Proof of X: Experimental Insights on Blockchain Consensus Algorithms in Energy Markets

*Travis Machacek
The MITRE Corporation
tmachacek@mitre.org

Milan Biswal and Satyajayant Misra
New Mexico State University
{milanb, misra}@cs.nmsu.edu

2021 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT) | 978-1-7281-8897-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISGT49243.2021.9372194

***Abstract*—The current centralized model of the electricity market is not efficient in performing distributed energy transactions required for the transactive smart grid. One of the prominent solutions to this issue is to integrate blockchain technologies, which promise transparent, tamper-proof, and secure transaction systems specifically suitable for the decentralized and distributed energy markets. Blockchain has already been shown to successfully operate in a microgrid peer-to-peer (P2P) energy market. The prime determinant of different blockchain implementations is the consensus algorithm they use to reach consensus on which blocks/transactions to accept as valid in a distributed environment. Although different blockchain implementations have been proposed independently for P2P energy market in the microgrid, quantitative experimental analyses and comparison of the consensus algorithms that the different blockchains may use for energy markets, has not been studied. Identifying the right consensus algorithm to use is essential for scalability and operation of the energy market. To this end, we evaluate three popular consensus algorithms: (i) proof of work (PoW), (ii) proof of authority (PoA), and (iii) Istanbul Byzantine fault tolerance (IBFT), running them on a network of nodes set up using a network of docker nodes to form a microgrid energy market. Using a series of double auctions, we assess each algorithm's viability using different metrics, such as time to reach consensus and scalability. The results indicate that *PoA* is the most efficient and scalable consensus algorithm to hold double auctions in the smart grid. We also identified the minimum hardware specification necessary for devices such as smart meters, which may run these consensus algorithms.**

***Keywords*—Microgrid energy trading, smart grid, blockchain, double-auction.**

I. Introduction

In the past, power grids have generated power in a largely centralized manner. The smart grid initiatives augmented by the prevalence of distributed energy resources (DERs) is gradually changing the nature of the grid towards a more decentralized and distributed paradigm. Many traditional consumers are being transformed to *prosumers*, which means that they not only consume but also produce electricity, bringing new challenges to distributed energy management and transactions [1].

*The author's affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions, or viewpoints expressed by the author.

The blockchain technology with distributed ledgers has created an opportunity for a decentralized energy market design that allows both consumers and prosumers to conduct bilateral and centralized trades to enable the transactive smart grid [2]. In addition, applying blockchain based smart contracts presents an opportunity to increase the speed, scale, and security of transactive energy applications. This can provide a robust path for a decentralized modern smart electric grid and paves the way for integration of energy internet-of-things (E-IoT) devices, such as smart meters and home energy management systems (HEMS).

The Brooklyn Microgrid [3] has already demonstrated that it is possible to design and operate a blockchain-based local energy market. In [4], the authors designed a blockchain mechanism to create a peer-to-peer (P2P) energy trading platform that shows their distributed double auction promotes more energy transfer than a centralized auction. In another work, Kang et al. [5], proposed a energy trading platform for plug-in hybrid electric vehicles (PHEVs) in the smart grid using consortium blockchains. The authors demonstrated that PHEVs can trade energy without reliance on a third party by utilizing their iterative double auction mechanism. They also ran a security analysis and showed that the proposed method could improve transaction security and privacy.

However, these works test their methods with only one consensus mechanism such as: proof of work (PoW), proof of authority (PoA), proof of stake (PoS), Istanbul Byzantine fault tolerance (IBFT), or practical Byzantine fault tolerance (PBFT). A key determinant in blockchain based trading is the time and resources required to reach consensus. This becomes particularly pronounced in the distributed energy market where a sizeable number of prosumers may participate with low-capability devices. Therefore, the consensus algorithms and their scalability plays a pivotal role in the auction design for the energy market in the transactive smart grid. However, a comparative evaluation of the different consensus algorithms, resulting in the identification of which ones are more suitable for peer-to-peer energy markets has not received due attention.

Contributions: Motivated by the above gap, in this paper, we identify pertinent consensus algorithms for energy trading, and evaluate the effectiveness of the more capable consensus algorithms, for usage in the double auction energy markets. From our identification, we chose three popular consensus algorithms: (i) PoW, (ii) PoA, and (iii) IBFT. The major

978-1-7281-8897-3/21/$31.00 ©2021 IEEE

contributions of this paper is: *an experimental evaluation of the quantitative results, to probe the performance of the chosen (popular and pertinent) consensus algorithms in a double auction setting, and illustration of their comparative performance and scalability.*

The paper is organized as follows: Section 2, presents a brief review of the related work. In Section 3, we present the proposed auction model. In Section 4, the experimental setup and the associated results are described. Finally, in Section 5, we summarize the findings in the conclusions.

II. Related Work

The first project to successfully facilitate a P2P energy transaction was the Brooklyn Microgrid (BMG) [3]. It derived seven market components that outline a foundation for constructing an efficient market for energy transactions, and demonstrated that the blockchain implementation can successfully operate and sustain an energy market. The BMG case study involved a three month trial period with P2P energy transactions taking place between two participants with a closed order book double auction in 15 minute time slots. This demonstrated a private blockchain can indeed support a microgrid energy market. While the BMG partially met all of the seven market components, it has certainly sparked an interest into the design of microgrid energy markets with blockchain technology.

In [4], the authors used blockchain to develop a P2P energy trade platform that allows any peer to act as an auctioneer and the blockchan mechanism ensures that a peer behaves lawfully when acting as the auctioneer. This demonstrated that the distributed auction converges quickly, minimizes loss due to transmission, overhead from blockchain is minimal and it can implement trade restrictions imposed by the energy distribution network. The authors used the proof of stake (PoS) consensus protocol to create and maintain the blockchain. A key limitation of their method is its applicability to only single-unit demands.

The authors in [5] proposed a blockchain that has a specified list of authorized nodes that distribute and manage the shared ledger. The authorized nodes in their blockchain are the local aggregators (LAGs). Energy transactions among PHEVs are uploaded to LAGs for auditing and recording transactions to the ledger. The authors also use a new digital cryptocurrency named, *energy coin*, as the digital asset to trade electricity. Just as in Bitcoin, they reward the fastest LAG to find a hash for the new block with a certain amount of energy coin. In addition, the PHEVs that discharge the most electricity are also rewarded with coins as incentive to keep discharging electricity.

In [6], the authors showed that continuous double auction (CDA), can be used in microgrid energy transactions, thus allowing both parties (buyers and sellers) to dynamically adjust quotes. To find the equilibrium price, the authors employed the adaptive aggressive strategy which makes decisions based off of market information and price constraints [7]. However, the paper does not discuss the exact type of blockchain used in their simulation, but frequently mention the use of bitcoin, or Satoshi, for payments of electricity, which indicates that they used PoW for the consensus. In [8], the authors addressed the current problem of single point of failures in iterative double auctions due to their centrality. To solve this problem, the authors devised a trustless and decentralized framework for iterative double auction based on blockchain.

While all of these works have exposed the potential benefits of the overall blockchain technology in the transactive smart grid, they fail to present a comparative evaluation of different alternative blockchain solutions. This limitation is particularly prominent considering the the unique characteristics of scalability and time-bounded nature of the energy market transactions. In this work, we attempt to bridge this gap by comparing the performance of three well-established consensus algorithms.

III. Consensus Algorithms and Smart Contracts

In this section, we discuss our choices of consensus protocols and give a brief overview of smart contracts.

a) *Proof of Work (PoW)*: In *PoW*, to commit a block, miners have to compute a hash that has a strict length requirements. Given that the hash function used is cryptographically secure, computation by bruteforce is the only option, making it computationally intensive. Miners can reach consensus easily by verifying that the proposed hash is below a certain length.

b) *Proof of Authority (PoA)*: In *PoA*, there is a predefined set of authorities have their identity at stake when they propose a block. Instead of bruteforcing a hash, authorities simply propose a block to the network and it is comitted to the chain. There is only one round of communication required for this process.

c) *Istanbul Byzantine Fault Tolerance (IBFT)*: In *IBFT*, similar to *PoA*, there is predefined set of validators that are allowed to commit blocks. *IBFT* has three main phases when a block is proposed by a validator which requires three rounds of communication.

Due to lack of space, we only give brief descriptions of each algorithm. More information can be found about each in [9]–[12].

d) *Smart Contracts:* A smart contract is a computer protocol intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract. Smart contracts help involved parties exchange money, property, shares, or anything of value in a trans-parent, conflict-free way while avoiding the services of a middleman. Contracts are easily customizable and give the developer a wide variety of options for how the contract will be executed. To provide a decentralized setting with transparent rules, we decided to use smart contracts to implement our double auction. One important item of note is the application binary interface (ABI) that each entity must possess in order to interact with the contract. This is made available by compiling the contract where more information can be found at [13].

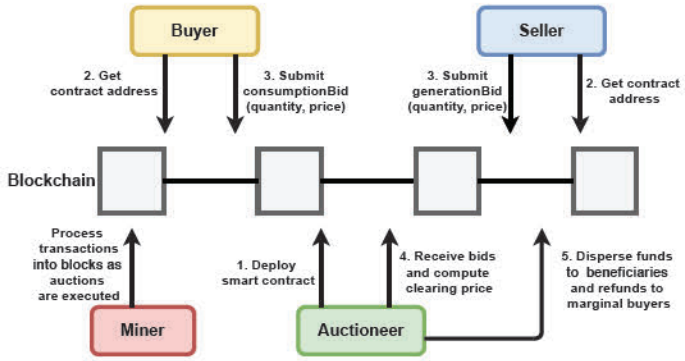


Fig. 1: System Model

IV. Double Auction System

Fig. 1, shows the components of our system model for the double auction experiment. The four entities in the model are the auctioneer, energy buyer, energy seller, and blockchain miner. First, we describe our setup prior to starting the double auction process, then we describe the steps mentioned in Fig. 1. The system is bootstrapped with every node compiling the same smart contract as referred to in Section 3. Next, the buyers and the sellers need to know when an auction has been started. We handle this by using Ethereum's subscribe function. We have the buyers and sellers subscribe to all transactions made by the auctioneer. The two parties receive transaction hashes from the blockchain. With these hashes, buyers/sellers forward them as an input to a transaction receipt function that tells them if that transaction was a contract deployment or not. The receipt always contains a $contract\_address$ field, the field has the value of $null$ if it was not a contract deployment. Otherwise, it will contain the address of the contract, which corresponds to the entity to whom the consumption and generation bids are sent. This will signal that an auction has started and the buyers and sellers will proceed to send their bids.

Starting with Step 1, the auctioneer will deploy a double auction smart contract in the blockchain. In Step 2, the buyers and sellers will fetch the contract address from the blockchain. In Step 3, the buyers and sellers will submit their consumption and generation bids to the contract. In Step 4, the auctioneer will retrieve the bids from the blockchain and compute the clearing price. In Step 5, the auctioneer will disperse the funds to the beneficiaries and will refund any buyers whose price was below the clearing price. In our model, we do not have communication showing that entities have received or delivered any energy. In this paper, we make the assumption that buyers who do not meet the clearing price, receive their desired energy from the traditional grid operator (i.e. utility) and once the sellers have received their funds, they send the requested energy to the buyers. These steps were repeated for each of the 100 auctions we ran for each experiment; more description in Section V.

V. Experiments

In this section, we first present our experimental setup and then present a comparative performance assessment of the consensus algorithms.

A. Experimental Setup

All the experiments were conducted on a machine equipped with a 2.3 GHz CPU, 32 GB of RAM, and an Ubuntu 18.04 LTS operating system on a virtual machine. To emulate different networks, we used docker containers to create an environment for the PoW, PoA, and IBFT blockchains. Each docker container represents a smart meter in the microgrid and has a 1 GB limit of RAM to emulate the smart meter's processing capabilities. Since CPU limiting can only be done by specifying how many cores a container can use, we could not constrain CPU usage on a single machine.

For each of the consensus protocols, we simulated networks that consisted of 10, 20, and 30 nodes. In each network, there are three miners/authorities/validators with respect to the PoW, PoA, and IBFT, respectively. In our experiment, the miners are also located at the same place as the auctioneers (they can be at separate locations), and are responsible for deploying smart contracts, and committing blocks to the chain. The rest of the nodes consist of consumers and producers of energy that place their consumption/generation bids on the smart contract. For each test, we executed 100 double auction smart contracts, and monitored each chain's performance to assess how long it took to commit a block to the chain, and how long each auction took to finalize. For smart contracts, we used the Truffle Suite [13] to compile and deploy our double auction contracts.

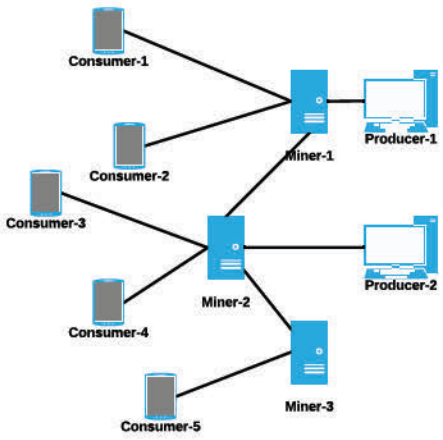
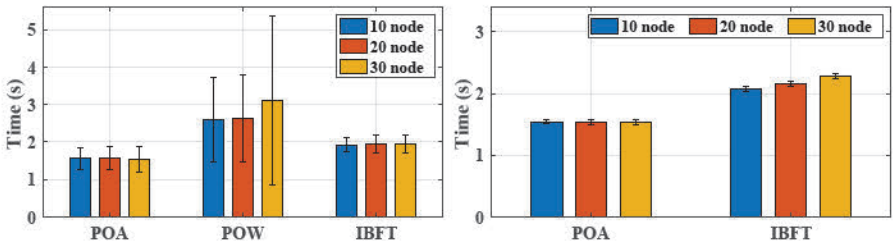


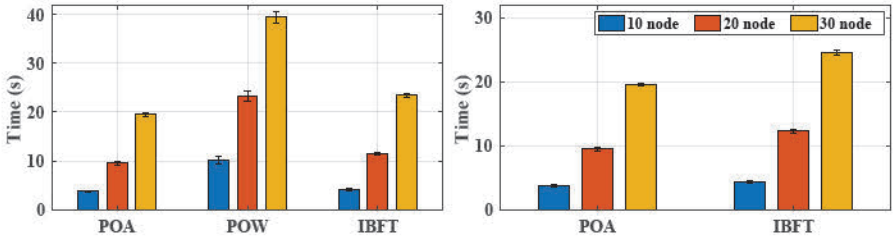
Fig. 2: 10-node blockchain topology for a microgrid

In Fig. 2, our topology consisting of 10 nodes, corresponding to a hypothetical microgrid is shown. We made a realistic assumption that the microgrid has fewer prosumers and auctioneers compared to the consumers. Out of the 10 nodes, there are 5 consumer nodes, 2 prosumers/producers, and 3 miners. We also had a 20 node network with 13 consumers and 4 prosumers, and a 30 node network with 21



(a) with 3 miners (b) with 5 miners

Fig. 3: Node sealing times for a network with 3 or 5 miners



(a) with 3 miners (b) with 5 miners

Fig. 4: Auction completion times for a network with 3 or 5 miners

consumers and 6 prosumers. In our experiments all prosumers were essentially operating as producers.

B. Experiment Results

We define the block sealing time (BST) as the time it takes to commit a block to the chain. The BST for each consensus protocol are shown in Figs. 3a and 3b for 3 and 5 miners, respectively, corresponding to the 10, 20, and 30 node topologies. The standard deviation (SD) of the BSTs for PoA, PoW, and IBFT are 0.28, 1.11, and 0.19 seconds, respectively for the 10 node topology. The SDs for PoA, PoW, and IBFT are 0.30, 1.14, and 0.24 respectfully, for a 20 node topology. Compared to the 10 node, there was a slight increase in SD and the average BST. The BST increased by 0.01, 0.03, and 0.05 seconds, respectively for PoA, PoW, and IBFT. However, when we increased the node size from 20 to 30 nodes, we observed a significant change in the BST. The average BST incresed by 0.48 seconds with the SD increasing by 1.1 seconds, for PoW. This is due to the fact that PoW can only commit a block to the chain once a miner has generated a valid hash which has no time guarantee. For IBFT, there was no change in averages, but the standard deviation decreased by 0.02 seconds. PoA had a decrease in average by 0.02 seconds, and an increase of standard deviation by 0.05 seconds. Since PoA and IBFT produce blocks at a designated period of two seconds, we can see from Figs. 3a and 3b that the BSTs do not vary significantly.

In Figs. 4a and 4b, we show the times required for all the 100 auctions to complete. We observe that the time to complete each auction rises for each consensus protocol with increase in number of nodes, due to the participation of additional entities in the auction. Even though PoA and IBFT have the same block generation period of two seconds, we can see that PoA manages to outperform IBFT in each scenario. This is due to the three rounds of validation that IBFT validators have to do before committing a block to the chain [9], compared to one round needed by the PoA. In each of the three figures, we can also see that PoW takes a significantly longer time to complete each auction. While the BSTs are certainly a factor, a considerable amount of resource on the machine is also taken by the PoW mining nodes, in addition to the resources needed by other docker containers.

We also attempted to increase the number of miners from three to five, but when trying to deploy smart contracts on the PoW network with 10 nodes, the deployment of the contract took longer than 12 minutes which resulted in a failed deployment of the contract. With PoW, the chain could only progress a few blocks before the miner's hash rates fell well below the limits of being able to find a valid hash in a reasonable amount of time. The PoA and IBFT did not experience these issues.

With the limit of 1 GB of RAM on the containers we expected PoW to perform poorly in comparison to PoA and IBFT, but we did not expect that having *just* 5 miners would have such an adverse effect. To see the effect of just one miner, we noticed that the miner had an average hash rate of 28 kH per second; 1 kH/s is the equivalent of $1,000$ hashes per second, meaning that one miner amongst 10 nodes was computing about $28,000$ hashes per second. When looking at the CPU usage of that miner, we saw that it was taking up approximately 180% of the CPU. Since we were running our experiments on a quad-core processor, the reading of 180% means that it was taking up nearly two cores worth of CPU resources. When we scaled PoW to five miners, four of the miners were averaging $6 - 7$ kH/s with one miner averaging

less than 10 H/s.

When looking at the CPU usage for this experiment, each of the miners were averaging 30% usage. This makes sense since we are not using our graphics processing unit (GPU) to mine blocks. CPU mining has a much lower yield compared to GPU mining [14]. The sudden increase in average and standard deviation for PoW in Fig. 3a led us to believe that the machine specifications were insufficient to run three PoW miners along with 27 other docker containers. To confirm this, we moved the PoW test to a machine with better specifications that had a 4.0 GHz quad-core CPU, 64 GB RAM, and an Ubuntu 18.04 LTS operating system. The results in Figure 5 show that when provided with sufficient computing resources, PoW results are more consistent as well as having a faster block sealing time. While there is an overall improvement, PoA and IBFT still perform significantly better than PoW.

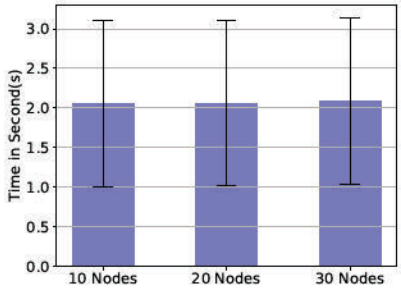


Fig. 5: Block sealing time on improved machine for 3 miners

As shown in Fig. 4b, PoA's time to complete all 100 auctions for each scenario changed very little in comparison to the experiments with three miners. With IBFT, on the other hand, we can see that with the addition of two more miners that the time to complete each auction went up by 0.3, 0.7, and 1.1 seconds on an average, respectively for 10, 20, and 30 auctions. Since all the validators in IBFT need to communicate with each other to remain secure, increasing the number of validators will have a negative impact on performance, which is in accordance with the qualitative research presented in [15].

VI. Conclusion

In this paper, we presented an evaluation of three well-established consensus algorithms for facilitating block chain based P2P energy transactions through double auction mechanisms, in the microgrid. We found that PoW is not a suitable choice for the microgrid energy transactions as the speed of transaction settlements are slow. Since PoW is a computationally intensive algorithm, running multiple miners on a single CPU was not sufficient. From the perspective of scalability, the performance of PoW will continue to degrade with increase in network size, thus increasing the time to complete each auction, making it a sub-optimal choice for conducting double auction energy transactions.

Protocols like PoA and IBFT require a certain degree of trust in their validator set and thus, do not need to rely on computationally intensive operations to prove a blocks validity. While IBFT is noticeably faster than PoW in our tests, it still has scalability issues due to it's three rounds of message exchange with each of it's validators, leading to increased time to add new blocks as the number of validators increases. PoA did not seem to be affected much by the increase of validators or the increase of transactions in the network. Comparing performances on our test networks with 10, 20, and 30 nodes, we found PoA as the most efficient and scalable consensus algorithm to hold double auctions in the microgrids and are easy for deployment in smart meters or HEMS with limited computational resources.

As a future work we intend to conduct a more exhaustive evaluation of other state-of-the-artconsensus algorithms and different market mechanisms by conducting experiments with real-time hardware prototypes.

References

[1] T. Lv and Q. Ai, "Interactive energy management of networked microgrids-based active distribution system considering large-scale integration of renewable energy resources," *Applied Energy*, vol. 163, pp. 408–422, 2016.

[2] M. Andoni, V. Robu, D. Flynn, S. Abram, D. Geach, D. Jenkins, P. McCallum, and A. Peacock, "Blockchain technology in the energy sector: A systematic review of challenges and opportunities," *Renewable and Sustainable Energy Reviews*, vol. 100, pp. 143–174, 2019.

[3] E. Mengelkamp, J. Gärttner, K. Rock, S. Kessler, L. Orsini, and C. Weinhardt, "Designing microgrid energy markets: A case study: The brooklyn microgrid," *Applied Energy*, vol. 210, pp. 870–880, 2018.

[4] S. Thakur, B. P. Hayes, and J. G. Breslin, "Distributed double auction for peer to peer energy trade using blockchains," in *2018 5th International Symposium on Environment-Friendly Energies and Applications (EFEA)*, Sep. 2018, pp. 1–8.

[5] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain, "Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3154–3164, Dec 2017.

[6] J. Wang, Q. Wang, and N. Zhou, "A decentralized electricity transaction mode of microgrid based on blockchain and continuous double auction," in *2018 IEEE Power Energy Society General Meeting (PESGM)*, Aug 2018, pp. 1–5.

[7] P. Vytelingum, D. Cliff, and N. R. Jennings, "Strategic bidding in continuous double auctions," *Artificial Intelligence*, vol. 172, no. 14, pp. 1700–1729, 2008.

[8] T. D. Nguyen and M. T. Thai, "Trustless framework for iterative double auction based on blockchain," in *International Conference on Security and Privacy in Communication Systems*. Springer, 2019, pp. 3–22.

[9] H. Moniz, "The istanbul bft consensus algorithm," 2020.

[10] T. J. V. Machacek, "A quantitative comparison of blockchain algorithms in peer-to-peer energy auctions in the smart grid," Ph.D. dissertation, New Mexico State University, 2020.

[11] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Manubot, Tech. Rep., 2019.

[12] Clique, "Clique poa protocol rinkeby poa testnet," 2017, [Online; accessed 12-July-2020]. [Online]. Available: https://www.github.com/ethereum/EIPs/issues/225

[13] Truffle Suite, "Sweet tools for smart contracts," 2020, [Online; accessed 12-July-2020]. [Online]. Available: https://www.trufflesuite.com/

[14] J. A. Dev, "Bitcoin mining acceleration and performance quantification," in *2014 IEEE 27th Canadian conference on electrical and computer engineering (CCECE)*. IEEE, 2014, pp. 1–6.

[15] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "Pbft vs proof-of-authority: applying the cap theorem to permissioned blockchain," 2018.