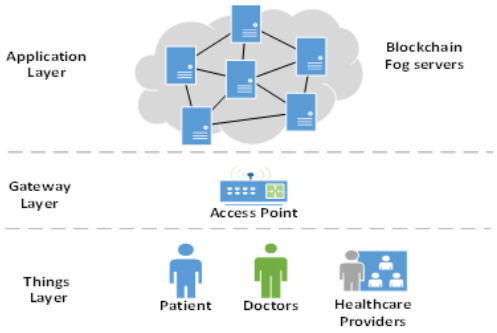

Figure 7: Blockchain of things architecture

ISSN:2222-758X

A. Blockchain of things based PoW

The first scenario of blockchain of Things architecture using PoW is described with the following characteristics: each patient has a profile to store medical records and uses asymmetric cryptography to facilitate encryption and decryption on arbitrary medical record. Profile uses unique public/private keys, which private key is used for signing data and public key is used for verifying signatures. The proposed architecture shares the public key with everyone in the network, while keeps the private key in secret. Standards for Efficient Cryptography 256bits (secp256k1) of Elliptic Curve Cryptography (ECC) is used to generate keys that is mathematically impossible to revert the keys. Patients and doctors have their own IDs (PatientKey) and (DoctorKey) respectively. Medical records can be formulated as JavaScript Object Notation (JSON): SensorData = timestamp, SensorValue, PatientKey, DoctorKey. Each medical record has a unique ID using Universally Unique Identifier (UUID) version 1 with 128 bits created by Microsoft. The JSON data is signed by using patient private key at certain time (timestamp) to ensure the integrity of that data has not been tampered. Then, Things broadcast the JSON data using WebSocket protocol to all miners in application layer (also called Fog layer) through access point gateway via peer- to- peer. This scenario contains three Fog servers acting as miners that are responsible for exchanging and sharing values of IoT sensor inside blocks between each other. This scenario is emulated in private network using Node.js and Postman. New miner can connect to blockchain by running the following Command Line Interface (CLI) on terminal:

HTTP-PORT=3003 P2P-PORT=5003 PEERS=ws://localhost:5001,ws://localhost:5002 npm run dev

For instance, miner 3 wants to connect to the two existing neighbors using two ports, one for HTTP protocol and the other for WebSocket protocol. Miner broadcasts the received JSON data from patient's profile to all its neighbors. Miner checks integrity and authenticity of the unconfirmed data. If it is valid, then miner stores data temporary on pool named mempool. Miner takes chunks of the confirmed data from pool and organizes them as Merkle tree structure. As the name suggest, blockchain consist of sequence of blocks connected through chain which starts with genesis block. The genesis is a starting point of blockchain programmed as a hardcoded with dummy values with no previous hash and empty data. While, blocks contain the following fields: timestamp (time of block creation in seconds), lastHash (hash of previous block), hash (hash of current block), data (array of sensor values), nonce (arbitrary number), and difficulty (specified by blockchain network to ensure block interval is 10 minutes). Secure Hashing Algorithm with 256-bit (SHA256) is a one- wat hash function used to hash the contents of block to produce a unique 32- byte output in order to preserve the integrity and immutability of each block. Therefore, these blocks cannot be modified unless changing all blocks in the blockchain, thus it is very expensive task. If a malicious node wants to tamper a single bit inside any block, then it will produce a completely different hash and it can be easily detectable and no longer be valid. The hash is used as pointer to explicitly reference the previous block and this process repeated each time with limited amount of data. Healthcare providers set difficulty to 4 and miners have to find nonce value that leads to 4 zeros at the beginning of hash of the current block. Difficulty is automatically adjusted so that new blocks are created on a steady interval around 10 minutes. The fastest miner that finds the solution, it will broadcast the block to the rest of miners using WebSocket protocol in peer-to-peer network. The researchers of this paper modifies PoW to fee-less because miners handle medical records not cryptocurrency. Once miners receive the new block,

ISSN:2222-758X

they have to reach a consensus to accept to reject according some rules:

1) check the genesis block is valid or not,
2) check the current hash is valid by matching with the previous hash lastHash, and
3) check the longest chain.

If two miners create new block at time, then a conflict will be created that sometimes-called fork. Simultaneously, one of the miners produces another new block and has the longest chain; and the other chain will be discarded.

LISTING 1: EXCEPT OF A JSON OF BLOCK IN BLOCKCHAIN OF THINGS

```
{
    "timestamp": 1589875796324,
    "lastHash": "c671c846.....",
    "hash": "000066b3.....",
    "data": [
        {
            "id": "8622018a....",
            "SensorData": [
            ]
                {
                    "timestamp": 1589875537706,
                    "SensorValue": "80",
                    "PatientKey": "04e6821b....",
                    "DoctorKey": "04d03ca0....."
                }
        }
    ],
    "nonce": 4774,
    "difficulty": 4
}
```

B. Blockchain of things based PoS

PoW consensus algorithm provides a high level of security by letting miners compete between each other on solving a complex task that requires a lot of time and energy. It prevents some devices from participate to create blocks because of their limited resources. To overcome these restrictions, PoS is proposed to improve the architecture. The details of profiles, public/private key pair generation, and generation of sensor data are the same as in the previous scenario. Block validation process of proposed architecture is carried out according to the algorithm emphasizes in Fig. 8. Fog servers are required to put a certain amount of cryptocurrency or called stake/fee in stake pool to participate in the selection process. Each validator connects to its neighbors using WebSocket protocol. Validators receive JSON data from Things and broadcast every data received from patient to other validators in peer-to-peer using WebSocket protocol. They have to check some rules before adding it to mempool as the same in PoW scenario. Validators pick one of them as a leader based on random selection to produce new block and broadcast it to the rest of validators using WebSocket. Other validators append the received block after checking if the block is valid or tampered according to rules as the same as in PoW scenario. Chain validation and replacement are the same in PoW scenario.

C. Blockchain of things based PBFT

The final scenario follows the same steps of the previous scenarios in A, and B except the following changes: six virtual Fog servers are simulated using Node.js to solve Byzantine General Problem. This architecture assumes one of the servers is a dishonest server that can prevent the network from reach a consensus or can lead to a false decision. Therefore, the

architecture is programmed to sustain the network and to reach a correct consensus even if a malicious node joins the network on one condition the number of dishonest nodes must be less than 1/3 of total honest nodes. Under this architecture, one primary server is chosen to be a responsible for creating a new block in round robin algorithm and the rest nodes are holding as backups for the leader in case of any failure would happen. Then, all virtual Fog servers can communicate with each other and exchange messages via five phased as described in section 3.C.

VII. RESULTS

The performance evaluation in term of Central Processing Unit (CPU) utilization and memory for the proposed blockchain of Things architecture are presented. To demonstrate the evaluation graphs, 100 data per second with the same type and size of virtual sensor are emulated using Node.js and Postman tool. Fig. 9 shows that CPU utilization of the proposed architecture for under normal condition and three scenarios using PoW, PoS, and PBFT.

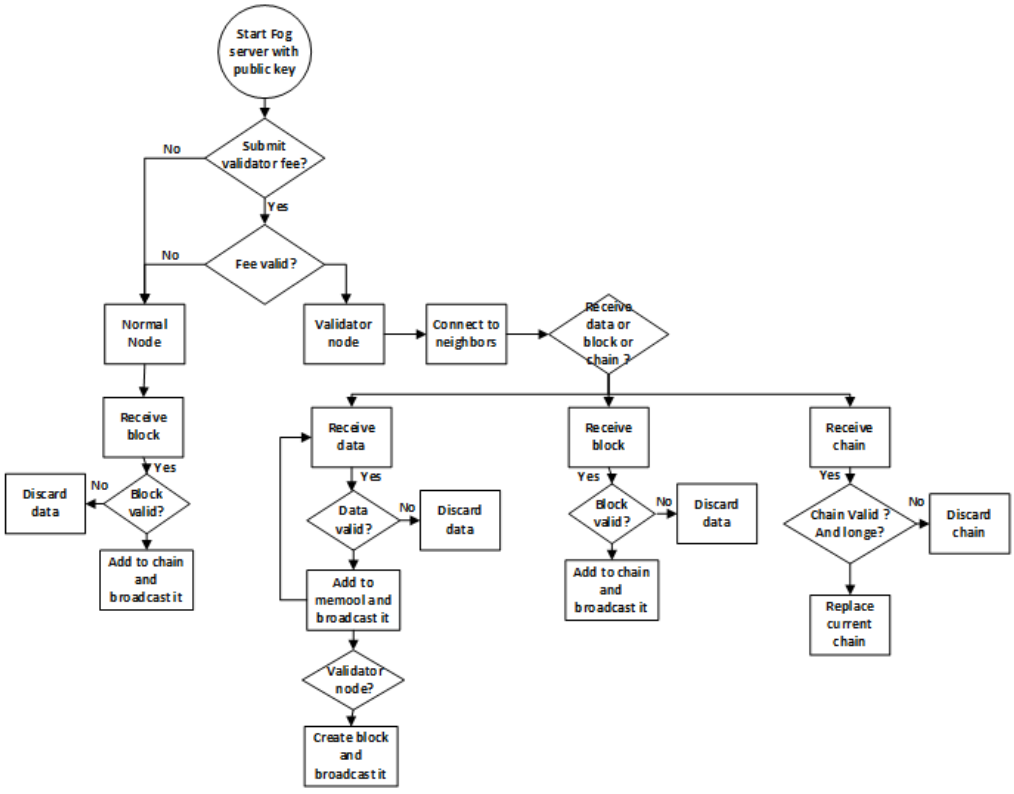

Figure 8: PoS algorithm

It can be noticed that PoW consumes the maximum CPU among the other consensus protocols because of its resources-intensive task which greatly limits the system with capabilities of servers. PoS and PBFT can greatly reduce the CPU computations because there is no mining. Fig. 10 illustrates the memory usage in percentage of three censuses protocols.

PoS has the same performance as the ideal system; whereas PBFT has the higher memory usage than PoW and PoS because PBFT requires each node sending messages to the neighbors in each round. As a result, PoS can be beneficial and lightweight solution in IoT application because of its limited- resources.

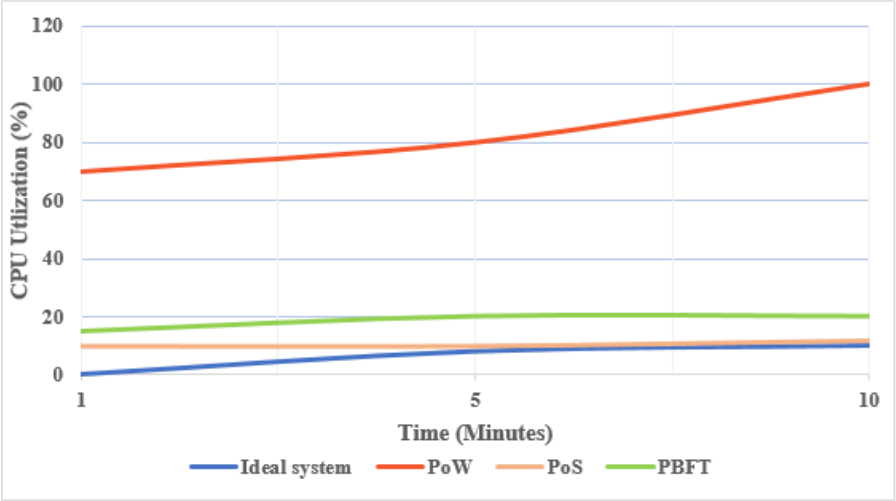

Figure 9: CPU utilization of proposed architecture

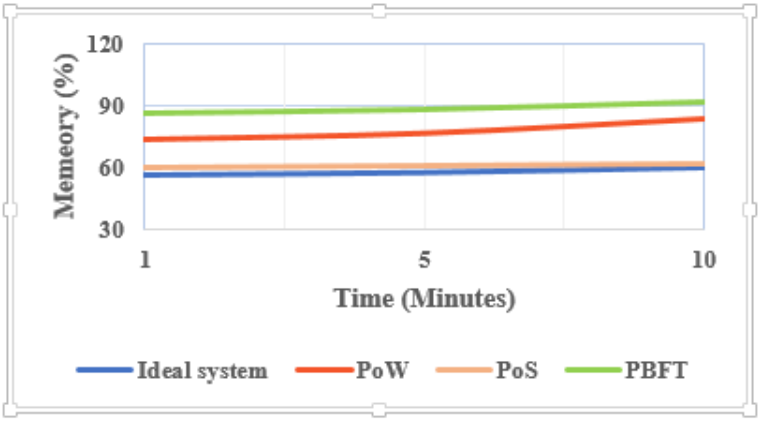

Figure 10: Memory of proposed architecture

VIII. Conclusions

In this paper, we propose to integrate blockchain with IoT named blockchain of Things based Fog layer for healthcare applications to remove the use of centralization of classical CC which makes the whole system crashes at any failure. The proposed architecture consists of three layers: Things layer generates data from virtual wearable sensors on patients to broadcast it in peer-to-peer using WebSocket protocol, gateway layer transmits data from Things to upper layer using WiFi,

ISSN:2222-758X

and application layer that receives data and stores it in blockchain. The architecture presents three scenarios, with each scenario uses one of the most used consensus protocols: PoW, PoS, and PBFT. The three scenarios are emulated on Ubuntu Fog server acts as private blockchain network and programmed using Node.js with the help of Postman. The architecture shows it is possible to implement blockchain on IoT to provide distribution, scalability, security and privacy. PoW and PoS are developed to be fee- less to send and receive data by patients and doctors. Utilizing private blockchain in IoT mitigates the network from attacks like cyber-attacks and prevents unknown users from creating invalid blocks. Malicious nodes can easily be detected and rejected from the system. In PoW, the attacker must have the greatest computational power of all miners to control the network, thus the attack will not be easy or even impossible. Whilst, the attacker in PoS requires to have the majority of stakes in the network, therefore the attack can be extremely expensive. Also, traitors in PBFT need to be more than 1/3 of total nodes in order to control the network. The performance evaluation in the matter of CPU utilization and memory usage, PoS has almost the same performance as in ideal system which makes PoS more crucial for IoT applications. PoS increases the performance of IoT in terms of distribution, storage, sharing information, and security with less of power consumption compared to PoW and PBFT.

REFERENCES

[1] S. El Kafhali, and K. Salah, "Performance modelling and analysis of Internet of Things enabled healthcare Monitoring Systems" , IET Networks 8, no. 1, pp 48- 58, 2018.
[2] J. Wang, K. Han, A. Alexandridis, Z. Chen, Z. Zilic, Y. Pang, G. Jeon, and F. Piccialli, "A blockchain- based eHealthcare system interoperating with WBANs" , Future Generation computer systems, 2019. DOI: 10.1016/j.future.2019.09.049.
[3] A. Benrazek, Z. Kouahla, B. Farou, M. Ferrag, H. Seridi and M. Kurulay, "An efficient indexing for Internet of Things massive data based on cloud-fog computing" , Transactions on emerging telecommunications technologies, vol. 31, no. 3, 2020. DOI: 10.1002/ett.3868.
[4] A. Dwivedi, G. Srivastava, S. Dhar, and R. Singh, "A decentralized privacy- preserving healthcare blockchain for IoT" , Sensors, vol. 19, no. 2, p. 326, 2019. DOI: 10.3390/s19020326.
[5] C. Dwork and M. Naor, "Pricing via processing or combatting junk mail" , In annual international cryptology conference. Springer, 1992, pp. 139-147.
[6] F.ul Hassan, A. Ali, S. L., J. Qadir, S. S. Kanhere, J. Singh, J. Crowcroft, "Blockchain and the future of the internet: a comprehensive review" , 2019, ArXiv, abs/1904.00733.
[7] O. Novo, "Blockchain meets IoT: an architecture for scalable access management in IoT" , IEEE internet of things journal, vol. 5, no. 2, pp. 1184-1195, 2018, DOI: 10.1109/jiot.2018.2812239.
[8] H. N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for internet of things: a Survey" , IEEE internet of things journal, vol. 6, no. 5, pp. 8076- 8094, 2019, DOI: 10.1109/JIOT.2019.2920987.
[9] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: architecture, Consensus, and Future Trends" , 2017 IEEE International Congress on Big Data (BigData Congress), 2017, DOI: 10.1109/bigdatacongress.2017.85.
[10] A. K. Yadav and K. Singh, "Comparative analysis of consensus algorithms of blockchain technology" , Advances in intelligent systems and computing ambient communications and computer systems, pp. 205- 218, 2020.
[11] H. D. Zubaydi, Y. W. Chong, K. Ko, S. M. Hanshi, and S. Karuppayah, "A Review on the role of blockchain technology in the healthcare domain" , Electronics, vol. 8, no. 6, pp. 679, 2019, DOI: 10.3390/electronics8060679.
[12] C. Natoli, J. Yu, V. Gramoli, and P. Esteves- Verissimo, "Deconstructing blockchains: a comprehensive survey on consensus, membership and structure" , arXiv:1908.08316, 2019.
[13] S. Zhang and J. H. Lee, "Analysis of the main consensus protocols of blockchain" , ICT express, 2019. DOI: 10.1016/j.icte.2019.08.001.
[14] P. Cui, U. Guin, A. Skjellum, and D. Umphress, "Blockchain in IoT: current trends, challenges, and future roadmap" , Journal of Hardware and Systems Security, vol. 3, no. 4, pp. 338- 364, Apr. 2019. DOI: 10.1007/s41635-019-00079-5.
[15] Natoli, C, Yu, J, Gramoli, V. and Esteves- Verissimo, P, "Deconstructing blockchains: a comprehensive survey on consensus, membership and structure, 2019, arXiv preprint arXiv:1908.08316.
[16] S. Wan, M. Li, G. Liu, and C. Wang, "Recent advances in consensus protocols for blockchain: a survey" , Wireless networks, 2019, DOI: 10.1007/s11276-019-02195-0.
[17] T. A. Syed, A. Alzahrani, S. Jan, M. S. Siddiqui, A. Nadeem, and T. Alghamdi, "A comparative analysis of blockchain architecture and its applications: problems and recommendations" , IEEE access, vol. 7, pp. 176838- 176869, 2019, DOI: 10.1109/access.2019.2957660.
[18] N. Witthayawiroj and P. Nilaphruek,"The development of smart home system for controlling and monitoring energy consumption using WebSocket protocol" , IOP conference series: materials science and engineering, vol. 185, p. 012019, 2017, DOI: 10.1088/1757-899x/185/1/012019.
[19] D. Skvorc, M. Horvat, and S. Srbljic, "Performance evaluation of Websocket protocol for implementation of full- duplex web streams" , 2014 37th international convention on information and communication technology, Electronics and microelectronics (MIPRO), 2014, DOI: 10.1109/mipro.2014.6859715.
[20] Nyamtiga, Sicato, Rathore, Sung, and Park, "Blockchain- based secure storage management with edge computing for IoT" , Electronics, vol. 8, no. 8, p. 828, 2019, DOI: 10.3390/electronics8080828.
[21] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: using blockchain for medical data access and permission management" , 2016 2nd international conference on Open and Big Data (OBD), 2016, DOI:10.1109/obd.2016.1.