IEEE Open Journal of
Intelligent Transportation
Systems

Received 1 April 2020; revised 22 May 2020; accepted 22 June 2020. Date of publication 25 June 2020; date of current version 14 July 2020.
The review of this paper was arranged by Editor P. H. J. Chong.

Digital Object Identifier 10.1109/OJITS.2020.3004870

Blockchain Based Trading Platform for Electric Vehicle Charging in Smart Cities

NOUREDDINE LASLA[1] **(Member, IEEE), MARYAM AL-AMMARI**[1]**,**
MOHAMED ABDALLAH[1] **(Senior Member, IEEE),**
AND MOHAMED YOUNIS (Senior Member, IEEE)

[1]Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar

[2]Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD 21250, USA

CORRESPONDING AUTHOR: N. LASLA (e-mail: nlasla@hbku.edu.qa).

ABSTRACT This paper presents a novel blockchain-based energy trading architecture for electric vehicles (EVs) within smart cities. By allowing local renewable energy providers to supply public charging stations, EV drivers can gain access to affordable energy and optimally plan for their charging operations. For this purpose, we present a smart-contract based trading platform that runs on top of a private Ethereum network. Contrary to existing solutions, we rely on the legacy billing and metering of the existing utility company in order to avoid making major changes to the existing infrastructure. The trading logic, including the auction mechanism, used to exchange energy can be defined in a smart-contract and applied within the platform. We conduct extensive experiments to evaluate the performance of some existing auction mechanisms and the underlying private Ethereum network in supporting the corresponding energy trading transaction load. We develop a virtualization-based simulator for Ethereum and measure both the transaction throughput and latency under different network and workload scenarios. The obtained results have shown that the current Ethereum implementation can support charging requests from EVs during peak hours in very crowded cities, such as Singapore.

INDEX TERMS Electric vehicle, P2P energy trading, auction, blockchain, smart contracts.

I. INTRODUCTION

BLOCKCHAIN is a promising technology that enables trading in a trust-less environment without a need for intermediaries. Thus, blockchains can support business-to-business (B2B), business-to-customer (B2C) and even customer-to-customer (C2C) transactions. It not only reduces costs and mitigates security threats resulting from the dependence on a third party, but also revolutionizes trading and motivates new business models. For the energy sector, blockchains would empower individuals to actively participate in both sides of the market, namely, as an energy supplier and consumer. Within smart-cities, the proliferation of renewable energy technologies, such as solar rooftops, battery storage and electric vehicles (EVs), are creating a new kind of customers, commonly named “prosumers”, with the capability of both producing and managing their own energy consumption. By giving the opportunity to prosumers to directly sell their surplus of energy to others on a peer-to-peer basis [1], blockchain could enable opening and decentralizing the energy market, promote renewable energy generation, reduce carbon footprint and facilitate meeting the growing demand for green energy [2].

In the near future, EVs are expected to be among the highest consumers for green energy. According to the International Energy Agency forecasts, around 120 million of EVs are about to join the roads by 2030 [3]. To satisfy the expected huge demand of energy and allow an easy access to charging stations, many governments are preparing for the future by encouraging investors to install new charging stations in public areas. These charging facilities will be connected to the main power grid and will engage multiples renewable energy sources via different suppliers, including major and small providers [4]. Ideally, EV owners would be able to directly trade energy by selecting the most appropriate provider for their next charging operations. Thanks to the transparency offered by blockchains, such kind of business transactions, known as peer-to-peer (P2P) energy trading [5], is made possible. By removing any kind of intermediaries, the incorporation of blockchain technologies (i) motivates individuals to share their energy surplus with others to charge their EVs, (ii) ensures better availability of clean energy, and (iii) eventually reduces range anxiety [2].

In this paper, we propose an efficient blockchain-based energy trading platform for EV charging where EV owners can request supply directly from renewable energy providers without intermediaries. In our system, an EV owner sends a charging request (preferences) to the blockchain-based trading platform, and active providers respond by sending attractive sealed offers in a form of bids. We employ a smart contract to act as an auctioneer that can apply different trading logic and manages the different users' accounts; particularly the smart contract is responsible of the selection of winners for each submitted request based on, for instance, the second-price, first-price or double auction mechanism, and the validation of a trade by following the metering data collected from both smart-meters sides. Similar to existing electricity billing systems in most countries, the utility company stays involved in the payment collection in our platform. To support peer-to-peer billing, the utility company relies on both the smart meter readings and the validated trades in our platform.

Nevertheless, ensuring a secure blockchain-based trading in a trust-less and decentralized environment comes with a cost of high storage footprint, computation and communication overhead. Specifically, blockchain consists of a set of computers communicating through a peer-to-peer network, maintaining the same copy of the ledger (replicate) and executing the same code (smart-contract), repeatedly. Therefore, it is important to validate the scalability of the system, to verify whether the application requirements can be fulfilled. This paper also assesses scalability by evaluating the transaction throughput and latency metrics. Our experiments have been conducted on the private Ethereum network, which is theoretically deemed to be more efficient compared to the public version.

The contributions of this paper can be summarized as follow:

- Contrary to existing solutions, our energy trading platform leverages the use of the contemporary utility billing system without major changes to the accounting infrastructure and the payment method. This facilitates earlier adoption of peer-to-peer energy trading in practice.
- To ensure an efficient energy allocation, we compare among some existing pricing schemes based on a sealed second-price (Vickrey), first-price, and periodic double auction mechanisms [6] that can be completely implemented on smart contract.
- In order to validate the performance of the proposed trading platform in supporting the growing number of EVs and their corresponding charging requests, we develop a new testing framework for private Ethereum network. The framework is based on lightweight virtualization and, contrary to existing tools, supports dynamic configuration of the size of the network, number of validators, transaction sending rate and the allocated computation power. The framework gauges the scalability of the blockchain network by reporting both the transaction throughput and latency.

The remainder of the paper is organized as follows. Section II discusses the related work. Section III presents our smart-contract energy trading platform. The performance evaluation methodology and the obtained experiment results can be found in Section IV. Finally, Section V concludes the paper.

II. RELATED WORK

In this section, we review the literature for work on using blockchain to enable peer-to-peer energy trading, and tools for evaluating the performance of blockchain networks.

Blockchain use in energy trading: The last five years have witnessed an increased interest in the use of blockchain to enable peer-to-peer trading, in general, [7], [8] and energy sector, in particular, to replace the conventional utility-centric powered delivery approach [9]. Zhang *et al.* [10] summarized existing local P2P electricity trading projects up to 2017 by elaborating on their main focuses and outputs. Examples of these projects include Piclo in the U.K., [11], Vandebron in the Netherland [12], SonnenCommunity [13] and Share & Charge [14] in Germany, and TransActive Grid [15] in the U.S. Their main function is to connect energy suppliers with consumers by providing information, metering and billing services. Despite their common function, these projects have some dissimilarities. First, they have different scale perspective. Some of them, like Piclo and Vandebron, aim at national scale services, while TransActive Grid are limited to regional and Microgrid scale. They also differ in termss of the applications they are targeting. For instance, Sonnen Community focuses on energy storage, TransActive on solar panel, and Share & Charge on EV charging.

On the other hand, significant research efforts have focused on protecting the privacy of EVs' drivers, ensuring anonymous payment, or securing the trading platform. Aitzhan and Svetinovic [16], have presented a token-based decentralized energy trading system to enable peers to perform transaction anonymously and securely, using multi-signatures and anonymous encrypted messaging streams. Based on consortium blockchain technology, a secure energy trading system has been developed in [17]. To reduce the transaction confirmation delay, a credit-based payment scheme is used to support fast and frequent energy trading. Security analysis and numerical results based on a real data set are used to confirm the efficiency of the proposed scheme. In [18], a decentralized security model to enhance the security of trading between EVs and charging piles in the P2P network is proposed. The EV charging management system leverages the use of lightning network [19] and smart contract technologies. Recently, Radi *et al.* [20] have proposed a blockchain-based anonymous payment system to protect the privacy of the EV owners during charging operations. The system leverages the use of trusted financial entities to exchange real currency into digital coins that are provably intractable. To avoid double spending attacks, all the performed payment transactions with their corresponding charging requests and offers are recorded in a consortium blockchain.

Recently, few studies [21], [22], [23] have investigated the use of blockchain to decentralized the energy auction market. Kang *et al.* [21] have proposed a pricing mechanism to balance local electricity demand using Plug-in Hybrid Electric Vehicle (PHEV). The pricing and the amount of traded electricity among PHEVs are determined by an iterative double auction scheme to maximize the social welfare. A consortium blockchain is established on local energy aggregators to audit, verify and secure transaction records without relying on a trusted third party. The iterative double auction mechanism, however, is still carried out by a third energy broker off the blockchain. Similarly, Mengelkamp *et al.* [22] have designed a local energy market on a private Ethereum blockchain. The market is based on a double auction mechanism with discrete market closing times that is implemented via a smart-contract. However, insufficient details have been provided about the smart-contract implementation and how the price is cleared at each trading period. In [23], Ping *et al.* present a blockchain-based decentralized EVs charging coordination scheme. The scheme consists of two main steps: (i) the charging power quotas are first allocated equitably between the charging stations, and then (ii) using a double auction mechanism, charging stations trade their allocated quotas with each other to satisfy stations with strict demand. The proposed system is implemented on a smart-contract to realize a trustworthy, transparent and decentralized trading platform. In order to efficiently implement the auction mechanism on blockchain, some effort has exclusively focused on ensuring bid confidentiality and securing the auction in decentralized environment [24], [25], [26]. Either zero-knowledge proof protocols or a trusted execution environments (TFEs) technology have employed for such a purpose.

To the best of our knowledge, no prior work has considered the decentralization of the energy market with minimal changes to the existing infrastructure for efficient EVs charging over blockchain. In contrast, our solution is adapting a private blockchain network to build trust between users for energy trading, while keeping the utility company for providing metering and billing services.

Blockchain performance evaluations: Here, we review prior work on evaluating the performance of private blockchain. In [27] and [28], Dinh *et al.* present a framework to evaluate Ethereum and Hyperledger private blockchains. The experiments are conducted by setting a constant transaction sending rate while changing both the number of users and the number of nodes. However, the evaluation of Ethereum used only the Proof of Work (PoW) consensus and not considered the private Proof of Authority (PoA) algorithm. Another similar study in [29] has evaluated the performance of both Hyperledger Fabric and Ethereum in terms of execution time, latency, and throughput while varying the number of transactions from 1 to 10000. Nevertheless, the analysis has been performed using only a single-node network, with a disabled consensus mechanism.

In [30], the performance of Quorum blockchain [31] is evaluated in termss of throughput and latency using different workload. The size of the network was fixed at three nodes, and both Raft and IBFT [32] consensus algorithms were considered in the evaluation. The experiment showed that the throughput scale linearly with the sending rate for IBFT and Raft with a very slight difference between them. Nasir *et al.* [33] have studied the performance of the two versions of Hyperledger fabric v0.6 and v1.0, in termss of latency, execution time and throughput. The experiments have been conducted by varying the workload up to 10,000 transactions, and the size of the network up to 20 nodes. The results show that Hyperledger fabric v1.0 outperforms v0.6 for all the evaluated metrics.

Nevertheless, the aforementioned studies do not provide a complete evaluation of the different parameters affecting the system performances, such as scaling the number of nodes and transactions workload, as well as varying the computational power. In addition, the private consensus algorithm version for Ethereum, PoA, was not evaluated in the previous work. In this paper, we consider the evaluation of the de facto smart-contract platform, Ethereum, in private mode by enabling PoA.

III. SMART-CONTRACT BASED P2P ENERGY TRADING FOR EV CHARGING

In this section, we describe our smart-contract based energy trading platform that enables efficient charging of EVs. We first provide an overview of the system architecture, followed by the different protocols for bidding, purchase agreement and billing. Although the proposed energy trading platform is mainly architected to efficiently support EV charging, it can be adapted to suit other peer-to-peer trading applications both in the energy and other sectors. For example, our system can be used for electricity trading to fulfill the needs of local households and industry because it requires only minor changes to the existing infrastructure. Nonetheless, this paper specifically focuses on meeting the requirements of EV charging applications and develops the necessary smart-contract, consensus algorithm and blockchain performance optimization for such an application. For other use-cases, the solution should be carefully redesigned with respect to the particular requirements of each case.

A. SYSTEM ARCHITECTURE

As shown in Fig. 1, our system is composed of the following five entities: electric vehicle (EV), energy provider (EP), smart meter (SM), utility company and consortium

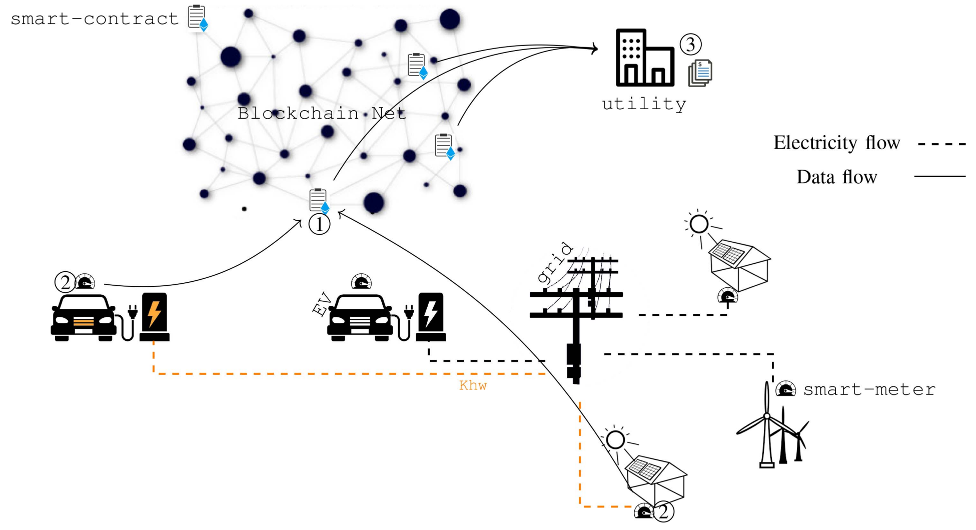


FIGURE 1. System architecture: Blockchain-based energy trading platform for EV charging.

blockchain. In the following, we discuss the role of each entity in achieving the design goals of the system.

Electric vehicle: The EV continually interacts with the system by: (i) requesting energy to charge the vehicle, and then (ii) selecting the most appropriate offer among the submitted bids. This interaction can be done manually by the EV's owner through a Web/mobile application, or an autonomous built-in vehicle bidding agent that automatically predicts and plans for optimal charging operations.

Energy provider: The EPs participate in the charging operation of EVs by powering the charging stations using their locally produced renewable energy. The EPs are indirectly connected to the existing public charging stations through the grid network. An EP can be a small producer, such as a home owner with solar panels placed at rooftop, or a company that is managing more sophisticated renewable energy resources, e.g., wind farms.

Smart meter: Both EV and EP are equipped with a SM to report their consumed and produced energy, respectively. The SM is an essential element of the system allowing to track the energy transfer between an EV and EP with respect to the corresponding pre-established energy purchase agreement. Each smart-meter has a unique identifier, a pair of public and private keys, and needs to be associated with a user account.

Utility: The utility company is responsible of preparing the bills for the EVs and EPs based on the smart meter readings and the performed purchase agreements on the energy trading platform. The utility company manages the grid network infrastructure including the deployed SMs, and ensures their security and correct function. The utility company also maintains the list of the registered users with their associated smart meters. For each energy exchange (transaction) that happens on the grid network, the utility company will receive corresponding fees, which represents the main economic incentive for the utility company to participate in the proposed energy trading system.

Blockchain: At the core of the system is a consortium blockchain network which facilitates transactions between EVs, EPs, SMs and the utility company. More precisely, the blockchain will manage a smart-contract that defines the trading business logic, including the *auction mechanism* and *billing*. The role of blockchain is to provide a single point of truth between all parties without relying on a trusted third party. Basically, blockchain consists of a set of blocks where each contains a set of transactions. The network agrees on a set of transactions to add in a block using a consensus algorithm. The consensus algorithm ensures that only valid transactions can be appended to the blockchain in the same order across the network. A smart contract is a piece of code that is deployed into the blockchain nodes and triggered by transactions stored in a blockchain. The blockchain also stores the new states of the smart contract every time it is executed. In an EV charging scenario, the smart-contract defines the energy trading logic and is executed every time a new transaction to buy or sell energy is sent by EVs and EPs, respectively. The smart contract will behave logically based on its code (EV charging trading logic) and will guarantee that all the agreed-upon terms and conditions will be fulfilled. Overall, the smart-contract can help in improving the accountability, interoperability, and credibility of all participants in such a decentralized environment.

We opt for a consortium (permissioned) blockchain for two main reasons. First, unlike public blockchains, no cryptocurrency or token is needed in private blockchains, which simplifies its adoption, especially that the use of cryptocurrencies is restricted in many countries. Second, a private blockchain provides better performances in termss of throughput and latency compared to most public blockchains. Furthermore, in order to extend the blockchain network and ensure a more secure platform, we propose a reward mechanism to incentify registered users who already have accounts with the utility company to join the network and become sealers. For this purpose, we assume that small fees will be taken from users for each transaction they perform. The fees will be included in the user electricity bills. Similarly, the authority nodes (sealers) will be paid by cutting from their electricity bills an amount calculated based on the number of blocks that they create. Note that part of these fees will go to the utility company managing the grid infrastructure. It is also important to note that it is not recommended in any case that EVs themselves run the blockchain nodes for two main reasons; (1) they have limited computation and storage resources that cannot handle a large number of transactions, and (2) they are mobile and battery-powered, so they cannot guarantee to be always available online."

In summary, the proposed platform interconnects all the parties involved in energy trading in a seamless, secure, and cost-effective manner. Without making major changes to the existing accounting and payment infrastructure, the new platform introduces blockchain into the core business of energy trading to satisfy EVs charging requests in a completely transparent and peer-to-peer fashion. Moreover, the platform makes use of existing smart-meters to actually report the electricity exchange between traders in a fully automated way. Finally, to perform billing, the utility company needs to simply read the performed trades by the user's account in the blockchain.

B. AUCTION MECHANISM

Through an auction mechanism, the system can achieve some important properties, such as social welfare and incentive compatibility, meaning that both EVs and EPs are willing to report their true valuation of the energy to buy and sell, respectively. In the electricity market, ensuring incentive compatibility is of paramount importance where EVs can have different charging needs in termss of urgency level, amount and arrival time. In the following, we assume that there are sufficient charging stations so no scheduling is needed to manage EV access to the charging service. This is a reasonable assumption as most countries are expanding their EV support infrastructure and continually deploying new charging stations in residential, commercial and public areas. We also assume that local renewable energy generation will emerge and be widely adopted in the future to satisfy the demands. Based on this assumption, we employ a reverse auction mechanism, where the traditional roles of buyers and sellers are inverted. Thus, buyers (EVs) are the ones to send their charging requests and sellers (EPs) compete against each other to gain the right to sell their surplus by submitting bids. The charging cost will typically be lowered as EPs underbid each other. In the following, we consider the second-price auction mechanism [34] as an example, and describe how it is implemented in our system.

1) REQUESTS

The bidding process is initiated by the EV owners who send requests to buy electricity for charging their vehicles, as illustrated by step 1 in Fig. 2. Motivated by the EV requests, the EPs start bidding by making offers to sell their locally produced electricity (Fig. 2 step 2). Each EV $i \in M$ can express its charging preference as $\alpha_i = (t_i, q_i, v_i, a_i)$, where t_i represents the time when the EV prefers for the charging to take place. q_i denotes the demanded amount while v_i is the EV valuation per unit of electricity (kWh). The valuation v_i can be expressed as the maximum *price* the EV i is ready to pay. Finally, a_i is the *auction time*, after which no offer from suppliers will be accepted. To ensure incentive compatibility and prevent EVs from misreporting their preferences, we made α_i public so each EV is forced to put its true valuation (the maximum price it can actually pay) of the electricity to maximize its chance for getting a timely charging service. For instance, if an EV owner is in urgent need of charging the vehicle, the only way to ensure getting quick service, especially when renewable energy resources are limited, is to announce willingness to pay a high price. This will enables prioritizing customers (EVs) based on their need, and helps them in devising effective strategies for selecting the charging time and cost.

2) OFFERS

In response to an EV request, α_i, an EP $j \in S$ makes an offer $\beta_{ji} = (\alpha_i, p_{ji})$, where $p_{ji} \leq v_i$ is the offered *price*. During the bidding period, EPs send their sealed offers, so that no one can uncover what other EPs are offering. All offers are only revealed at the end of the auction time a_i. A sealed bid is a hash version of the corresponding offer: $hash(\beta_{ji})$. After the termination of bidding period, EPs reveal their offers by sending their actual bids (Fig. 2 step 4); the smart-contract validates that the calculated hash values are the same as the ones provided during the bidding period (sealed bid). We employ a hash function to protect the bids since it is a computationally effective technique and suitable for smart contract execution environment. More details about the execution complexity of the hash function as well as its impact on the overall system performance are provided in Section IV.

3) SECOND LOWEST-PRICE BID

To select the winner w (Fig. 2 step 6) for a charging request α_i, while ensuring incentive compatibility between EPs, we employ the second lowest-price strategy. Basically, the winner is the one offering the lowest price, but the amount to pay is the price of the second-lowest bid p'_{iw}. This strategy gives EPs an incentive to reduce the electricity cost by telling their true minimum value, and tends to increase the social welfare of both the EPs and the EVs [6].

C. PURCHASE AGREEMENT

The auctioneer (smart-contract) determines the EP offering the lowest price and a purchase agreement will be established

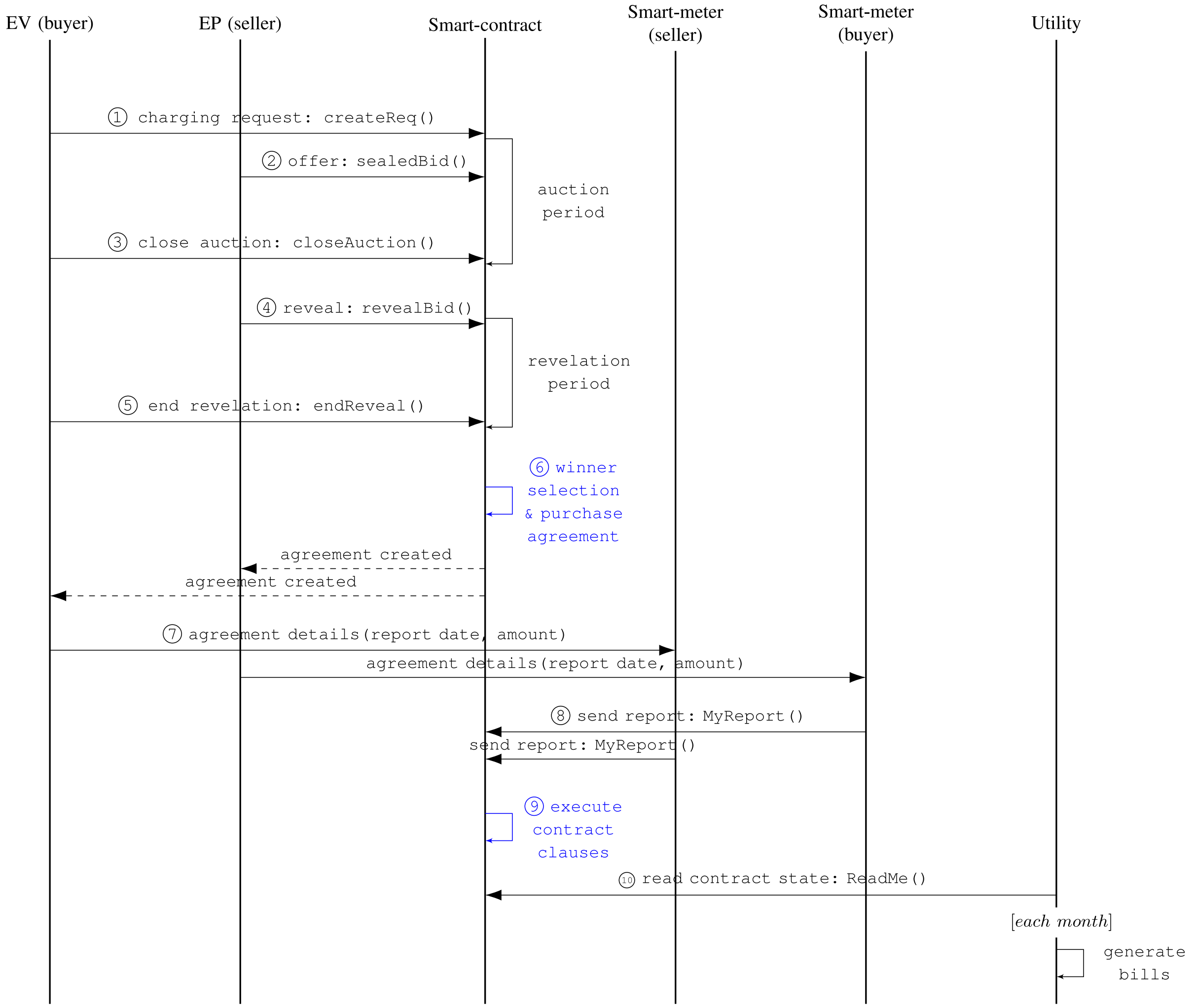


FIGURE 2. Peer-to-peer energy trading scheme for EVs charging sketch.

(Fig. 2 step 6) between the EV and the winner. Both the SMs of the corresponding EV and EP will be notified about the established purchase agreement (Fig. 2 step 7) to further report the actual state of the physical energy transfer (Fig. 2 step 8). Each time the SMs send their reports, the smart contract self executes to update the state of each of the EV and EP accounts (Fig. 2 step 9). For instance, if the SM's report of the EP indicates a non complete transfer of the electricity, the smart contract will not accept any future offers from that EP until the remaining energy is injected into the grid.

D. BILLING

As discussed before, the billing is done by the utility company which already manages the grid network, and the different smart-meters devices and accounts. Thus, no change is needed for the existing infrastructure and the payment method. The utility company can be also viewed as trusted third party with respect to money transfer from EV to EP. Because the EPs are using the grid network, the utility company can charge them based on their usage that is recorded in the smart-contract. To generate bills, the utility company needs to consider both the smart meter readings and all purchase agreements recorded in the blockchain (Fig. 2 step 10). The final bill of each customer, both EV and EP, will be calculated based on the agreed price and the reported consumed/provided energy by the corresponding smart meter.

IV. PERFORMANCE EVALUATION

A. AUCTION PERFORMANCE

For simulating the presented EV charging auction mechanism, we consider a number of energy providers (EPs) with a surplus of energy that they want to sell to EV customers through the smart-grid. Each EP has a surplus in a range of 70kWh and 210kWh. The minimum prices of the EPs to sell their surplus are picked randomly in the interval of [10, 50] unites per kWh. The maximum prices that EVs' owners are willing to pay to charge their vehicles are chosen randomly from an interval of [15, 60] unites per KWh. The charging request of each EV is selected randomly between 20kWh and 60kWh. For the simulation, we neglect the cost of using

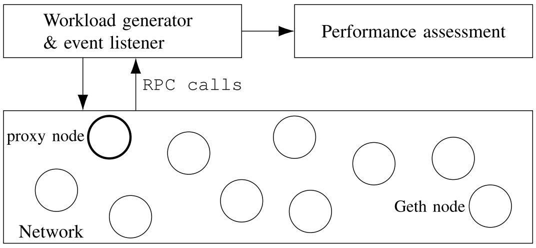


FIGURE 3. Performances evaluation methodology architecture.

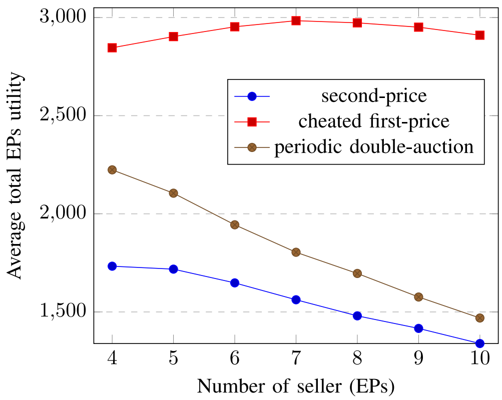


FIGURE 4. Total average EPs utility as the number of seller (EPs) varies for 10 buyers.

the smart-grid network. The simulation results are obtained as the average over all possible random values for the different parameters (prices, charging requests, energy surplus) using $10,000$ independent simulation runs.

Fig. 4 shows the average total EPs utility, that reverse auction usually aims to maximize, when varying the number of EVs (buyers) for a set of 10 EPs (sellers) using the lowest second-price auction, the lowest-price auction by cheating in the price and periodic double auction. The EP utility can be defined as the difference between the minimum offered and actual selling price. Contrary to the first and second-price auction, periodic double auction gathers all requests and offers for a pre-known period of time before clearing the market. In the simulation, we sort all the requests (resp. offers) in a decreasing (resp. increasing) order of their respective bids. We then generate both the supply and the demand curve and consider the intersection point for determining the selling price and clear the market accordingly. More details about how to clear the market price while ensuring truthfulness can be found in [35]. Cheating in the first-price auction is simulated by adding to the EP minimum selling price a random value between [1, 20].

As we can see from the figure, the EPs utilities in the case of double auction is better compared to the second lowest price auction. However, because the lowest-price or first-price auction is not incentive-compatible [34], bidders (EPs) tend to increase their minimum selling price and hide the true valuations (minimum prices) of their energy surplus in order to boost their profit. Such behavior negatively affects the utility of the EVs and tends to increase the charging cost. On the other hand, making EV charging requests public motivates EVs owners to tell their true valuation of the maximum price they are willing to pay for charging their vehicle and also improve EPs utilities as more charging operations can take place. In summary, the simulation results show how the auction mechanism can affect the utility of EPs, specifically selecting the period double auction mechanism and making the EVs request public lead to better performance.

B. BLOCKCHAIN PERFORMANCE

In this section, we describe the implementation of our proposed trading platform on a private Ethereum blockchain, and the evaluation of its performance in termss of throughput and latency. The validation environment consists of three modules: the network module, workload generator and data listener module, and performance assessment module, as shown in Section IV-B and explained below.

1) NETWORK MODULE

This module allows the creation of a test network by configuring a number of Ethereum nodes using Docker container technology [36]. Each node runs Geth, the GoLang-based Ethereum client [37], with or without active mining option. The network runs the Geth implementation of the PoA consensus algorithm, named Clique [38]. The idea behind PoA is to allow a set of authority nodes (sealers) to create (seal) new block in a mining rotation scheme. At each epoch of time, one sealer is elected to create new blocks. In case the elected sealer delays or fail in submitting the block, some other sealers can take its place and propose another block instead. We refer the readers to [39] for a detailed description of the algorithm. In PoA network, it is also possible to control the block generation time in second, by configuring the *period* value in the genesis file.

In our test network, we use a proxy node (with mining option disabled) to serve as a gateway between the network and the workload generator, and the data listener module. It is responsible of receiving transactions from the workload generator module and broadcasts them back into the network. The node also notifies the data listener module every time a new block gets confirmed.

2) WORKLOAD GENERATOR & DATA LISTENER MODULE

This module is implemented using a JavaScript API that communicates to our proxy node through RPC calls. It basically sends the configurable workload to the network, and listens for block confirmation events. The workload is configured by specifying the total *number of transactions* (*#tx*) to send as well as the *sending rate* ($\#tx/sec$). The module continuously listens to the blockchain network, and for every new generated block, the following block-related attributes are recorded: *block number*, *validation time* and the *number of transactions* within the block.

TABLE 1. Gas cost for different functions of the EV-charging contract.

premitive	exec. cost	transaction cost
`createReq()`	165552	187784
`sealedBid()`	85147	88923
`revealBid()`	50758	72478

3) PERFORMANCE ASSESSMENT MODULE

At the end of the experiment, this module collects measurements, i.e., takes the saved information from the data listener module and calculates both the throughput and latency as follow:

- The throughput, or transactions per second (TPS), can be seen as the average number of confirmed transactions per block divided by the block time, or the total number of transactions divided by the time needed to validate all of them (time between the block containing the first transaction and the block containing the last transaction)

  $$\text{TPS} = \text{total } \#tx/(\text{last BlockTime} - \text{first BlockTime})$$

  where BlockTime is the block validation time.
- Transaction latency reflects the time difference between the instant the transaction is sent and the time it gets validated by the blockchain network. This metric is computed as an average per transaction and is calculated as the total time to process N transactions divided by N.

C. EXPERIMENT

We have implemented our EV charging contract that includes the second-price auction mechanism using Solidity, which is the de facto scripting language for writing Ethereum smart contracts. The pseudo code of the main used functions specifically, sending requests, making sealed offers, revealing offers and winner selection, are given in Algorithm , and . We made the complete implementation of our smart-contract available on GitHub,[1] for the community to review and leverage. The associated execution and transaction cost in unites of Gas are also given in Table 1. According to the table, the execution cost of *createReq*() is higher than the other functions as it consist of using SLOAD and LOG instructions to store a new request into memory. These instructions consume significantly more Gas unites compared to a hash function. For instance, with respect to the Ethereum yellow paper [40], SLOAD and LOG consume 200 and 375 Gas unites, respectively, while Keccak256 (hash function), which is equivalent to SHA3, consumes only 30 unites of Gas.

1) REAL WORLD EV CHARGING EXPERIMENT

In order to validate and test our architecture, we first set up a prototype for Ethereum-based P2P energy trading, as illustrated in FIGURE 5. The prototype system is established using a Docker machine to emulate an Ethereum network, Android devices (tablets) running EP and EV client applications, and Raspberry Pi devices to emulate the respective smart meters of both the EP and EV. The EV and EP mobile applications are developed using the Android Studio environment. The EV mobile application instantiates transactions by sending EV charging requests to the Ethereum network. The EP mobile application lists the available charging requests from the Ethereum network and makes charging offers (bids). The smart contract (auctioneer) selects the best offer for each request and establish the respective contracts. The smart meters run a JavaScript program that uses Web3 library [41] to listen to the established contracts and send electricity exchange reports to the Ethereum network. Based on the reports sent by the two smart meters, the smart contract settles the trading and updates the accounts of the EV and EP.

Algorithm 1 Pseudo Code for Create Charging Request

```
enum AgreState {NotCreated, Created, HasOffer,
    Established, ReadyForPayement, ReportNotOk}
struct SealedBid {
       address bidder;
       bytes32 bid; //hash of the bid
       }
struct Auction {
       uint nbBid;
       SealedBid[] bids; //Set of received offers
       }
struct Agreement {
       address buyer; // EV address
       address seller; // Winner EP address
       uint  amount;
       uint  buyerMaxPrice;
       uint  price;
       Auction auction;
       AgreState state;
   }
mapping (uint => Agreement) public agreements;

function createReq(amount, price, time, aucTime)
 {
   uint aucId = totalAuction++;
   //Store to Auction and initiate a new agreement
   storeAndLogNewReq(msg.sender, aucId, amount,
       price, time, aucTime,);
 }
```

Algorithm 2 Pseudo Code for Make Sealed Offer

```
function makeSealedOffer( aucId, sealedBid)
   auctionExisit(aucId)
   auctionNotClosed(aucId)
   revealNotEnded(aucId)
 {
   agreements[aucId].auction.bids.push(SealedBid(
       msg.sender, sealedBid));
   agreements[aucId].auction.nbBid ++;
 }
```

2) SIMULATION EXPERIMENT

In order to evaluate the throughput and latency of our trading platform based on a private Ethereum blockchain, we conduct several experiments under different workloads and network sizes. In the experiment, we test the platform by calling the function *createReq() as it is the most costly*

---

1. https://github.com/noureddinel/SecondLowestPriceAuction

Algorithm 3 Pseudo Code for Reveal Bid and Winner Selection

```
function revealOffer (aucId, price, bidId)
   auctionExisit(aucId)
   auctionClosed(aucId)
   revealNotEnded(aucId)
 {
   if (agreements[aucId].auction.bids[bidId].bid
       != keccak256(price)) {
       // Bid not correctly revealed.
       return;
   }
   if (agreements[aucId].state == greState.
       HasOffer)
   {
       require(price < agreements[aucId].price);
       //better offer received, keep previous
           price (second lowest price)
       agreements[aucId].seller = msg.sender;
   } else {
       // first offer received
       require(_price <= agreements[aucId].
           buyerMaxPrice);
       agreements[_aucId].price = price;
       agreements[_aucId].seller = msg.sender;
       agreements[_aucId].state = AgreState.
           HasOffer;
       }
 }
```

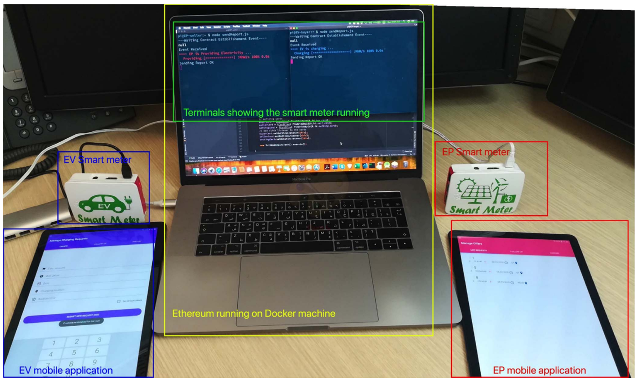


FIGURE 5. Real blockchain-based peer-to-peer energy trading experimental setup for EV charging.

transaction. Four workload types are used by varying the transaction sending rate: 100, 300, 500 and 1000 tx/sec. For the network size, a blockchain with 1, 5, 10 and 20 sealers are considered. The block time (time between two consecutive blocks) is fixed and set to 2 seconds in the genesis file. The total number of transactions to be sent by the workload generator module is also fixed and set to 10, 000. The experiments are conducted on a workstation machine with Intel Xeon Gold 6130 CPU, 2.10 GHz, 64 core CPU, 256GB RAM, and running Ubuntu 18.04.2. A summary of the different parameters used to conduct the different experiments are given in Table 2.

TABLE 2. Experiment parameters.

Parameters	Value
Net. size	[1, 5, 10, 20] sealers
Sending rate	[100, 200, 300, 500, 1000] tx/sec
Workload	10,000 tx
Geth version	1.9.9
Consensus	Clique (Ethereum PoA)
Block Time	2 seconds
Sim. environment	Intel(R) Xeon(R), 2.10 GHz, 64 core CPU, 256GB RAM, Ubuntu 18.04.02

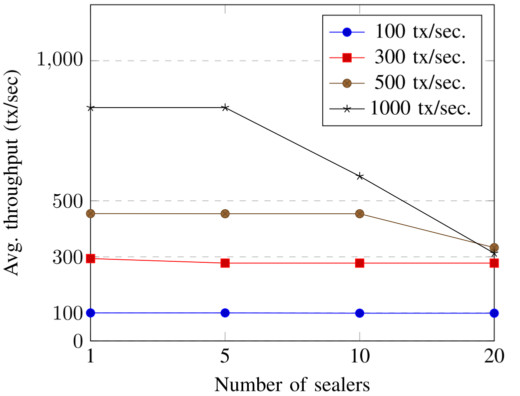


FIGURE 6. Throughput Vs. Number of sealers.

D. RESULTS

Fig. 6 illustrates the average transaction throughput under different sending rates and number of sealers. For moderate rates; 100 and 300 tx/sec, the system achieves its max throughput and all transactions get processed and added to blockchain at, approximately, the same sending rate regardless the size of the network. For higher rates (500 and 1000 tx/sec.), the throughput is significantly affected by the number of sealers, and drops below 50% when scaling up to 20 sealers of a sending rate of 1000 tx/sec. Fig. 7 confirms the previous results, where the latency is inversely proportional to the throughput. For a moderate sending rate or a small network size, the average transaction latency is between 2.5 and 3.5 seconds. Given the block time, which is set to 2 seconds, and the time needed to propagate a block through the network, this is a reasonable latency. However, a higher sending rate and a bigger network size cause the latency to increase significantly and exceed 10 seconds. This can be attributed to two reasons: (i) a higher sending rate produces larger block sizes, thus more time is needed in order to propagate the corresponding volume of data to all the sealers; (ii) in Ethereum, each transaction needs to get validated (check if it is properly signed), propagated to the entire network, executed to update the smart-contract state, included into a new block, and then propagated again to the network to be executed by the other sealers. Clearly, the handling of transactions will be affected by the available computation resources, and consequently the validation delay will grow for higher sending rates. The results also show that the maximum throughput that the current private Ethereum implementation (Geth version 1.9.9) can support is 350 tx/sec.

Fig. 8 and 9 illustrate better the impact of computation power on the transaction throughput and latency. We repeat

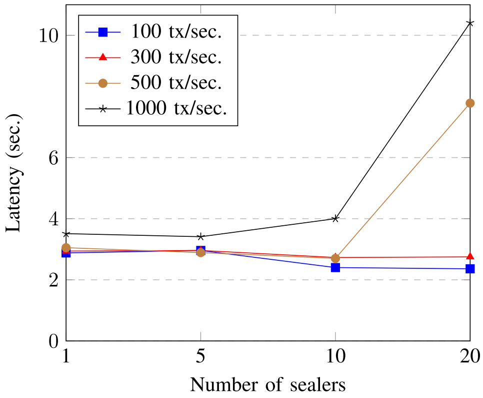


FIGURE 7. Transaction latency Vs. Number of sealers.

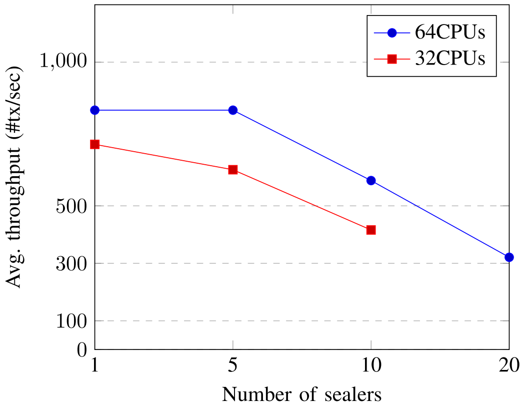


FIGURE 8. Throughput Vs. Number of sealers (1000 tx/sec).

the same experiment for a sending rate of 1000 tx/sec, but this time we limit the number of used CPUs to only 32. The results clearly show the negative impact of the limited computational power on the overall system throughput and latency. In case of 20 sealers with 32 CPUs, it can be noticed that the plot is not reporting any result. After further debugging, we find out that only few transactions get processed and included within new blocks in this case, whereas the remaining transactions get stuck forever in the sealers' transaction pool with pending status. Similar behavior has been observed when we increased the sending rate above 1000 tx/sec with 64 CPUs. We believe that this is a bug in the Go Ethereum implementation that is experienced every time extensive transaction processing is required. The same conclusion has been also reached in [42] and [43].

In order to show how the limited computation power also impacts the size and time of the block, we report in Fig. 10, 11 and 12 the number of transactions per block for sending rates of 200, 500 and 1000, respectively. For a moderate sending rate of 200 tx/sec with 5 sealers (Fig. 10(a)), most of generated blocks are uniform in size and receive the same number of transactions (400), which is equivalent to the maximum possible throughput given that the block time is set to 2 sec. However, with the increase in the number of sealers and the sending rate, block sizes become more and more irregular. For instance, in Fig. 12(c), blocks number 2 and 3 are empty, whereas block number 4 contains more than 4000 transactions. To confirm this observation, we plot in Fig. 13 the standard deviation of block sizes versus the number of sealers for different sending rates. It is clearly shown that the deviation from the mean grows for larger network sizes and sending rates. Higher sending rates incur significant delay in the process and propagation of transactions to the entire network, and would affect the generation time of new blocks.

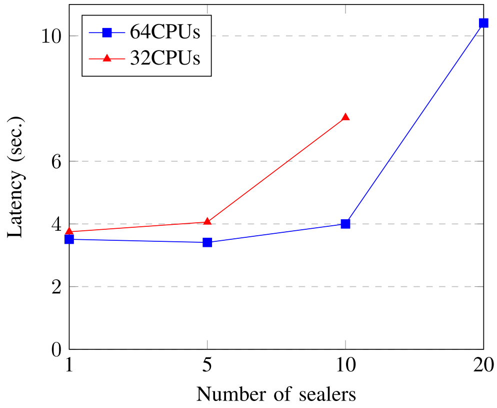


FIGURE 9. Latency Vs. Number of sealers (1000 tx/sec).

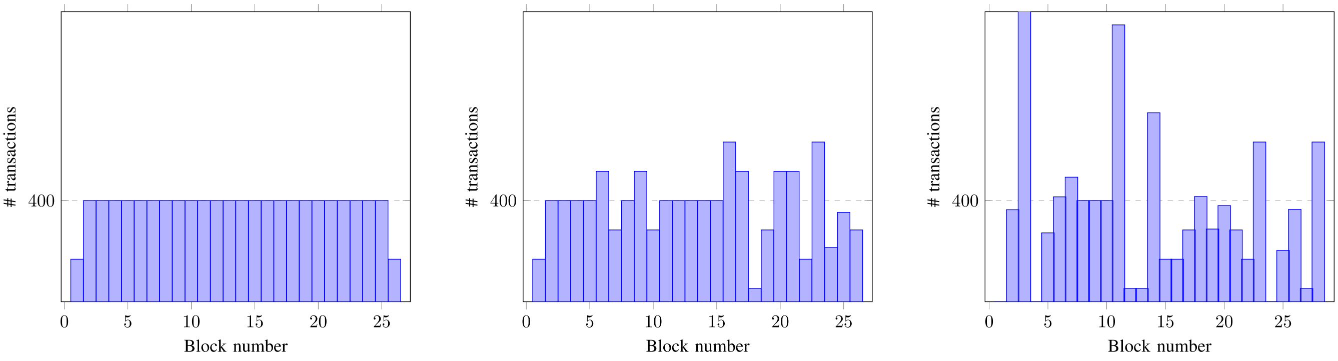
E. PRACTICAL CONSIDERATION

The obtained results have shown that the current Ethereum implementation allows our energy trading platform to reach transaction throughput of about 350 tx/sec. For a major city like Singapore, the study of Wang *et al.* [44] has shown that the average charging requests by each EV are about 0.34 and 0.17 a day for EVs with a driving range of 180km and 350km, respectively. Assuming the 180km's driving range, which generates more charging requests, and considering the total number of vehicles in Singapore is $600,000$, the charging amount is $204,000$ a day. If all these requests are to be submitted within the same hour of a certain day, we will have about 56 requests/sec. Consequently, the current implementation can satisfactorily support EV charging requests from crowded city in peak hours.

From the presented results we can also conclude that the limited computational power constitutes a serious bottleneck that can dramatically diminish the performance of the Ethereum blockchain in terms of throughput and latency. Therefore, the selection of adequate computation resources for blockchain nodes should be carefully considered during the design.

V. CONCLUSION

In this paper, we have presented a smart-contract based energy trading platform atop private Ethereum network. Our platform enables local energy providers to directly trade their energy surplus with EV owners in a P2P manner. In order

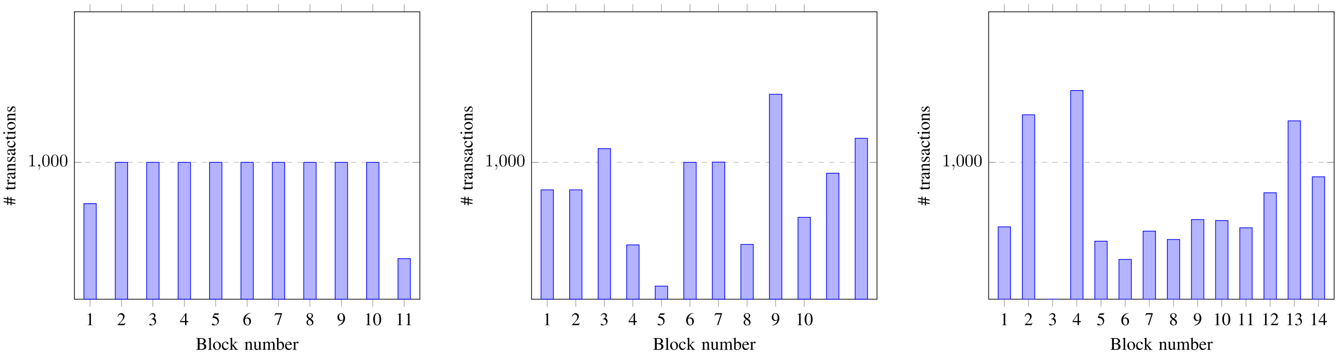


(a) 5 sealers

(b) 10 sealers

(c) 20 sealers

FIGURE 10. Number of transactions within each block. The sending rate = 200 *tx*/*sec*.

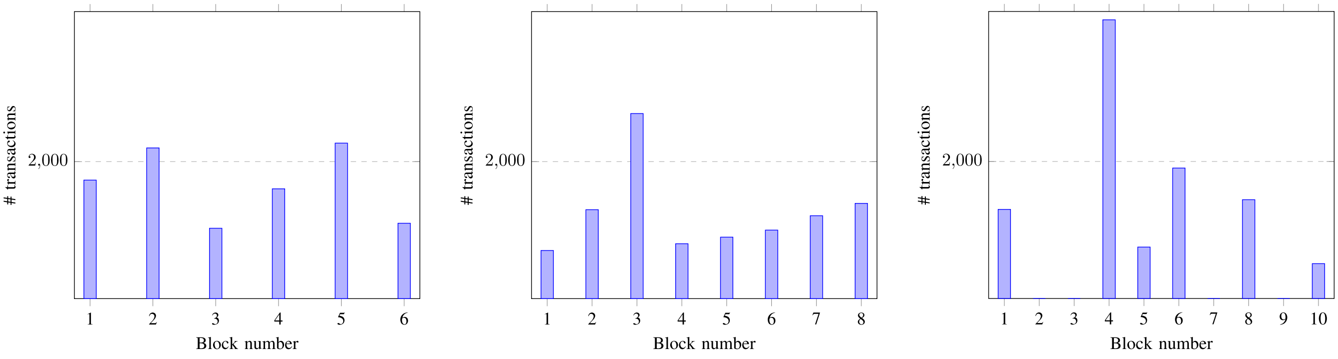


(a) 5 sealers

(b) 10 sealers

(c) 20 sealers

FIGURE 11. Number of transactions within each block. The sending rate = 500 *tx*/*sec*.



(a) 5 sealers

(b) 10 sealers

(c) 20 sealers

FIGURE 12. Number of transactions within each block. The sending rate = 1000 *tx*/*sec*.

to avoid making major changes to the existing infrastructure, our platform continues to rely on utility companies for metering and billing. In addition, our platform employs a second lowest-price auction mechanism to ensure both social welfare and incentive compatibility.

To assess the scalability of the trading platform in supporting a large number of EV charging transactions, we have conducted several experiments and evaluated the throughput and latency under different workload, network size, and computational power. The validation results have shown that good performance could be achieved under moderate workload. However, in the case of high transactions rates, the performance gets significantly degraded. Deep investigation on the issue allowed us to conclude that the computational resources are a potential bottleneck for Ethereum platform, since executing a large number of transactions involves heavy computation load. Nevertheless, our current implementation can support charging requests from EVs during peak hours in very crowded cities, like Singapore. We have also studied the performance of some auction mechanisms that can be used

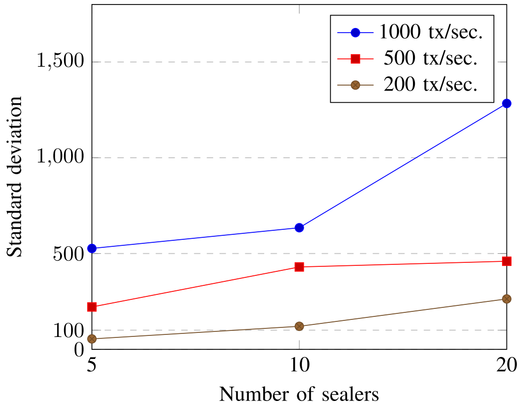


FIGURE 13. Standard deviation for the number of transactions per block.

by our trading platform and specifically evaluated the social welfare for each of them. In summary, we believe that our proposed platform is a viable and practical solution that promotes local energy generation by enabling fare and effective P2P trading. Our testing and experiment results, also serve as a guideline to assess the scalability and efficiency of the system before deployment.

In the future, we plan to extend our solution to sustain the privacy of EVs and EPs while ensuring an efficient auction mechanism. Information related to charging amount, date, and location as well as the profits made by EPs, should not be viewed by unauthorized entities. Particularly, we envision the use of zero-knowledge proof to be a promising technique for that purpose.

REFERENCES

[1] W. Tushar, T. K. Saha, C. Yuen, D. Smith, and H. V. Poor, “Peer-to-peer trading in electricity networks: An overview,” *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3185–3200, Jul. 2020.
[2] N. U. Hassan, C. Yuen, and D. Niyato, “Blockchain technologies for smart energy systems: Fundamentals, challenges, and solutions,” *IEEE Ind. Electron. Mag.*, vol. 13, no. 4, pp. 106–118, Dec. 2019.
[3] *World Energy Outlook 2018*, IEA, Paris, France, 2018. [Online]. Available: https://www.oecd-ilibrary.org/content/publication/weo-2018-en
[4] W. Tushar, C. Yuen, S. Huang, D. B. Smith, and H. V. Poor, “Cost minimization of charging stations with photovoltaics: An approach with EV classification,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 156–169, Jan. 2016.
[5] W. Tushar, C. Yuen, H. M. Rad, T. K. Saha, H. V. Poor, and K. L. Wood, “Transforming energy networks via peer-to-peer energy trading: The potential of game-theoretic approaches,” *IEEE Signal Process. Mag.*, vol. 35, no. 4, pp. 90–111, Jul. 2018.
[6] J. Levin. (2004). *Auction Theory*. [Online]. Available: www.stanford.edu/jdlevin/Econ
[7] J. Xie *et al.*, “A survey of blockchain technology applied to smart cities: Research issues and challenges,” *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2794–2830, 3rd Quart., 2019.
[8] J. Qiu, D. Grace, G. Ding, J. Yao, and Q. Wu, “Blockchain-based secure spectrum trading for unmanned-aerial-vehicle-assisted cellular networks: An operator’s perspective,” *IEEE Internet Things J.*, vol. 7, no. 1, pp. 451–466, Jan. 2020.
[9] G. Koutitas, *The Smart Grid: Anchor of the Smart City*. Cham, Switzerland: Springer, 2018, pp. 53–74.
[10] C. Zhang, J. Wu, C. Long, and M. Cheng, “Review of existing peer-to-peer energy trading projects,” *Energy Procedia*, vol. 105, pp. 2563–2568, May 2017.
[11] *Piclo Website*. Accessed: 2020. [Online]. Available: https://piclo.uk/
[12] *Vandebron Website*. Accessed: 2020. [Online]. Available: https://vandebron.nl/
[13] *Sonnen Website*. Accessed: 2020. [Online]. Available: https://sonnen.de/
[14] *SHARE & CHARGE*. [Online]. Available: https://shareandcharge.com
[15] E. Mengelkamp, J. Gärttner, K. Rock, S. Kessler, L. Orsini, and C. Weinhardt, “Designing microgrid energy markets: A case study: The Brooklyn microgrid,” *Appl. Energy*, vol. 210, pp. 870–880, Jan. 2018.
[16] N. Z. Aitzhan and D. Svetinovic, “Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams,” *IEEE Trans. Depend. Secure Comput.*, vol. 15, no. 5, pp. 840–852, Sep./Oct. 2018.
[17] Z. Li. J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, “Consortium blockchain for secure energy trading in industrial Internet of Things,” *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3690–3700, Aug. 2018.
[18] X. Huang, C. Xu, P. Wang, and H. Liu, “LNSC: A security model for electric vehicle and charging pile management based on blockchain ecosystem,” *IEEE Access*, vol. 6, pp. 13565–13574, 2018.
[19] J. Poon and T. Dryja. (2016). *The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments*. [Online]. Available: https://lightning.network/lightning-networkpaper.pdf
[20] E. Radi, N. Lasla, S. Bakiras, and M. Mahmoud, “Privacy-preserving electric vehicle charging for peer-to-peer energy trading ecosystems,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–6.
[21] 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 Trans. Ind. Informat.*, vol. 13, no. 6, pp. 3154–3164, Dec. 2017.
[22] E. Mengelkamp, B. Notheisen, C. Beer, D. Dauer, and C. Weinhardt, “A blockchain-based smart grid: Towards sustainable local energy markets,” *Comput. Sci. Res. Develop.*, vol. 33, nos. 1–2, pp. 207–214, 2018.
[23] J. Ping, S. Chen, Z. Yan, H. Wang, L. Yao, and M. Qian, “EV charging coordination via blockchain-based charging power quota trading,” in *Proc. IEEE Innovat. Smart Grid Technol. Asia (ISGT Asia)*, Chengdu, China, 2019, pp. 4362–4367.
[24] H. S. Galal and A. M. Youssef, “Verifiable sealed-bid auction on the ethereum blockchain,” in *Proc. Int. Conf. Financ. Cryptography Data Security*, 2018, pp. 265–278.
[25] H. S. Galal and A. M. Youssef, “Trustee: full privacy preserving vickrey auction on top of ethereum,” in *Proc. Int. Conf. Financ. Cryptography Data Security*, 2019, pp. 190–207.
[26] E.-O. Blass and F. Kerschbaum, “Strain: A secure auction for blockchains,” in *Proc. Eur. Symp. Res. Comput. Security*, 2018, pp. 87–110.
[27] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, “Untangling blockchain: A data processing view of blockchain systems,” *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1366–1385, Jul. 2018.
[28] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K.-L. Tan, “BLOCKBENCH: A framework for analyzing private blockchains,” in *Proc. ACM Int. Conf. Manag. Data*, New York, NY, USA, 2017, pp. 1085–1100.
[29] S. Pongnumkul, C. Siripanpornchana, and S. Thajchayapong, “Performance analysis of private blockchain platforms in varying workloads,” in *Proc. 26th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Vancouver, BC, Canada, Jul. 2017, pp. 1–6.
[30] A. Baliga, I. Subhod, P. Kamat, and S. Chatterjee, “Performance evaluation of the quorum blockchain platform,” 2018. [Online]. Available: https://arxiv.org/abs/1809.03421.
[31] J. P. M. Chase. *A Permissioned Implementation of Ethereum*. Accessed: Feb. 20, 2018. [Online]. Available: https://github.com/jpmorganchase/quorum
[32] M. Pietro. (2018). *Distributed Ledger Technologies Consensus Mechanisms*. [Online]. Available: https://ssrn.com/abstract=3389871
[33] Q. Nasir, I. A. Qasse, M. A. Talib, and A. B. Nassif, “Performance analysis of hyperledger fabric platforms,” *Security Commun. Netw.*, vol. 2018, Sep. 2018, Art. no. 3976093.
[34] M. S. Robinson, “Collusion and the choice of auction,” *RAND J. Econ.*, vol. 16, no. 1, pp. 141–145, 1985.
[35] D. Friedman, *The Double Auction Market: Institutions, Theories, and Evidence*. Oxfordshire, U.K.: Routledge, 2018.

[36] *Docker: Enterprise Container Platform*. Accessed: 2020. [Online]. Available: https://www.docker.com

[37] (Nov. 2019). *Go-ethereum*. [Online]. Available: https://github.com/ethereum/go-ethereum/wiki/geth

[38] *Proof of Authority*. Accessed: 2020. [Online]. Available: https://eips.ethereum.org/EIPS/eip-225

[39] S. D. Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain," in *Proc. Ital. Conf. Cybersecurity*, 2017, p. 11.

[40] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Ethereum, Zug, Switzerland, Yellow Paper, 2014.

[41] *Ethereum Javascript API*. Accessed: 2020. [Online]. Available: https://github.com/ethereum/web3.js/

[42] H.-T. Hsieh. *Geth Not Broadcasting Transactions*. Accessed: Jun. 5, 2018. [Online]. Available: https://medium.com/getamis/geth-not-broadcasting-transactions-1a881c50dafc

[43] V. Ruzans. *JPM Organ's Quorum Blockchain Performance Testing*. Accessed: Aug. 13, 2018. [Online]. Available: https://blog.scandiweb.com/article/jpmorgan-s-quorum-blockchain-performance-testing-hfn1asbjf8v

[44] H. Wang, D. Zhao, Q. Meng, G. P. Ong, and D.-H. Lee, "A four-step method for electric-vehicle charging facility deployment in a dense city: An empirical study in Singapore," *Transp. Res. A, Policy Pract.*, vol. 119, pp. 224–237, Jan. 2019.

NOUREDDINE LASLA (Member, IEEE) received the B.Sc. and M.Sc. degrees in computer science from the University of Science and Technology Houari Boumediene (USTHB) and the Superior Computing National School (ESI), in 2005 and 2008, respectively, and the Ph.D. degree in computer science from the USTHB in 2015. He is currently a Postdoctoral Research Fellow with the Division of Information and Computing Technology, Hamad Bin Khalifa University, Qatar, with expertise in distributed systems, network communication, and cyber security.

MARYAM AL-AMMARI received the master's degree from the College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar. Her main research interests include cybersecurity and networking.

MOHAMED ABDALLAH (Senior Member, IEEE) received the B.Sc. degree (Hons.) from Cairo University, Giza, Egypt, in 1996, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, MD, USA, in 2001 and 2006, respectively. In 2006, he joined Cairo University, where he holds the position of an Associate Professor with the Electronics and Electrical Communication Department. He is currently an Associate Professor with Hamad Bin Khalifa University, Doha, Qatar. His current research interests include the design and performance of physical layer algorithms for cognitive networks, cellular heterogeneous networks, sensor networks, smart grids, visible light and free-space optical communication systems, and reconfigurable smart antenna systems.

MOHAMED YOUNIS (Senior Member, IEEE) received the Ph.D. degree in computer science from the New Jersey Institute of Technology, USA. He is currently an Associate Professor with the Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County. He was with the Advanced Systems Technology Group, an Aerospace Electronic Systems Research and Development Organization, Honeywell International, Inc., where he led multiple projects for building integrated fault tolerant avionics and dependable computing infrastructure. He also participated in the development of the redundancy management system, which is a key component of the vehicle and mission computer for NASA's X-33 space launch vehicle. He has published over 200 technical papers in refereed conferences and journals. He has six granted and two pending patents. His technical interest includes network architectures and protocols, wireless sensor networks, embedded systems, fault tolerant computing, secure communication, and distributed realtime systems. He serves/served on the editorial board of multiple journals and the organizing and technical program committees of numerous conferences. He is a Senior Member of the IEEE Communications Society.