IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 26, 2022, accepted March 19, 2022, date of publication March 25, 2022, date of current version April 5, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3162214

Optimized User-Friendly Transaction Time Management in the Blockchain Distributed Energy Market

MARTIN ONYEKA OKOYE AND HAK-MAN KIM, (Senior Member, IEEE)
Department of Electrical Engineering, Incheon National University, Incheon 406-772, South Korea

Corresponding author: Hak-Man Kim (hmkim@inu.ac.kr)

This work was supported by the Energy Efficiency and Resources of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) Grant through the Korea Government Ministry of Knowledge Economy under Grant 20192010106750.

ABSTRACT The blockchain nexus with energy transactions in the distributed energy-trading arena successfully achieved a decentralized transaction and increased security. With the elimination of a third-party middle man, an electronic app is introduced to achieve decentralization and aid transactive communications. Observing the internet-of-things (IoT) intense protocols in the transactive communications amongst blockchain participants, however, leads to transaction time delay with associated uncertainty. This paper integrates the practical byzantine fault tolerance (pBFT) algorithm with the private Hyperledger Sawtooth blockchain network (P) to achieve a P-pBFT algorithm. The P-pBFT achieved a combined two-step transaction latency optimization (minimization) amongst the blockchain participants in the distributed energy generation (DEG) ecosystem. Through their combined feature extraction, further minimization is achieved by simulating the resulting transaction model derived from the integration. Subsequently, an optimization method is proposed to achieve the shortest transaction time given transaction constraints based on participants' comfort. Thus, the ratio of node population to the transaction size and the choice of constraints can be regulated at the participants' convenience to achieve minimum transaction time. Hence, the benefit of deciding the transaction time is achieved thereby eliminating the undesired characteristic uncertainty.

INDEX TERMS Blockchain transaction, blockchain transaction time optimization, blockchain transaction time simulation, distributed energy system, practical byzantine fault tolerance.

I. INTRODUCTION

The global consensus to deviate from the conventional energy generating standard to the clean energy paradigm fostered the increased exploration of alternative energy sources [1], [2]. Also, there is a rising electrical energy demand amongst the populace with increasing electrical activities. Meanwhile, the supply from the conventional utility is sometimes threatened by various issues leading to sudden outages during which alternative source is inevitable [3]. Such issues include natural disasters, faults, overloads, etc. These threaten the power utility's resiliency and reliability leading to a necessity for an alternative supply backup approach. Furthermore, there are yearnings for alternative energy sources that would be more friendly and accommodating to the consumers' peak loads. Consumers wish to shift their peak load usages to a cheaper supply to avert the high cost associated with the demand charges. These demand charges are billed by the utility companies as the extra cost of running the peaking units to satisfy consumers with extraordinary peak loads. Also, serving the local loads with local energy sources helps in reducing the energy losses in long-distance transmission and distribution.

Solar, wind, and hydro became some of the frontline focus owing to the cleanness and renewability of their sources. Following the typical portability of the generating plant (mainly the solar PV) and the enterprising advantage, private individuals, who are formerly sole consumers, now engage in modular generations of electricity. Those with surplus generations consequently engage in energy trade with the sole consumers and those with insufficient consumptions at a price lower than that of the utility. The larger-scale companies likewise engage

The associate editor coordinating the review of this manuscript and approving it for publication was Yassine Maleh.

in a similar profit-making venture with the intent to generate quantities capable of powering an entire establishment such as an institution, a military base, small settlements, etc. Also, profit-driven middlemen participate in buying and storing energy for sale during peak periods. The overall activities led to distributed energy generations amongst the individuals and energy companies. The entire process results in the desire for a well-structured and reliable transaction strategy amongst the participants for easy and seamless energy trading and distribution.

Blockchain technology is a fairly new trading paradigm in the distributed energy generation (DEG) arena borrowed from the financial sector, the cryptocurrency [4]. It replaces the human intervention interface with a decentralized topology in distributed systems [5]. In summary, its promising benefits include but are not limited to the following: reduced transaction cost, increased security, satisfactory transparency, reduced risk, minimized error possibility, easy reconciliation of conflicting financial details, instant financial settlements, etc [6], [7]. These led to its acceptability. Its solution however largely depends on the internet of things (IoT) technology protocols, such as WiFi, ZigBee, Bluetooth, etc, for transaction completeness [8]. This implies that for every transaction to be successful, every participant requires an internet-active device and a strong connection on the move. The internet protocol on the other hand may be unable to offer adequate support at the moment, for example, in times of extremely high-traffic transactions and node population. This and other reasons could lead to unforeseen transaction latency amongst the members of the blockchain consortium. It thus becomes a concern as this raises anxiety amongst participating nodes due to the completion time uncertainty of the initiated transactions.

A. THE BLOCKCHAIN TRANSACTION LATENCY AND UNCERTAINTY

The blockchain transaction time delay is contributed by the activities of the participating members as well as the inherent factors in the blockchain design. The increasing number of nodes and transactions in the blockchain network grossly amounts to increased transaction traffic leading to an unforeseen delay [9]. During the peak traffic period in the blockchain transaction, several undesired bottlenecks are experienced. The transaction demands are queued up amounting to a huge backlog of transaction traffic. Consequently, transaction costs accelerate leading to an overall increased system cost. In addition, since the supplies are outweighed by the demand queues, miners would become selective in the transactions' validation process. This prompts the miners' greater preference to select transactions with higher attached incentives and ignore those with the least incentives. The individual participating nodes would, as a result, slide into competition in the quest to ensure a faster validation of their initiated transactions. This is tactically achieved by increasing the mining token attached to each transaction to attract miners' immediate validation attention. Basically, in a pool of transaction queues, the higher a transaction fee limit is set, the faster the transaction validation would be. The undesired resultant effect is that some transactions are stuck in the Mempool (a queue in the network where all transactions are kept until validation is confirmed) due to a lower transaction fee set [10]. Sometimes, they are rejected by the system after a long stay in the memory pool, and the funds are refunded to the owner, yet the initiated transactions are not invalid [11].

Furthermore, in blockchains, for all transaction blocks, there is a fixed minimum time required before each block is created and appended to the blockchain [12]. For example, the bitcoin and Ethereum blockchains append their new blocks about every 10 minutes and 15 seconds, respectively [13], [14]. Thus, the minimum transaction time required for each transaction in the block is the number of transactions in the block divided by the minimum time required before each block is created. It is logical to focus on how to increase the size of a block in order to increase the number of contained transactions thereby reducing the transaction time. However, each block in the blockchain also has a maximum size limit that cannot be exceeded. For instance, bitcoin has a fixed block size of 1MB [7]. Some blockchains however offer the members the opportunity to alter these factors in consensus, however, it does not solve the transaction time reduction intent. If the block size and the block transaction time are successfully increased and reduced, respectively, there would be a consequent requirement for a corresponding bandwidth increment [15], [16]. This is to aid faster transaction downloads resulting from the increased throughput. Unfortunately, increasing the internet transaction bandwidth is a factor of the individual nodes' connecting devices and internet facilities. These can hardly be achieved in totality.

Other reasons for the transaction delay in the blockchain could include lossy block, insufficient miners, malicious attacks, slow internet, etc [17]. Lossy blocks emanate from the loss of one of the two or more blocks that were validated concurrently [18]. Such blocks cannot be added to the chain at the same time, thus, one is validly added while others, known as orphanage blocks, result in the creation of a temporary fork that is lost in the future [19]. Spam attacks come in form of some aggrieved nodes or adversaries sending loads of minute transaction sizes with tiny fees from one address to another with the aim to slow down the network transaction speed. Additional time delays exist as a result of other unforeseen transaction delay circumstances within the blockchain network. The transaction delay uncertainty would affect consumers' energy purchases in times of urgent demands and emergencies. It is therefore critical to embrace methods that would minimize latency and achieve faster transaction throughput within the distributed energy-trading ecosystem. These formed the basis of the motivations for our paper.

B. RESEARCH CONTRIBUTIONS

Researchers had focused on means to improve energy trading from the distributed generations using blockchain technology. Blockchain itself gained attention in the recent past, in 2009, but in the financial sector initially. Its breakthrough in cryptocurrency transactions spearheaded its acceptance in electrical engineering and other sectors of the economy, hence, the adoption and integration in the energy-trading domain. Several researchers had subsequently made various breakthrough findings following the blockchain-driven enhancement in the energy-trading sector to maneuver its accompanied flaws. The transaction latency was optimized in [20] using a replicated and fault tolerance (RAFT)-based private blockchain but by intelligently migrating transactions from the region of high concentration to the region of low concentration. Several factors, including the choice of constraints by the participants, were not considered in the optimization technique utilized. The transaction latency in the private (permissioned) blockchain is calculated in [21] but was focused on various network configurations. Optimization analysis of the latency time was not considered. Literature [9] performed an optimization in the blockchain model but centered on minimizing the malicious attacks, reduction of communication loss, and the reduction of network loads resulting from the high data storage requirement. Literature [22] proposed an optimized method of verifying blockchain transactions amongst trustworthy members. Using an optimized Merkle tree, a more efficient transaction verification method was achieved. Its transaction time was, however, not considered. In the literature [4], the blockchain transaction time is simulated and the transaction time reduction was considered. A method of reducing the transaction time was consequently proposed. It however did not consider accommodating its optimization (minimization) possibility and participants' transaction constraints accommodation.

Our research contributions are as follows:

i Because the greater portion of the transaction latency is contributed by the transaction validation delay, we take combined advantages of the validation time minimization approach in the network packet transmission proposed by the Practical byzantine fault tolerance (pBFT) model and the transaction time minimization offered by the private (P) hyperledger sawtooth blockchain. These are integrated to achieve the P-pBFT model. Thus, the pBFT algorithm is fused into the private hyperledger sawtooth blockchain network to achieve a combined feature extraction for further transaction latency minimization and prediction. This led to a minimum possible transaction time achievement.

ii We also take advantage of the transaction time prediction simulation result obtained in 'i' to propose an optimized transaction time controllability model supported by the prediction algorithm. Here, the blockchain participants are availed the opportunity of defining their transaction constraints according to their comfort demand. Through optimization, the constraints are accommodated into the prediction model thus achieving more flexible and user-friendly controlled transactions amongst the participants.

C. PAPER ORGANIZATION

The private blockchain model and the pBFT algorithm are briefly introduced in section II. Also, the proposed algorithm derived from their integration, the P-pBFT, is presented and demonstrated in the same section. In section III, the simulation model of the P-pBFT is performed in two categories. Firstly, a P-pBFT-driven simulation is performed by fitting the transacting nodes and transaction blocks to obtain their equivalent transaction time. Secondly, constraints-accommodated optimization of the derived transaction time is performed. The results in the two categories are presented and analyzed in Section IV. Transaction time prediction and optimization approaches are analyzed. Section V concludes the research findings.

II. OPTIMIZED BLOCKCHAIN TRANSACTION FEATURES

A. PRIVATE HYPERLEDGER SAWTOOTH BLOCKCHAIN MODEL

The hyperledger is a blockchain framework with an interest in minimum resource consumption. This makes the choice a good option for cost minimization [23]. It is an open-source system hosted by the Linux foundation. Its flexibility is an interesting and attractive feature. For example, it can host both private and public consortium blockchain. It thus possesses several platforms selectable based on the requirement of the members. These platforms include Burrow, Fabric, Composer, and others [23]. It also offers diverse consensus algorithms to be selected based on suitability to the users' needs. Its consensus algorithms include replicated and fault tolerance (RAFT), practical Byzantine fault tolerance (pBFT), proof of elapsed time (PoET) with several variants, etc.

The Hyperledger Sawtooth is a private blockchain platform that offers the users flexibility of customizing the features according to their preferences. It separates the core domain of the system from the distributed domain thereby allowing users access to a more flexible modular framework [24]. One of the promising features of the hyperledger sawtooth is that it offers a dynamic consensus algorithm. This gives the users the capability of switching from one consensus algorithm to another while remaining on the same blockchain platform. This grants the users more scalability power [25]. Because our paper focused on transaction latency minimization, the consensus algorithm adopted is the pBFT. This also offers other advantages such as the elimination of block-creation delay constraints found in other consensus algorithms such as proof-of-work. It, thus, also offers cost-effective validation. For instance, the expensive computations performed and the high-power consumption, such as that found in the proof-of-work algorithm, are bypassed.

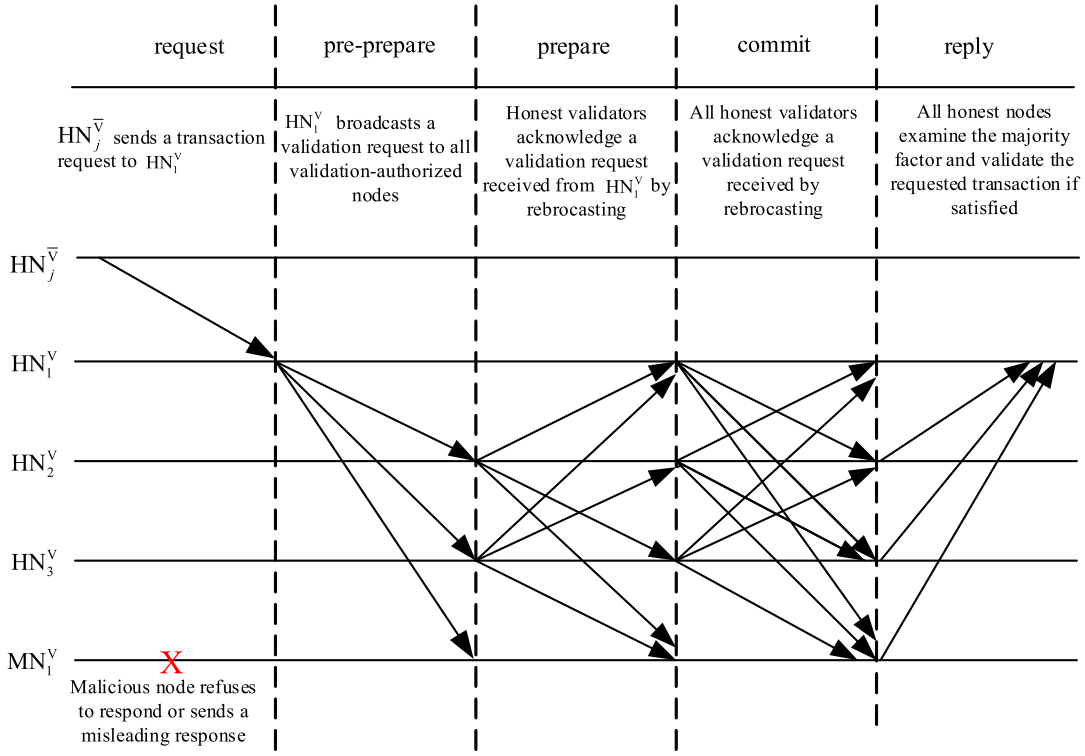

FIGURE 1. The pBFT algorithm.

B. THE PRACTICAL BYZANTINE FAULT TOLERANCE (pBFT)

The BFT is derived from the Byzantine General's problem, a term in computer science in which involved parties embrace a strategy to avert the possibility of total failure [26]. The assumption is that some of the parties are either unreliable or corrupt. The pBFT typically uses this technique to reach a consensus in a distributed network where some of the nodes are malicious and fail to send out information or send incorrect information [27]. All nodes in the network communicate with one another with the aim that all honest nodes would come to an agreement using the majority rule. Thus, possible catastrophic impacts from the influences of malicious nodes are checkmated. In the blockchain transaction, the pBFT is introduced in the consensus algorithm to maneuver possible attacks from the malicious node(s) that could lead to transaction delays and failures [28]. It is integrated with the private blockchain consensus algorithm in this paper to achieve increased transaction speed. For the pBFT system to function, the number of malicious nodes must be less than one-third of the total node population in the network [29]. In comparison with a similar type of algorithm, the RAFT is also a fault tolerance consensus algorithm and they both have other similar features, such as leader-based, non-forking, quorum-based consensus, etc. However, as at the time of this writing, while pBFT is byzantine fault-tolerant (BFT), RAFT is crash fault-tolerant (CFT) [30], [31]. Generally speaking, while BFT supports faults that are less than one-third ($<(1/3)$) of the total blockchain nodes, CFT supports faults that are less than half ($<(1/2)$) of the total nodes [31], [32]. Also, while the BFT focuses on handling the malicious faults, the CFT focuses on node losses resulting from crashes (such as that caused by a partial network failure). RAFT was designed not to consider malicious nodes, i.e., it does not guarantee consensus where malicious nodes are present. Furthermore, the efficiency of RAFT increases with increasing network size and is hence less suitable for smaller networks whilst PBFT supports extensive network sizes that are equal to or greater than 4 nodes. Since our paper proposed transaction time minimization through an integrated algorithm for handling malicious nodes via the Byzantine fault tolerance mechanism, the pBFT is selected for suitability.

To describe the pBFT algorithm, we consider the simplest pBFT node set-up for simplicity and clarity of purpose. It consists of 4 nodes, namely 3 honest nodes and 1 malicious node as shown in Fig. 1. This satisfies the node aggregate condition for the application of pBFT. It states that the total number of malicious nodes must be less than one-third of the total node population saddled with a validation task [33]. The possible minimum (simplest) value of the total node population, whose less-than-one-third value (number of the malicious node) is an integer, is 4. Hence, the chosen value.

LEGEND

MN = Malicious node

HN = Honest node

The consensus steps are divided into four phases as follows [28]:

i *Request*: In the *request* mode, transaction requests, such as energy-purchase requests, are sent from the transaction-requesting nodes to the energy-selling nodes. Requesters may either be a validator (V) or a non-validator ($\overline{\text{V}}$). It is commonly regarded as the leader node.

ii *Pre-prepare*: In this phase, the receiving seller nodes confirm their satisfaction with the requested transaction and consequently proceed to broadcast validation requests to validators in acknowledgment They thereafter wait for the validation request acknowledgment.

iii In the *prepare* phase, the contacted validators, in the *pre-prepare* phase, send a validation request acknowledgment receipt by rebroadcasting the received validation request to all validators. Here, the malicious nodes resume their malicious activities. They either refuse to respond to the validation request or deliberately send an incorrect response, such as a validation-rejection packet.

iv *Commit*: In this phase, because the seller node in the *pre-prepare* phase did not take part in the validation request broadcasts sent in the *prepare* phase, a second phase validation request broadcasts are sent, now by all validators. This is to achieve the number of responses that would suffice to reach the consensus quorum in the replies, thus, satisfying the pBFT condition.

C. THE P-pBFT ALGORITHM

The P-pBFT algorithm extracts the individual promising features of the private hyperledger sawtooth blockchain and the pBFT algorithm to achieve minimal transaction latency. This paper thus implements this nexus to achieve its latency minimization purpose. Generally, for the public blockchain (entire members having validation qualification), the minimum number of validation-authorized nodes required must be greater than half of the total node population. This is as shown in (1).

$$\sum_{i=1}^{n} V_i > \frac{1}{2}\left(\sum_{i=1}^{n} V_i + \sum_{j=1}^{m} \overline{V}_j\right) \tag{1}$$

where V_i and $\overline{V}_j$ are ith validation-authorized node and jth nonvalidator node, respectively. Considering transaction time minimization, the least possible number of validator nodes required (quorum) must satisfy the conditions in (2) and (3). That is, given the total node population, $m+n$, (2) gives the least possible number of validator nodes. This is when the value of $m+n$ is an even (even number) positive integer. For instance, when $m+n=8$, $(8+1)/2=4.5$. Ceiling 4.5 gives 5. Therefore, the number of validator nodes must not be less than 5. Likewise, (3) gives the least possible number of validator nodes when the value of $m+n$ is an odd (odd number) positive integer. For instance, when $m+n=7$, $7/2=3.5$. Ceiling 3.5 gives 4. Hence, the number of validator nodes must not be less than 4.

$$\downarrow \sum_{i=1}^{n} V_i = \left\lceil \frac{\left(\sum_{i=1}^{n} V_i + \sum_{j=1}^{m} \overline{V}_j\right)+1}{2} \right\rceil \Bigg|_{m+n=\text{Even positive integer}} \tag{2}$$

$$\downarrow \sum_{i=1}^{n} V_i = \left\lceil \frac{\left(\sum_{i=1}^{n} V_i + \sum_{j=1}^{m} \overline{V}_j\right)}{2} \right\rceil \Bigg|_{m+n=\text{Odd positive integer}} \tag{3}$$

where $\left(\sum_{i=1}^{n} V_i + \sum_{j=1}^{m} \overline{V}_j\right)$ is the total number of nodes in the public blockchain network.

For private blockchains, the number of validation-qualified nodes required is primarily based on the transaction and role requirements within the members [34]. This is determined and selected in consensus among the entire members. The number of validation-authorized nodes must satisfy the condition in (4). Hence, for validation-qualified nodes, the least possible number of validation-authorized nodes that would achieve the shortest transaction time is given by (5) when n is an even positive integer, and (6) when n is an odd positive integer.

$$\sum_{i=1}^{n'} V_i > \frac{1}{2}\left(\sum_{k=1}^{n} V'_k\right) \tag{4}$$

$$\downarrow \sum_{i=1}^{n'} V_i = \left\lceil \frac{\left(\sum_{k=1}^{n} V'_k\right)+1}{2} \right\rceil \Bigg|_{n=\text{Even positive integer}} \tag{5}$$

$$\downarrow \sum_{i=1}^{n'} V_i = \left\lceil \frac{\left(\sum_{k=1}^{n} V'_k\right)}{2} \right\rceil \Bigg|_{n=\text{Odd positive integer}} \tag{6}$$

where V_i and V'_k are the ith validation-authorized node and kth validation-qualified node, respectively.

For the pBFT consensus algorithm, the validator nodes could comprise malicious nodes and honest nodes as shown in (7).

$$\sum_{k=1}^{n'} VN_k = \sum_{i=1}^{n} HN_i + \sum_{j=1}^{m} MN_j \tag{7}$$

where VN_k, HN_i, and MN_j are the kth validator node, ith honest node, and jth malicious node, respectively.

For the pBFT to be implemented, the condition in (8) must be satisfied. This represents the general formula of the pBFT algorithm. It states that, given the coexistence of malicious nodes and honest nodes in the blockchain transaction platform, the malicious activities of the malicious nodes can be

tolerated and neglected thereby bypassing their anticipated delay to achieve a faster transaction. The number of such malicious nodes however must be less than one-third of the total number of validation-saddled nodes [33]. Therefore, such a total number must be equal to or greater than 4 to achieve a less-than-one-third integer value. This is as given in (8). Thus, the maximum possible number of such tolerable malicious nodes can be determined in every instance of the node population. Given the total node population, $m+n$, (9) gives the maximum possible number of malicious nodes. This is when the value of $m+n$ is a positive non-multiple of 3 where $m+n \geq 4$. For example, when $m+n = 5, 5/3 = 1.67$. Flooring 1.67 gives 1. Therefore, the tolerable number of malicious nodes must not exceed 1. Similarly, (10) gives the maximum possible number of malicious nodes when the value of $m+n$ is otherwise a positive multiple of 3 where $m+n \geq 6$. For example, when $m+n = 9$, (9-1)/3 = 2.67. Flooring 2.67 gives 2. Therefore, the tolerable number of malicious nodes must not exceed 2. Equations (9) and (10) are variants of (8).

$$\sum_{i=1}^{m} MN_i < \frac{1}{3}\left(\sum_{i=1}^{n} HN_i + \sum_{j=1}^{m} MN_j\right) \tag{8}$$

$$\uparrow \sum_{i=1}^{m} MN_i = \left\lfloor \frac{\left(\sum_{i=1}^{n} HN_i + \sum_{j=1}^{m} MN_j\right)}{3} \right\rfloor \tag{9}$$

$\forall(m+n) =$ positive non-multiple of 3, where $(m+n) \geq 4$.

$$\uparrow \sum_{i=1}^{m} MN_i = \left\lfloor \frac{\left(\sum_{i=1}^{n} HN_i + \sum_{j=1}^{m} MN_j\right) - 1}{3} \right\rfloor \tag{10}$$

$\forall(m+n) =$ positive multiple of 3, where $(m+n) \geq 6$.

$\left(\sum_{i=1}^{n} HN_i + \sum_{j=1}^{m} MN_j\right)$ is the total number of validation-authorized nodes in the network. To achieve an integer value of the malicious nodes as a less-than-one-third of the total number of validation-authorized nodes as stated in (8), this total number must be greater than or equal to 4 as shown in (11). That is, it must not be less than 4.

$$\left(\sum_{i=1}^{n} HN_i + \sum_{j=1}^{m} MN_j\right) \geq 4 \tag{11}$$

The total number of validators is given by (12). Thus, the minimum number of validators to achieve the shortest transaction time is given by (13).

$$\sum_{k=1}^{n'} VN_k = 3\sum_{j=1}^{m} MN_j + \Gamma \tag{12}$$

where $1 \leq \Gamma \leq 3$

$$\downarrow \sum_{k=1}^{n'} VN_k = 3\sum_{j=1}^{m} MN_j + 1 \tag{13}$$

We consider a situation where the total node population suddenly goes less than or equal to 3 times the number of malicious validator nodes as shown in (14). This happens when more nodes become corrupt and thus adversaries in the network. The effect is that validations would be halted leading to a pile of initiated transactions that have not been validated. This situation remains a deadlock as long as the validators quorum is not reached. This situation is, however, almost never realistic as the high-security measure in the hyperledger blockchain sawtooth does not permit members to join the blockchain at will but must be added by the administrator. However, at this juncture, taking advantage of the dynamic consensus algorithm spectacularly offered by the hyperledger sawtooth blockchain, we propose the pBFT replacement with the PoET consensus algorithm as shown in (15). With PoET, validation access is granted individually and randomly amongst the nodes [35]. The network assigns each node a random waiting period within which the node with the shortest waiting period wins the validation access of each block. In this manner, malicious nodes never confuse validation protocol. Also, malicious node(s) are easily identified and fished out through their malicious activities when the next PoET validation access accidentally falls on them. The system reverts to the pBFT when normalcy is maintained, i.e when the condition in (8) is reinstated. A similar approach is also taken when the condition in (11) is not satisfied until the otherwise.

$$\left(\sum_{i=1}^{n} HN_i + \sum_{j=1}^{m} MN_j\right) \leq 3\left(\sum_{i=1}^{m} MN_i\right) \tag{14}$$

$$\left(\sum_{i=1}^{m} MN_i < \frac{1}{3}\left(\sum_{i=1}^{n} HN_i + \sum_{j=1}^{m} MN_j\right)\right) \rightarrow \begin{cases} 1, & \text{pBFT} \\ 0, & \text{otherwise, PoET} \end{cases} \tag{15}$$

For the P-pBFT, we fuse (13) into (5) and (6) while maintaining the conditions in (4) and (8).

III. THE TRANSACTION TIME SIMULATION

Similar transaction time optimization was previously achieved by the private blockchain simulation and analysis performed in our previous paper in [4]. This paper further optimizes the transaction time by the P-pBFT method. The simulation results are compared considering similar transaction topology. We consider a distributed energy market transaction comprising of 12 participants as shown in Fig. 2. Maintaining similar notations as were in [4], 7 of the members are the consumers. These are designated by the circles in

green. They are saddled with the validation function which is tied to their condition for subsequent energy purchase ability. Hence, they would only be able to purchase their future energy needs only if they participated in the processing of previous validation requests received. The smaller-size circles, in yellow, are consumption-sufficient energy producers (prosumers) that sell energy to the consumers and other middle agents. They use previous consumption histories of the sole consumers to adapt to their generation requirements, thus, meeting energy consumption demand. The larger circles in yellow are the middlemen who participate in the blockchain transaction solely for the money-making intent. They monitor price changes and buy from the producers at lower prices and sell to the consumers for profit margin.

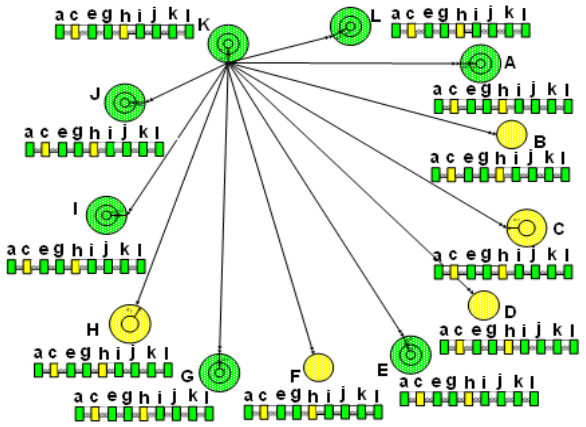

FIGURE 2. Blockchain transaction members.

The features of each member are given in Table 1. For example, *A* has the ability to receive transaction notifications, create transaction blocks, and validate created blocks. Similarly, *B* only has the ability to receive transaction notifications and lacks the ability to create and validate transaction blocks. Let x and y be block-creation time and validation time, respectively. For the demonstration purpose, we respectively assign 0.1 and 0.5 to x and y. Each created block is sent to every validation-authorized node for the transaction validation. Assuming equal network transaction condition and nonconcurrent block creation and validation time, the minimum transaction time (T) required before each created block is added to the chain is given by (16).

$$T = 0.5n + 0.1 \tag{16}$$

where n = number of validation-authorized nodes. With respect to energy transactions and block formation, this is, hence, investigated among the public, private, and P-pBFT blockchain models. Assuming public blockchain, given 12 validation-qualified nodes in Fig. 2, the least number of validation-authorized nodes to achieve minimum transaction time as given by (2) is 7. For private blockchain, given 7 validation-qualified nodes, the least number of validation-authorized nodes given by (6) is 4. Likewise, considering P-pBFT, given 4 validation-qualified nodes, validation time

TABLE 1. Members' transaction features.

Blockchain members (Nodes)	Receive transacion notification	Energy purchase/ block creation	Validation-authorized
A	✓	✓	✓
B	✓	x	x
C	✓	✓	x
D	✓	x	x
E	✓	✓	✓
F	✓	x	x
G	✓	✓	✓
H	✓	✓	x
I	✓	✓	✓
J	✓	✓	✓
K	✓	✓	✓
L	✓	✓	✓

of 3 pBFT-compliant nodes are administered as given by (13) under the condition in (12). This is under the condition of one malicious node. Therefore, for each transacted block, the respective minimum transaction time considering public, private, and P-pBFT are 3.6s (i.e, (0.5*7) + 0.1), 2.1s (i.e, (0.5*4) + 0.1), and 1.6s (i.e, (0.5*3) + 0.1), respectively as obtained from (16). Further energy purchases from 50kWh to 1500kWh resulting in the creation of 8 more transaction blocks were considered. The cumulative minimum transaction time for the public, private, and P-pBFT are tabularly and graphically shown in Table 2 and Fig. 3, respectively. For instance, in Table 2, the public blockchain transaction time of 3.6s is cumulated 9 times, corresponding to 9 energy transactions, to yield 32.4s. A similar cumulative effect yielded 18.9s and 14.4s for the private and P-pBFT blockchain, respectively. The transaction time was shortest in the P-pBFT blockchain followed by the private blockchain and then the public. The transaction time in the P-pBFT was reduced by 23.8% and 55.56% than in the private and public blockchain, respectively. This is as obtained from Table 2 from the final cumulative value as follows: ((18.9 - 14.4)/18.9)*100% = 23.8%. This is for the P-pBFT transaction time reduction relative to the private-based blockchain. The P-pBFT transaction time reduction relative to the public blockchain is obtained as follows: ((32.4 - 14.4)/32.4)*100% = 55.56%. This achievement is universal.

It implies that, given any transaction time reached by the public and private blockchain, further transaction time reduction would be achieved with the P-pBFT algorithm to the tune of the aforementioned respective percentages. This is on the condition that the same transaction network topology is maintained. That is, the ratio of the total number of members and the number of validation-saddled members remain the same. However, if this is altered, such as during network growth, transaction time reduction, further than that of the private blockchain, would always be achieved according to the alteration with the P-pBFT algorithm. This holds for all kinds of network topology. For graphical visualization, the individual values of the public blockchain, private blockchain,

TABLE 2. Block transaction time for public, Private, and P-pBFT Blockchain.

Block created/energy purchased (kWh)	Public blockchain (s)	Private-based blockchain (s)	Private-pBFT blockchain (s)
50	3.6	2.1	1.6
200	7.2	4.2	3.2
500	10.8	6.3	4.8
900	14.4	8.4	6.4
1400	18	10.5	8
1400	21.6	12.6	9.6
1450	25.2	14.7	11.2
1500	28.8	16.8	12.8
1500	32.4	18.9	14.4

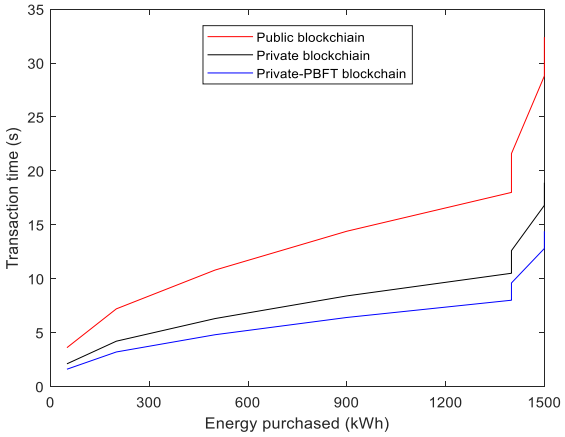

FIGURE 3. Energy transaction time comparison between public, private, and P-pBFT blockchain models.

and P-pBFT blockchain are plotted in comparison against the various energy transactions made. This is as shown in Fig. 3. To compare with our existing paper in [4], the transaction reduction rate achieved in this paper with private blockchain is the same as that achieved in [4]. The difference in the reduction percentages is a result of the difference in their network topologies. For instance, in [4], while the total number of members and the total number of validation-saddled members is 8 and 3, respectively, it is 12 and 7 in this paper. A different topology is however selected in this paper to suitably accommodate the condition for the introduction of the P-pBFT algorithm given in (8). The additional achievement of this paper is the further transaction time reduction to the tune of 23.8% with the integration of the P-pBFT algorithm, hence, one of the novelties. The extra novelty is the further transaction time minimization achieved in section IV subsection B. This makes it more flexible for blockchain users in determining their transaction time ahead of time as well as its adjustment to their convenience.

A. PUBLIC, PRIVATE, & P-pBFT COMPARISONS FOR 1, 2, AND 3 PRACTICAL BYZANTINE FAULTS

To further compare the P-pBFT transaction time minimization achievement with those of the private and public blockchain models, the number of transaction blocks is extended to 50 blocks and assumes equal individual energy purchases. In each case, instead of a single byzantine fault (i.e, malicious node) as performed in the previous simulation, the number of byzantine faults is instead increased from 1 to 3. This is to gain more visualization by comparing the effect of increased malicious nodes on the transaction time. The second intent is to compare this increase among the public, private, and P-pBFT blockchains. To unify the transaction features so as to provide an equal condition amongst the 3 faults, equal quantities of transacted energy were considered. Thus, the individual quantities of transacted energy become negligible. Hence, this is replaced by the number of transactions, ranging from 1 to 50. The emerging graph of the simulation is shown in Fig. 4. The results show that, in each of the three faults, a transaction time ratio, similar to that in the previous simulation, was recorded between the three blockchains. That is, the transaction time reduction amongst the various 3 blockchains maintained the same percentages as was obtained from Table 2 and Fig. 3. The P-pBFT-specific visualization is shown in Fig. 5 by comparing the P-pBFT transaction time in the 1, 2, and 3 byzantine faults. The quantitative difference between the 3 faults is a result of the increased number of nodes required for the increasing byzantine faults. This consequently results in increased transaction time. This is so as to satisfy the condition, the valid maximum number of malicious nodes, given in (8).

IV. TRANSACTION TIME PREDICTION AND MINIMIZATION

A. TRANSACTION TIME REGRESSION AND PREDICTION

To compare the P-pBFT results with those obtained from the private blockchain network in [4], subsequent simulations were performed. The network simulator (NS3) was used to emulate and simulate the transactive property of the blockchain transactive platform. This is to obtain the respective transaction time for each node increment and block size increment. The intent is to extract the underlying pattern in the relationship between the transaction time, block size, and the number of nodes. This focus is to obtain a fitting equation, from the underlying relationship, that would be used as an objective function to perform optimization analysis. NS3 is a network simulator for a series of discrete events using the internet. It provides researchers with features that are utilized to test-rout algorithms and protocols in lieu of the physical hardware [36], [37]. Just like the node-to-node transactive feature existent in the blockchain transactive platform, NS3 offers similar features that achieve point-to-point wireless communication using the internet connection. It allows for the installation of devices, internet stacks, etc. that enable users to animate data transfer between nodes. It emulates the blockchain transactive network and offers the opportunity to simulate the transaction time obtainable from the blockchain transaction blocks and nodes. It possesses several other parameter settings required to achieve the blockchain

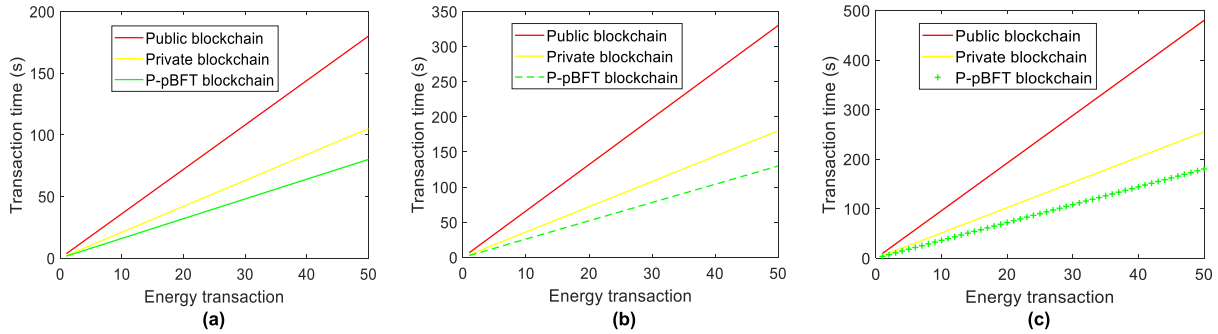

FIGURE 4. Public, Private, and P-pBFT blockchain model comparisons for (a) 1 PB Fault, (b) 2 PB Faults, and (c) 3 PB Faults.

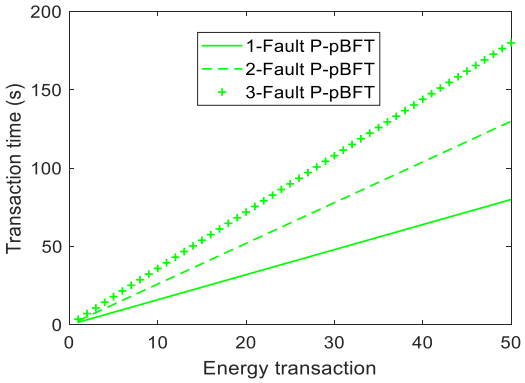

FIGURE 5. P-pBFT comparison for 1, 2, & 3 PB Faults.

network simulation, such as source IP, destination IP, sequence number, etc. Its transaction interval is stochastic and was achieved by the Montecarlo equation [38] given in (17) and (18). Equation (17) generates the transaction time of the longer transaction while the transaction time of the shorter transaction is generated by (18).

$$T_S = (-1/\mu)\ln U_1 \tag{17}$$

$$T_L = (-1/\lambda)\ln U_2 \tag{18}$$

where T_S and T_L are shorter transaction and longer transaction time, respectively; μ and λ are the transaction rates of the shorter transaction time and longer transaction time with values, 0.8 and 0.2, respectively; μ and λ are related as μ = 1- λ; U_1 and U_2 are set of random numbers between 0 to 1. In each of the increments of the block size and number of nodes, the other was kept constant. When the block size was increased from 10KB to 5120KB at the step increments of 10KB, the node was fixed at 250. The simulation result is as shown in Fig 6(a). Similarly, the nodes were increased from 4 to 500 during which the block size was kept constant at 2.5MB as shown in Fig 6(b).

To unify the block size and number of nodes (in Fig. 6(a) and Fig. 6(b), respectively) and obtain a common transaction time (T), first, the DecisionTreeRegressor algorithm was used to fit the block size (Z) and its corresponding transaction time (T_z). The fitting data are obtained from Fig. 6(a). The train_test_split parameters of the simulation include:

TABLE 3. Block transaction time optimization scripts.

```
1  import matplotlib.pyplot as plt
2  from docplex.mp.model import Model
3  milp_model = Model(name = 'MILP')
4  N=milp_model.integer_var(name = 'N' , lb=0)
5  Z=milp_model.continuous_var(name = 'Z' , lb=0)
6  c1=milp_model.add_constraint(6*Z + 2*N >=50,  ctname = 'c1')
7  c2=milp_model.add_constraint(N>=5,  ctname = 'c2')
8  obj_fn = 0.0088*Z + 0.1054*N + 0.9172
9  milp_model.set_objective('min', obj_fn)
10 milp_model.print_information()
11 milp_model.solve()
```

test_size = 0.2, random_state = 90, and a prediction accuracy of 0.99 was obtained. Second, the resulting fitting model was thereafter used to predict the node transaction time (T_N) from Fig 6(b) to obtain the equivalent new block size Z_{NEW}. Thus, the emerging variables include the existing number of nodes (N), T_N, and Z_{NEW}. The Z_{NEW} ranges from 10KB to 3590KB. The visualization of the graph of N, T_N, and Z_{NEW} is shown in Fig. 7. A unified transaction time is common to both the block size and the number of nodes axes, hence, the optimization function equation could be easily obtained subsequently.

To minimize the transaction time in Fig. 7, first, its fitting equation is obtained. Since the relationship in the data is linear, the linear regression model is used to obtain the fitting equation as shown in (19). The simulation was performed in PYTHON using the Jupyter Notebook IDE. The simulation details include test_size = 0.2, random_state = 90, and the prediction accuracy of 0.99. Knowing the number of blockchain participants and estimate of the block size, the equation can conveniently be used to achieve transaction time awareness ahead of time via prediction. It is necessary to note that this is specific to the transaction consortium in question. Equation (19) would vary depending on the transaction pattern and transaction history of the consortium under consideration.

$$T = 0.0088Z + 0.1054N + 0.9172 \tag{19}$$

B. TRANSACTION TIME MINIMIZATION

Hitherto, the anxiety of transaction time uncertainty grossly affects the pace of development and acceptability of

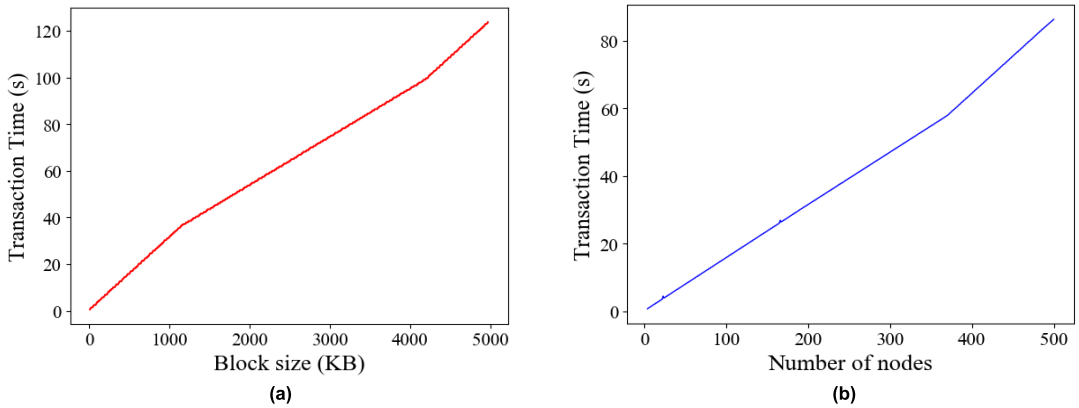

FIGURE 6. Blockchain transaction time for (a) Increasing block sizes, (b) Increasing number of nodes.

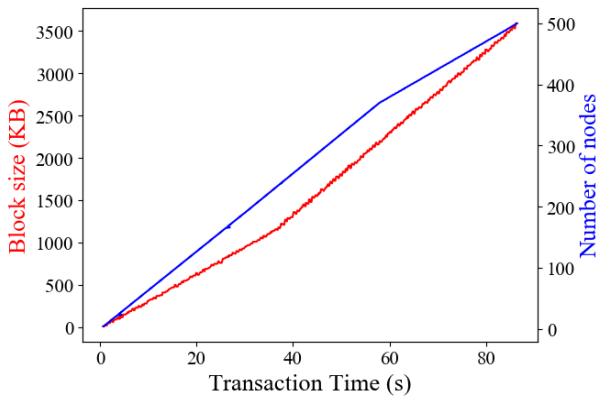

FIGURE 7. The unified transaction time for the increasing block sizes and nodes.

```
Model: MILP
  - number of variables: 2
    - binary=0, integer=1, continuous=1
  - number of constraints: 2
    - linear=2
  - parameters: defaults
  - objective: minimize
  - problem type is: MILP

Out[1]: docplex.mp.solution.SolveSolution(obj=1.50287,values={N:5,Z:6.66667})
```

FIGURE 8. Simulation result of the transaction time optimization scripts in Table 3.

blockchain technology. Hence, its adoption and subsequent integration in various sectors still face admissibility challenges varying from user to user. With the derivation of a model equation from the blockchain transaction history, it becomes feasible to leverage such an equation in determining and controlling the transaction time. In furtherance to the minimization focus, various exploitations of the equation lead to transaction time prediction and subsequent minimization. This is via manipulation of the equation variables. Since the block size can also take float values while the number of nodes can only take integer values, representing the number of humans, the mixed-integer linear programming (MILP) optimization model was used to achieve transaction time minimization. Equation (19) was consequently optimized leading to the transaction time minimization approach as given in the 11 lines of code in Table 3. During this process, the transaction time could be manipulated according to the transaction time collective preference of blockchain members. This is achieved via two approaches. The first is by varying the equation variables according to the members' collective preference reached in consensus. The resulting predictive transaction time is thereafter calculated. For instance, the members may decide on limiting their population size (N) to a certain number and/or controlling the transaction block size (Z) within a certain range. The second is by introducing a preferred minimization constraint and calculating the resulting value of the objective function. Both approaches are tailored towards controlling transaction time minimization according to the blockchain members' unanimous preference.

In Table 3, Line 3 chooses the type of optimization model to be mixed-integer linear programming (MILP). Line 4 restricts the variable type of the *number of node* to an integer variable. Line 5 defines the variable type of the *block size* as a continuous variable. Line 6 contains a test-optimization constraint, $6^*Z + 2^*N \geq 50$, where Z = transaction block size and N = number of nodes. Line 7 contains another test-optimization constraint that requires the minimum number of nodes to be greater than or equal to 5. Line 8 contains the objective function equation, (19), that was optimized. Line 11 solves the optimization model. The optimization model presented a minimum transaction time of 1.5029s at $Z = 6.67$ and $N = 5$ as shown in Fig. 8. This is the output of the optimization performed using (19), transaction time equation, as the objective function. It implies that for a minimum transaction time of ≈1.50s to be achieved, the transaction block size and number of blockchain members must be limited to ≈6.67 and 5, respectively.

In a similar approach, the blockchain consortiums in consensus can alter constraints to suit their collective interest

to determine and achieve the minimum transaction time ahead of the transactions. The simulation was performed in PYTHON under Jupyter Notebook IDE and using the libraries of the CPLEX optimization package. The blockchain participants, thus, possess the ability to determine and embrace the minimum transaction time given preferred constraints.

V. CONCLUSION

In our proposed model, the transaction latency was reduced to the tune of 23.8% by the P-pBFT relative to the private chain model. This is a result of increased energy transactions given a time-limit constraint. Reducing the transaction latency consequently achieves faster transactions, annul or reduces extra transaction fees, and promotes more satisfactory blockchain admissibility. This offers the consortium members increased profit given the same transaction time as in the former. Several other benefits are achieved by the proposed model. With the proposed transaction consensus algorithm, the hyperledger sawtooth, waiting periods, such as that found in the bitcoin and Ethereum blockchains, are eliminated thereby promoting faster transactions. Forking is also absent which upholds consistent transaction protocols. Furthermore, following the selected consensus algorithm, energy consumption-intensive computations, such as that associated with the proof-of-work algorithm, are eliminated. Hence, its associated cost is consequently reduced. In addition, the fewer number of validators required leads to a reduction in the transaction bandwidth requirements. This in return increases the transaction throughput needed to achieve the fastest transaction. Also, the flexibility in regulating the node population and the transaction size ultimately allows the members to own the transaction time gross management and control.

REFERENCES

[1] A. Y. Ali, A. Hussain, J.-W. Baek, and H.-M. Kim, "Optimal operation of networked microgrids for enhancing resilience using mobile electric vehicles," *Energies*, vol. 14, no. 1, p. 142, Dec. 2020.

[2] A. Hussain and H.-M. Kim, "Evaluation of multi-objective optimization techniques for resilience enhancement of electric vehicles," *Electronics*, vol. 10, no. 23, p. 3030, Dec. 2021.

[3] A. Hussain, A. O. Rousis, I. Konstantelos, G. Strbac, J. Jeon, and H.-M. Kim, "Impact of uncertainties on resilient operation of microgrids: A data-driven approach," *IEEE Access*, vol. 7, pp. 14924–14937, 2019.

[4] M. O. Okoye, J. Yang, J. Cui, Z. Lei, J. Yuan, H. Wang, H. Ji, J. Feng, and C. Ezeh, "A blockchain-enhanced transaction model for microgrid energy trading," *IEEE Access*, vol. 8, pp. 143777–143786, 2020.

[5] Y. Long, Y. Chen, W. Ren, H. Dou, and N. N. Xiong, "DePET: A decentralized privacy-preserving energy trading scheme for vehicular energy network via blockchain and K—Anonymity," *IEEE Access*, vol. 8, pp. 192587–192596, 2020.

[6] H. R. Hasan, K. Salah, R. Jayaraman, I. Yaqoob, M. Omar, and S. Ellahham, "COVID-19 contact tracing using blockchain," *IEEE Access*, vol. 9, pp. 62956–62971, 2021.

[7] O. Ali, A. Jaradat, A. Kulakli, and A. Abuhalimeh, "A comparative study: Blockchain technology utilization benefits, challenges and functionalities," *IEEE Access*, vol. 9, pp. 12730–12749, 2021.

[8] A. S. M. S. Hosen, S. Singh, P. K. Sharma, U. Ghosh, J. Wang, I.-H. Ra, and G. H. Cho, "Blockchain-based transaction validation protocol for a secure distributed IoT network," *IEEE Access*, vol. 8, pp. 117266–117277, 2020.

[9] W. Junlu, L. Qiang, and S. Baoyan, "Research on the optimization model of blockchain hierarchical proxy," *IEEE Access*, vol. 9, pp. 144327–144340, 2021.

[10] H. Shi, S. Wang, and Y. Xiao, "Queuing without patience: A novel transaction selection mechanism in blockchain for IoT enhancement," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 7941–7948, Sep. 2020.

[11] L. Zhang, R. Zhou, Q. Liu, J. Xu, and C. Liu, "Transaction confirmation time estimation in the bitcoin blockchain," in *Proc. Int. Conf. Web Inf. Syst. Eng.* Cham, Switzerland: Springer, 2021, pp. 30–45.

[12] L. D. Negka and G. P. Spathoulas, "Blockchain state channels: A state of the art," *IEEE Access*, vol. 9, pp. 160277–160298, 2021.

[13] C. Xu, Y. Qu, T. H. Luan, P. W. Eklund, Y. Xiang, and L. Gao, "A lightweight and attack-proof bidirectional blockchain paradigm for Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4371–4384, Mar. 2022.

[14] M. Kaur, M. Z. Khan, S. Gupta, A. Noorwali, C. Chakraborty, and S. K. Pani, "MBCP: Performance analysis of large scale mainstream blockchain consensus protocols," *IEEE Access*, vol. 9, pp. 80931–80944, 2021.

[15] K. Lei, M. Du, J. Huang, and T. Jin, "Groupchain: Towards a scalable public blockchain in fog computing of IoT services computing," *IEEE Trans. Services Comput.*, vol. 13, no. 2, pp. 252–262, Mar. 2020.

[16] N. Kannengießer, S. Lins, T. Dehling, and A. Sunyaev, "Trade-offs between distributed ledger technology characteristics," *ACM Comput. Surveys*, vol. 53, no. 2, pp. 1–37, Mar. 2021.

[17] X. Sun, F. R. Yu, P. Zhang, Z. Sun, W. Xie, and X. Peng, "A survey on zero-knowledge proof in blockchain," *IEEE Netw.*, vol. 35, no. 4, pp. 198–205, Jul. 2021.

[18] F. Z. da N. Costa and R. J. G. B. de Queiroz, "A blockchain using proof-of-download," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, Rhodes, Greece, Nov. 2020, pp. 170–177.

[19] Y. Liu, Y. Hei, T. Xu, and J. Liu, "An evaluation of uncle block mechanism effect on ethereum selfish and stubborn mining combined with an eclipse attack," *IEEE Access*, vol. 8, pp. 17489–17499, 2020.

[20] L. Hou, X. Xu, K. Zheng, and X. Wang, "An intelligent transaction migration scheme for RAFT-based private blockchain in Internet of Things applications," *IEEE Commun. Lett.*, vol. 25, no. 8, pp. 2753–2757, Aug. 2021.

[21] X. Xu, G. Sun, L. Luo, H. Cao, H. Yu, and A. V. Vasilakos, "Latency performance modeling and analysis for hyperledger fabric blockchain network," *Inf. Process. Manage.*, vol. 58, no. 1, Jan. 2021, Art. no. 102436.

[22] J. Wang, B. Wei, J. Zhang, X. Yu, and P. K. Sharma, "An optimized transaction verification method for trustworthy blockchain-enabled IIoT," *Ad Hoc Netw.*, vol. 119, Aug. 2021, Art. no. 102526.

[23] *Sawtooth PBFT Latest Documentation*. Accessed: Feb. 22, 2021. [Online]. Available: https://sawtooth.hyperledger.org/faq/consensus.html

[24] Z. Leng, Z. Tan, and K. Wang, "Application of hyperledger in the hospital information systems: A survey," *IEEE Access*, vol. 9, pp. 128965–128987, 2021.

[25] W. Li, C. Feng, L. Zhang, H. Xu, B. Cao, and M. A. Imran, "A scalable multi-layer PBFT consensus for blockchain," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 5, pp. 1146–1160, May 2021.

[26] S. Guo, T. Zhang, H. Yu, X. Xie, L. Ma, T. Xiang, and Y. Liu, "Byzantine-resilient decentralized stochastic gradient descent," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Oct. 1, 2021, doi: 10.1109/TCSVT.2021.3116976.

[27] Y. Li, L. Qiao, and Z. Lv, "An optimized byzantine fault tolerance algorithm for consortium blockchain," *Peer-to-Peer Netw. Appl.*, vol. 14, no. 5, pp. 2826–2839, Sep. 2021.

[28] Y. Meshcheryakov, A. Melman, O. Evsutin, V. Morozov, and Y. Koucheryavy, "On performance of PBFT blockchain consensus algorithm for IoT-applications with constrained devices," *IEEE Access*, vol. 9, pp. 80559–80570, 2021.

[29] Y. Wang, S. Cai, C. Lin, Z. Chen, T. Wang, Z. Gao, and C. Zhou, "Study of blockchains's consensus mechanism based on credit," *IEEE Access*, vol. 7, pp. 10224–10231, 2019.

[30] J. Abdella, Z. Tari, A. Anwar, A. Mahmood, and F. Han, "An architecture and performance evaluation of blockchain-based peer-to-peer energy trading," *IEEE Trans. Smart Grid*, vol. 12, no. 4, pp. 3364–3378, Jul. 2021.

[31] A. Barger, Y. Manevich, H. Meir, and Y. Tock, "A byzantine fault-tolerant consensus library for hyperledger fabric," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, Sydney, NSW, Australia, May 2021, pp. 1–9.

[32] S. Zhou and B. Ying, "VG-raft: An improved byzantine fault tolerant algorithm based on raft algorithm," in *Proc. IEEE 21st Int. Conf. Commun. Technol. (ICCT)*, Tianjin, China, Oct. 2021, pp. 882–886.

[33] P. Zhu, J. Hu, Y. Zhang, and X. Li, "A blockchain based solution for medication anti-counterfeiting and traceability," *IEEE Access*, vol. 8, pp. 184256–184272, 2020.
[34] J. W. Kim, "Blockchain technology and its applications: Case studies," *J. Syst. Manage. Sci.*, vol. 10, no. 1, pp. 83–93, 2020.
[35] S. Johar, N. Ahmad, A. Durrani, and G. Ali, "Proof of pseudonym: Blockchain-based privacy preserving protocol for intelligent transport system," *IEEE Access*, vol. 9, pp. 163625–163639, 2021.
[36] A. Aldalbahi, M. Rahaim, A. Khreishah, M. Ayyash, and T. D. C. Little, "Visible light communication module: An open source extension to the NS3 network simulator with real system validation," *IEEE Access*, vol. 5, pp. 22144–22158, 2017.
[37] Z. Liu, C. Guo, and B. Wang, "A physically secure, lightweight three-factor and anonymous user authentication protocol for IoT," *IEEE Access*, vol. 8, pp. 195914–195928, 2020.
[38] M. O. Okoye, J. Yang, and Y. Li, "The nonlinearity property accommodation in the Monte Carlo method of generation system reliability prediction by the neural network model," *Energy Rep.*, vol. 7, pp. 505–510, Apr. 2021.

MARTIN ONYEKA OKOYE received the B.Eng. degree in electrical/electronics and computer engineering from Nnamdi Azikiwe University, Awka, Nigeria, in 2008, the M.Sc. degree in electronic systems design engineering from Universiti Sains Malaysia, Penang, Malaysia, in 2018, and the Ph.D. degree in electrical engineering from the Shenyang University of Technology, Shenyang, China, in 2021. He is currently a Postdoctoral Researcher with the Department of Electrical Engineering, Incheon National University, South Korea. His research interests include generation systems, power system reliability assessment, cyber-physical systems, blockchain technology, and microgrid operations.

HAK-MAN KIM (Senior Member, IEEE) received the first Ph.D. degree in electrical engineering from Sungkyunkwan University, South Korea, in 1998, and the second Ph.D. degree in information sciences from Tohoku University, Japan, in 2011. He worked at the Korea Electrotechnology Research Institute (KERI), South Korea, from October 1996 to February 2008. Currently, he is a Professor with the Department of Electrical Engineering, Incheon National University, South Korea. His research interest includes microgrid operation and control.