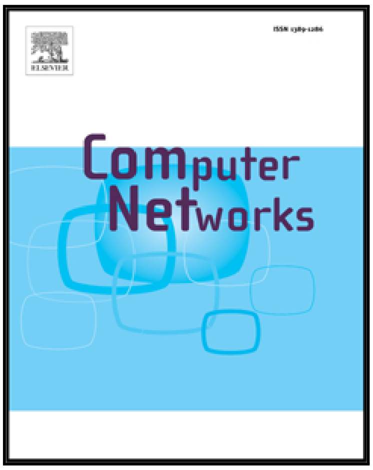


SURVIVOR: A Blockchain based Edge-as-a-Service Framework for
Secure Energy Trading in SDN-enabled Vehicle-to-Grid Environment

Anish Jindal, Gagangeet Singh Aujla, Neeraj Kumar

PII: S1389-1286(18)31106-X
DOI: https://doi.org/10.1016/j.comnet.2019.02.002
Reference: COMPNW 6728

To appear in: *Computer Networks*

Received date: 23 October 2018
Revised date: 21 January 2019
Accepted date: 4 February 2019

Please cite this article as: Anish Jindal, Gagangeet Singh Aujla, Neeraj Kumar, SURVIVOR: A Blockchain based Edge-as-a-Service Framework for Secure Energy Trading in SDN-enabled Vehicle-to-Grid Environment, *Computer Networks* (2019), doi: https://doi.org/10.1016/j.comnet.2019.02.002

SURVIVOR: A Blockchain based Edge-as-a-Service Framework for Secure Energy Trading in SDN-enabled Vehicle-to-Grid Environment

Anish Jindal[a], Gagangeet Singh Aujla[b], Neeraj Kumar[c,*]

[a]*School of Computing & Communications, Lancaster University, Lancaster, UK*
[b]*Computer Science & Engineering Department, Chandigarh University, Mohali, India*
[c]*Computer Science & Engineering Department, Thapar Institute of Engineering and Technology (Deemed to be University), Patiala, India*

Abstract

Electric vehicles (EVs) have transformed the smart transportation sector by providing diverse energy management solutions to the smart grid. Energy trading among EVs and charging stations (CS) in a vehicle-to-grid (V2G) environment is one of the popular verticals in smart grid. However, processing the energy trading decisions at remote control centers lead to an increase in delay and network overhead. Apart from these issues, the security concerns while trading the energy in such an environment remain persistent. Therefore, to handle the aforementioned issues, this paper presents *SURVIVOR*: A Blockchain based Edge-as-a-Service Framework for Secure Energy Trading in software defined networking (SDN)-enabled V2G Environment. In the proposed framework, the energy trading decisions are processed closer to the location of EVs through edge nodes. Moreover, for securing the energy trading transactions, blockchain is used wherein the approver nodes are selected amongst all the present nodes on the basis of a utility function and are made responsible for validating the transactions. Once such nodes are selected, a consensus-based blockchain mechanism for secure energy trading in SDN-enabled V2G environment is presented. In this mechanism, edge nodes are responsible for generating proof-of-work puzzles. The proof-of-work is

---

[*]Corresponding Author

Email addresses: `anishjindal90@gmail.com` (Anish Jindal), `gagi_aujla82@yahoo.com` (Gagangeet Singh Aujla), `neeraj.kumar@thapar.edu` (Neeraj Kumar)

a unique hash value which is computed for each EV and the transactions for which the approver nodes compute the same proof-of-work for each EV are added in the blockchain. The complete scheme is backed by the SDN architecture to reduce the overall latency and increase the throughput of the smart transportation network. The results obtained prove that the proposed scheme is effective for trading the energy between EVs and CS while securing the underlying trading transactions using blockchain. Moreover, the communication and computation cost of the proposed scheme comes out to be small which proves that it can be used in real-world applications. The latency in the complete transportation sector is also greatly reduced by using the SDN-architecture.

Keywords: Blockchain, Edge-as-a-service, Energy trading, Smart city, Software defined networking.

1. Introduction

In the recent times, transportation sector has undergone a global transformation with the emergence of smart cities, thereby making it a smart transportation system. This system comprises of vehicles enabled with information and communication technologies (ICT) to provide various automated services such as-navigation, traffic control, parking, and information sharing. The communication technologies used for inter-vehicle communication are short to medium range communication protocols like IEEE 802.15.1, IEEE 802.15.4 , IEEE 802.11 a/b/g/p; and for vehicle to service provider include long range communication protocols like IEEE 802.16, Long Term Evolution, etc. [1]. These vehicles may belong to the conventional fuel-based vehicles or the upcoming electric or plug-in electric vehicles (EVs).

The emergence of EVs has led to the drastic transition in the smart transportation sector. EVs have thrived in this transition as they provide manifold benefits apart from the transport services. The primary benefit of EVs in the smart transportation sector is that they can be used for energy sustainability of a smart city [2]. Moreover, the EVs are eco-friendly vehicles as they produce near-zero tapline carbon emissions and have less operating cost than the conventional fuel vehicles [3]. Because of these advantages, the number of EVs on the roads are exponentially increasing. The global trend in growth of EVs is shown in Fig. 1 and according to the Global EV outlook 2018, there will be more than 20 million EVs by 2020 [4]. The EVs are also an essential component in managing the energy needs of the smart city as the primary source of their fuel is energy. Therefore, one needs to manage their energy requirements in such a way that these EVs can be used as a carrier for self-sustainability in the smart cities. For example, if there is surplus energy in the city, this energy can be bought by the EVs and when the city faces energy deficit, EVs can supply the energy back to the system. Moreover, mismanagement of the energy demands of a large number of EVs can lead to power fluctuations in the smart city [5]. This problem is more prominent in developing countries where energy generation sources are limited, but the demand is increasing day by day. Therefore, one needs to leverage the vehicle-to-grid (V2G) capabilities of EVs to strike a balance between energy demand and supply.

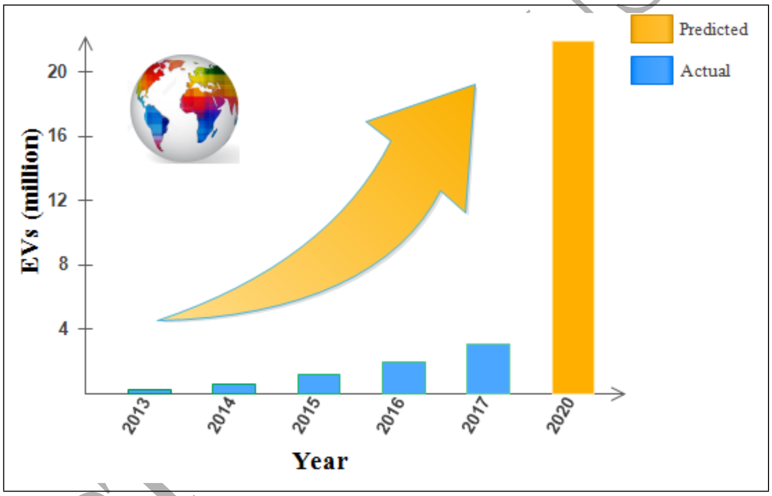


Figure 1: Growth of EVs globally [4].

However, there are many issues pertaining to the energy trading with EVs which restrict their participation in the V2G activities for energy management. For instance, the energy trading decisions has to made at remote control center, which in turn lead to delay and network overhead. However, in real-time communications, it is always preferable to handle the computational processing closer to the location of the user/EVs. Moreover, the time required to reach the destination (where trading would take place) and its distance from their present location are few of the factors which hinder the energy trading process. Moreover, with the advances in technology, a new computing paradigm has emerged which can provide services at a rapid pace at a location closer to the end-users. This paradigm is known as edge computing and is considered closer to the end-user as compared to the traditional cloud computing architecture. Thus, edge can provide services relatively faster to the cloud which means that the quality of service is invariably enhanced.

Apart from these aspects, trust and security is always highly desirable. So, when the EVs trade for energy (to make a profit/fulfill their energy demands) over a remote server, the security concerns remain utmost importance to make a successful transaction. The incoming requests in the network should be authenticated prior to broadcasting for taking the energy trading decisions. Moreover, with the increased usage of ICT for communicating the energy trading requests, the security of the overall system needs to be robust. In this system, the adversaries, if present in the network, should not be able to modify the transaction for their personal benefits. Moreover, implementing the conventional security mechanisms on the distributed edge nodes should be carefully considered so as not to jeopardize the overall security of the system. An adversary may act as edge node and impact the entire energy trading mechanism thereby causing loss to both EVs and CSs. Thus, the security mechanisms should be distributed in nature (so that no single entity is responsible for the system security) and lightweight (so that these do not put an additional burden on the system).

In this regard, blockchain has emerged as one of the most prominent technique to meet the security requirements of the system in an effective way [6, 7]. One of the most popular examples where blockchain has been primarily used for providing the security is the exchange of crypto-currency known as Bitcoin [8]. The blockchain is distributed in nature and does not depend on the single trusted third party to authenticate the incoming requests in the system [9]. So, it even works well in the scenarios where the centrally managed security schemes fail. Blockchain is a ledger-based mechanism which takes care of different security constraints related to confidentiality, integrity, and authentication. All the entities present in network maintains a ledger for keeping the information of all the valid transactions executing in the network. Whenever a new transaction arrives in the network, the entities present in the network authenticate it by solving a puzzle (also know as proof-of-work). So, the transaction is only added in the blockchain if the puzzle is correctly solved and authenticated by rest of the nodes.

While implementing the security and serving the energy trading requests in the smart transportation sector, the traditional network architecture suffers from the various drawbacks. Firstly, these are not able to dynamically re-configure the network topology for reducing the latency in the network. Secondly, the network becomes overloaded when incoming requests increase as there is a fixed path for transmitting the data packet. Third, the networking functionality is mainly implemented on the devices such as switches and routers in a dedicated manner, so it is very difficult to update each device to make the network consistent in terms of updating policies. Therefore, to handle all these issues, a new networking paradigm has emerged which is capable of dynamic routing of packets and adapt to the changing network requirements. This paradigm is known as software-defined networking (SDN) where the complete system is divided into number of planes with each plane performing a specific task [10]. These planes are namely data, control, and application. All the forwarding devices are placed at the data plane, the control logic of the network is implemented at the control plane, and the applications such as packet routing, request forwarding, etc. are handled at the application plane.

To handle all the aforementioned issues, *SURVIVOR*: A Blockchain based Edge-as-a-Service Framework for Secure Energy Trading in SDN-enabled V2G Environment is presented in this paper. In this framework, the EVs select an edge node (from all available edge nodes) on the basis of utility functions. This selected edge node serve the EVs for handling their energy trading computational load. Using this edge-as-a-service platform, EVs trade for the required energy with the charging stations (CSs) deployed in the smart city. This platform significantly reduces the burden on EVs which have less computational resources and makes the complete process faster and on the edge of the network. Being a distributed and insecure layout, a consensus-based blockchain mechanism is used to authenticate the energy trading transactions in the system. The edge nodes (other than the one responsible for computation load of an EV) act as approver nodes for transaction authentication. The underlying paradigm follows openflow protocol to provision dynamic network policies in the distributed edge-as-a-service platform.

1.1. Contributions

The following are the major contributions of the proposed research work.

- An SDN based system model is designed for edge-as-a-service platform which follows openflow protocol to provision dynamic network policies in the distributed setup.

- An edge-as-a-service platform is selected on the basis of different utility functions for energy trading between EVs and CSs in SDN-enabled V2G environment. An efficient algorithm responsible for convergence to an optimal solution is also designed.
- A consensus based blockchain mechanism is designed for securing the energy trading transactions in distributed edge-as-a-service environment. In this scheme, the edge nodes other than the one selected to handle the computational workload of energy trading mechanism act as approver nodes.

1.2. Organization

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 illustrates the complete system model of the scheme and Section 4 describes the proposed scheme in detail. Section 5 outlines the simulation results and the paper is concluded in Section 6.

2. Related work

The related work section is divided into various categories with emphasis on smart transportation and energy trading, providing edge-as-a-service, use of SDN in energy trading, and blockchain applications in energy trading.

2.1. Smart transportation and energy trading

The research in smart transportation sector has been given much emphasis, now more than ever, with the evolution of smart cities. Many of the research works have focused on providing transportation solutions related to the traffic management. For example, Li *et al.* [11] applied the intelligent transportation system in Macao for enhancing the quality of traffic operations in the city. The authors used deep belief network and support vector regression to forecast the traffic characteristics. The results prove that the error in prediction for their model to predict the traffic congestion was tolerable and it helped in learning the traffic behavior in Macao in an effective way. ICT has also become an integral part of smart transportation which forms the communication backbone of the complete system. Zhang *et al.* [12] presented an enhanced security and mobility mechanism by leveraging the ICT and autonomous driving. The authors modeled the traffic flow in a smart city which augers well with the real world traffic behavior to understand the impact of various scenarios related with traffic.

The use of EVs in smart transportation sector has made it possible to balance the load demand in the smart cities without depending only on the energy supplied by the grid or the use of renewable sources of energy. The energy from EVs can be exchanged with various types of users in the smart cities such as residential, commercial, and industrial to manage their load demands. The authors in [13] surveyed multiple distributed energy trading concepts in the smart city environment. The authors noted that the use of renewable energy sources and EVs would play a vital role in mitigating the energy requirements of the local communities. They also noted that the majority of the energy trading architectures were based on game theoretic approach, objective maximization, and simulation-based scenarios. Using the concept of game theory, the authors in [14] proposed a contract-based game for energy trading between energy suppliers and consumers so as to increase the incentive for both of these. EVs are a major player in energy trading because they can act as mobile energy carriers to suffice the energy requirements of an area. One of such survey has been conducted by Shuai *et al.* [15] which studies the economically led schemes for charging of EVs in a smart city environment.

To cater to some the research challenges in the local electricity market, the authors, in our previous work [2], have utilized the EVs as a service to meet the local load requirements in a smart city. The EVs act as buyer when a locality has excess energy while EVs were used as sellers when there was an energy requirement in the locality. This provides enough employment opportunities for EV owners to make a profit while the proposed scheme alleviates the need to build new infrastructure to manage the load in the city. However, the security requirements in the energy trade were not considered in this paper. Apart from these, several energy trading projects are being studied by the researchers to understand the dynamic of users in a realistic trading scenario. The review of several peer to peer energy trading projects was carried out in [16]. The authors in this paper noted that the local energy market structure in microgrids was largely ignored in those projects while trading the energy. Moreover, the authors noted that it is necessary to have a strong communication backbone in the energy network and the concept of blockchain can be introduced in this market for simplifying and securing the peer to peer energy trading transactions.

2.2. Edge-as-a-service

The edge computing compliments the cloud computing environment by serving the requests more close to the users. For instance, Aujla *et al.* [17] proposed a workload slicing scheme to serve the big data processing requests in multi edge-cloud environment. The authors focused on providing the optimal traffic scheduling of the incoming requests so that the network takes minimum energy. Apart from this, Carvallo and Cooper [18] emphasized on the role of edge computing in the smart grid to provide flexibility in communication and computation while serving the user requests. The edge computing has been widely used in applications where fast computation is required. The authors in [19] designed an energy management scheme for data centers for sustainability using renewable energy sources wherein the computation load of data centers was shifted to the edge nodes. The incoming user requests were classified on the basis of delay sensitivity and the delay sensitive jobs were routed to the edge node. This brought down the overall load on the cloud DC and provided the services closer to the user.

Kumar *et al.* [20] presented a vehicular delay-tolerant network to handle the data generated in the smart grid on the basis of the mobile edge computing. This scheme used 'store and forward' data management scheme to manage the large data generated from various types of devices in the smart grid environment for reducing the overall transmission delay and fasten the response time. Bao *et al.* [21] moved one step further and presented a fog computing-based solution for data offloading between various mobile devices in a seamless manner in time-sensitive applications. The authors named their proposed solution as 'Follow me cloud' which represented that the computations can be done on the fly while making the data offloading process smooth. All these aforementioned schemes prove that the edge computing can be effective in providing faster solutions to the end-users as compared to traditional computing services.

2.3. SDN in energy trading

To provide a re-configurable communication backbone, SDN has emerged as one of the most prominent technology in the recent times. It has been widely studied by the research community in the energy market for its effectiveness. The authors in [22] explored the smart city from the viewpoint of enhancing the quality of experience for the user. For this purpose, the city's architecture was divided into two regions with respect to the data plane and control plane of the city. The authors pointed out that the design and implementation of the control plane is a difficult task since a large number of devices are added for seeking the services in smart cities. To this end, SDN paradigm can help in catering the scalability requirements of the network to provide the underlying services because it can be reconfigured on the fly.

Working in this direction, Rego *et al.* [23] proposed an efficient control mechanism for managing the traffic in the smart cities during emergency scenarios. The authors minimized the delay time while processing the request received from IoT devices during emergency situations. This was done by dynamic re-routing of traffic and controlling the request for resources to modify the routes. Zhong *et al.* [24] used SDN in the peer to peer energy transactions on large scale to improve the flexibility and efficiency. The hierarchical architecture controls the energy network in a way that the energy flow in peer to peer trading was flexible and the control plane was dynamically able to reconfigure the network according to needs of the users. Their results validated the effectiveness of their proposed scheme as the delay in transmission was very low. On the similar lines, Lu *et al.* [25] also presented a SDN-based communication framework for managing the energy-related transaction in the smart energy environment. In our previous work [10], we have used SDN-architecture to manage the energy requirements of data centers. In this paper, all the energy related devices like renewable sources, EVs, and data centers have been placed in data plane, all the control algorithms and modules were placed in control plane with the help of which, the user can access all the services through the application plane. However, the aspects related to delay minimization and security were not considered in this paper.

2.4. Blockchain in energy trading

The blockchain is a relatively newer technology which has been recognized worldwide to provide the security in a distributed manner. It has been used in wide variety of application ranging from crypto-currency to energy trading in smart cities. Mannaro *et al.* [26] have started a crypto-trading project to explore the integration of smart contracts in blockchain and the energy trading in the Sardinia Region for smart grid management while preserving the user privacy. The authors allocated different crypto-currency profiles to different users to suggest best selling strategy and manage the sale and purchase of energy in a decentralized energy market. Aggarwal *et al.* [27] proposed a blockchain-based scheme for secure data transmission and access in smart grid environment. The authors selected some entities present in the smart grid network as miner nodes and used these to validate the energy trading transactions in the network. For this purpose, a proof of work was computed by the miner nodes which was mapped to the hash function generated by the associated entity. Dorri *et al.* [28] presented a private blockchain-based scheme to remove the dependency on a trusted third party by using atomic meta-transactions for preserving the privacy while trading the energy. A meta-transaction was created when both the participating entities in the energy trade commit within the given time, otherwise, the transaction was invalidated and trade was suspended.

Apart from these, few of the authors have specifically focused on using the concept of blockchain for addressing the security requirements with respect to EVs which participate in the energy trade [29, 30]. The authors in [29] used the concept of energy coins to attract the EVs for taking part in stabilizing the local electricity market and make secure energy trading using blockchain. The authors used auction game theory to further solve the problem of trading the energy in order to maximize the overall social welfare in the market while protecting the privacy of the participating EVs and securing the energy transactions. Similar to this scheme, the charging and discharging strategies of the EVs for energy trading were modeled on the basis of blockchain in [30]. However, all these schemes neglected the delay to process the requests in the energy market which is one of the major factors to improve the quality of service in the electricity market.

After analyzing all the above discussed proposals, Table 1 depicts the comparative analysis of the existing proposals.

Table 1: Comparative analysis of existing proposals

Authors	**Technique used**	**V2G**	**ET**	**Sec**	**BC**	**EC**	**SDN**
Mannaro *et al.* [26]	Cryto-trading scheme	×	✓	✓	✓	×	×
Aggarwal *et al.* [27]	Energychain scheme	✓	✓	✓	✓	×	×
Dorri *et al.* [28]	Blockchain-based scheme	×	✓	✓	✓	×	×
Kang *et. al* [29]	Energy coins-based scheme	✓	✓	✓	✓	×	×
Liu *et. al* [30]	Blockchain-based scheme	✓	✓	✓	✓	×	×
Aujla *et. al* [2]	Electric vehicles-as-a-service	✓	✓	×	×	×	✓
Aujla *et. al* [31]	Demand response optimization	×	✓	×	×	×	✓
Shuai *et al.* [15]	EV charging scheme	✓	✓	×	×	×	×
Aujla *et al.* [17]	Big data migration scheme	×	×	×	×	✓	✓
Zhang *et al.* [14]	Contract-based energy trading	✓	✓	×	×	×	×
Aujla and Kumar [19]	MenSuS scheme	✓	✓	×	×	✓	✓
Zhong *et al.* [24]	Peer to peer energy trading	×	✓	×	×	×	✓
Lu *et al.* [25]	SDN-based energy trading	×	✓	×	×	×	✓
SURVIVOR	The proposed scheme	✓	✓	✓	✓	✓	✓

V2G: Vehicle to Grid, ET: Energy Trading, Sec: Security, BC: Blockchain, EC: Edge Computing, SDN: Software Defined Networking.

3. System Model

The comprehensive architecture of the proposed scheme is depicted in Fig. 2. There are primarily two participating entities in this architecture namely EVs and CSs. The EVs move to different CSs located at different locations on the basis of the various parameters such as distance, pricing, etc. which give the maximum profit to the EVs. This architecture is based on SDN and is responsible for providing the computation, control, and communication capabilities to all the EVs and CSs. In addition to it, SDN-enable architecture also minimizes the delay and improves the response time to serve the requests generated by the EVs or the CSs.

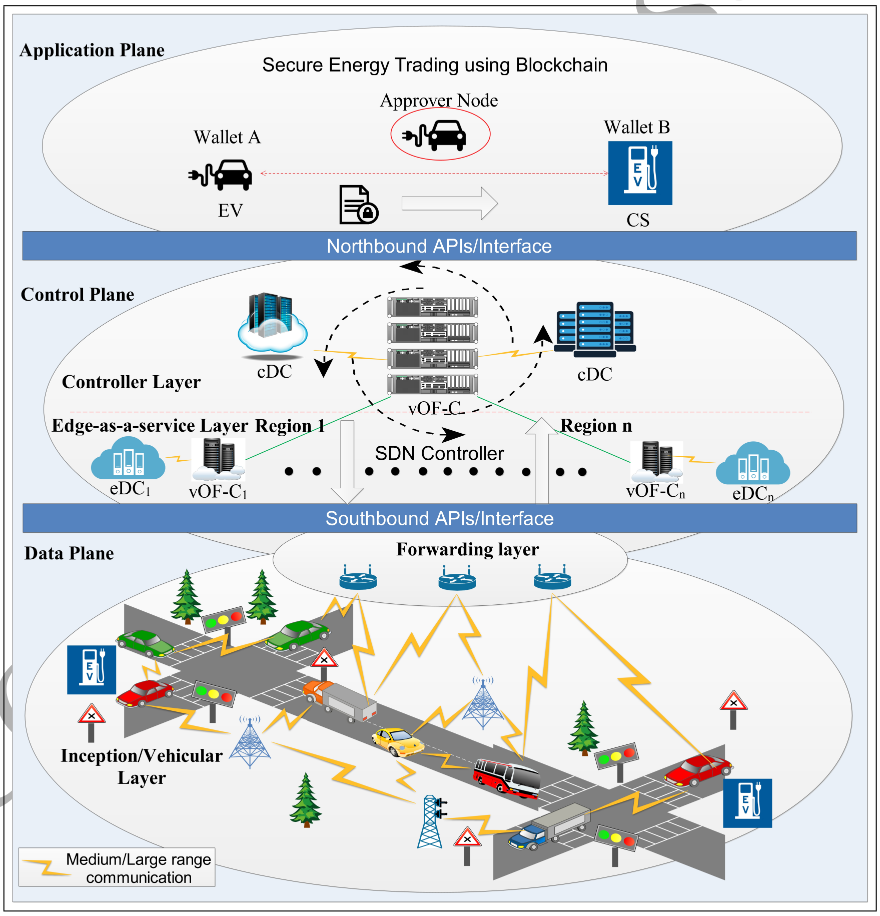


Figure 2: SDN-enabled system architecture for secure energy trading.

The traditional TCP/IP protocol suite is not suitable for this architecture as all the application and control tasks are coupled with each other which makes it difficult to update the dynamically changing network policies. In contrast to this, SDN provides complete flexibility as the application, data, and control are de-coupled from each other which makes it more adaptable to the changing network policies. Moreover, SDN is an autonomous approach because of the use of high level programming languages like Python, Java, C++ etc. to handle the complete network topology with efficient management of requests from mobile vehicles. SDN uses virtual control scheme to manage the network resources in an efficient way to provide the underlying services such as forwarding the energy trading requests to the different entities. The SDN architecture for energy trading in a secure manner is divided into three planes namely data, control, and application plane which are described in brief as follows.

1. Data Plane: The data plane is further divided into two sub-layers, i.e., inception layer and forwarding layer, which are elaborated as below.

- *Inception layer:* In this layer, all the entities which generate the energy trading requests such as-EVs and CSs are located as well as the roadside units (RSUs) which provide the communication back-end to the entities.

- *Forwarding layer:* In this layer, the devices such as-switches and routers which collects and forwards the data to different entities are placed. These devices gather the requests from the EVs or CSs through the RSUs and send these requests to the intended recipients. Moreover, the forwarding layer also floats back the results after the request processing to the associated entities through the RSUs. The communication protocols that are used in this layer belong to short/medium range (which sends requests from EVs or CSs to the RSUs and vice-versa) and medium/long range (to transfer the requests from RSUs to the switches/routers and vice-versa).

2. Control Plane: The control plane is the actual heart of the system architecture. All the control decisions and network policies are implemented at this plane. The SDN control logic analyzes the network topology and monitors the status of the forwarding devices to take the decision of the optimal routing path from the source node to the destination to reduce the overall delay. The control logic of the SDN controller is installed on the network operating system (NOS) at this plane and executed through a hypervisor which is present to perform network slicing using network virtualization. This virtualization helps to create an instance of the virtual SDN controllers from the physical controllers and also helps to accommodate virtual machine (VM) placement at every virtual SDN controller location. These locations cater to all the user requests for a particular region with the help of edge-as-a-service (which is closer to the data plane) which comprises virtual instances of OFF-switches and controller. With the help of these virtual switches and controllers, the user requests are served locally via edge data centers while the global cloud controller handles all the virtual instances of these switches and controller via cloud data center. The data plane communicates with the control plane through southbound application programming interfaces (APIs); whereas, the control plane communicates with the application plane via northbound APIs.

3. Application plane: The application plane can be considered as a front end where the users can exchange their information amongst each other or seek services from a service provider. In the present work, the application plane is tasked with providing the energy trading services to the EVs in a secure manner. This plane also provides a platform to the entities for dissipating their information which is required to make an agreement for the energy trading.

The brief working of the proposed system model is described as follows. The EVs and CSs in the data plane layer share their energy trading requirements with one another using the medium/long range communication technologies. These requirements are then passed to the forwarding layer of SDN architecture as shown in Fig. 2, which using the control plane, are passed onto the regional edge layers. To forward these requests and to trade the energy in a secure manner, blockchain service is used at the application plane. The detailed working of the complete scheme is discussed in the subsequent section.

4. Edge-as-a-service for Secure Energy trading using Blockchain

The proposed scheme is classified into three parts; 1) In the first part, energy trading scheme between EVs and CSs is presented, 2) In the second part, a computational offloading scheme using edge-as-a-service is designed, and 3) Finally, a blockchain-based secure energy trading mechanism using edge-as-a-service is presented. All these parts are elaborated in the subsequent subsections.

4.1. Edge-as-a-service for Energy trading in V2G Environment

The EVs in a smart city have to charge energy as and when required from the CSs deployed at various locations. For this purpose, EVs have to travel some distance and pay some price in order to charge the required amount of energy. Therefore, energy price and distance to be traveled by an EV up till the selected CS are two major parameters apart from the amount of energy required in any energy trading scheme. Since there are multiple options (in terms of CSs) for EVs to charge the required amount of energy, so the above mentioned two parameters play a major role in the selection of an optimal CS. In this process, EVs need to trade for energy with the geo-located CSs so as to achieve maximum benefit in terms of energy and price. Similarly, CSs have to sell the energy available with them so as to maximize their profit. Fig. 3 shows the energy trade process between multiple EVs and CSs in edge-as-a-service environment.

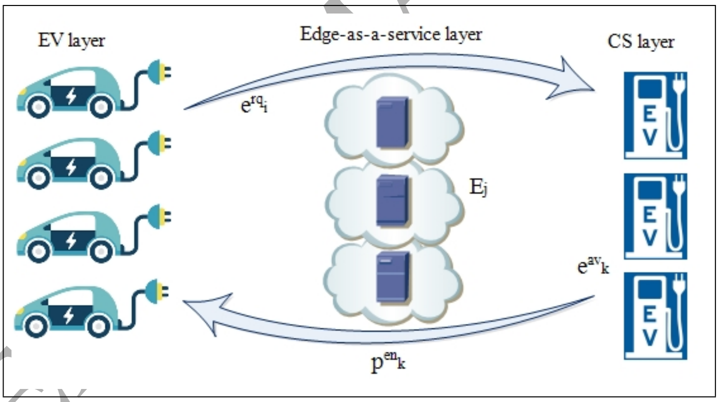


Figure 3: Energy trading mechanism.

The decision making for the entire process depicted in Fig. 3 is performed using some computational resources (cloud server, local server, etc). However, transmitting the data from EVs to remote cloud or server may end up in higher delay and additional cost for the energy trading service provider. To overcome this situation, edge computing is one of the popular technology which provides data processing and decision making platform closer to the location of the end user. This service oriented architecture of edge nodes reduces the additional delay and cost involved in the energy trading process and can be termed as edge-as-a-service. In the proposed energy trading mechanism, i EVs trade for energy with k CSs, wherein j edge nodes handles the entire computational load. In this process, apart from selecting an optimal CSs to charge the required amount of energy, the task of selecting optimal edge node for handling the computational load is also important, For this purpose, in this paper, an edge-as-a-service framework for energy trading between EVs and CSs is designed.

In the proposed scheme, EVs act as buyers and CSs act as sellers of energy. The major objective of EVs is to maximize their energy level, i.e, state of charge (SoC) or it can be said that EVs aims at reaching their rated battery capacity (e_i^{rt}). Hence, the energy required (e_i^{rq}) by i^{th} EV from k^{th} CS in order to reach e_i^{rt} is given as below.

$$e_i^{rq} = (SoC_i^{mx} - SoC_i^{pr})\, e_i^{rt} \tag{1}$$

where, SoC_i^{mx} and SoC_i^{pr} denotes maximum attainable SoC and present SoC levels of i^{th} EVs battery, respectively.

In order to supply e_i^{rq} to the i^{th} EV, all the CSs available at any instance must have sufficient amount of energy available with them. Now, the energy available (e_k^{av}) with k^{th} CS depends on the present SoC (SoC_k^{pr}), a threshold SoC (SoC_k^{th}) fixed by utility and the rated energy capacity (E_k^{rt}) of k^{th} CS. Using all these, e_k^{av} with k^{th} CS is given as below.

$$e_k^{av} = \left(SoC_k^{pr} - SoC_k^{th}\right) E_k^{rt} \tag{2}$$

s.t.

$$SoC_k^{pr} > SoC_k^{th} \tag{3}$$

Each EV has to pay some price for charging the required amount of energy from the CSs. Here, the price (P_k^{en}) charged by k^{th} CS from i^{th} EV for selling e_i^{rq} is given as below.

$$p_k^{en} = \alpha \left(\frac{SoC_k^{mx}}{SoC_k^{av} - SoC_k^{th}} \right) \tag{4}$$

s.t.

$$\alpha > 1 \tag{5}$$

$$P_i^s > P_i^b \tag{6}$$

where, α is a predefined scaling constant which is set by utility (whose value can vary depending on the utility constraints) and P_i^b and P_i^s are the price of buying and selling the energy respectively.

Each EV has to travel some distance to reach the selected CS for charging the required amount of energy. If the distance between an EV and CS is more, then more energy would be dissipated from EVs battery for traveling till the selected CS. Therefore, it is preferable for an EV to select a CS closer to its location. In the proposed scheme, it is assumed that the geographical area of smart city is divided into equal sized blocks. However, the proposed scheme is not limited to any specific scenario or equal sized blocks only and is applicable to any scenario. If in a case, the smart city is not divided into equal sized blocks, then the distance can be calculated using third-party services such as GPS. Let us assume an EVs (a) and a CS (b) located at a distance ($D_{a\to b}$) from each other. Now, D_a is the distance of EV_a from end of block and D_b is the distance of CS_b from end of the block. Here, D is considered as a fixed distance of the edges of each block. Now, the distance ($D_{a\to b}$) between two EVs (a and b) is calculated as mentioned below.

$$D_{a\to b} = \left|\frac{D_a}{D}\right| \times D + \left|\frac{D_b}{D}\right| \times D + N_{a\to b} \times D \tag{7}$$

where, $N_{a\to b}$ represents the number of blocks between a and b.

Using the above concept, i^{th} EV travels a distance ($D_{i\to k}$) to charge e_i^{rq} from k^{th} CS. The energy dissipated ($e^{ds}_{i\to k}$) to cover $D_{i\to k}$ is given as below.

$$e^{ds}_{i\to k} = SoC_i^{mx}\left(\frac{D_{i\to k}}{D_{mx}}\right) e_i^{rt} \tag{8}$$

where, D_{mx} denotes the maximum distance of i^{th} EV from all available CSs.

Now, once the i^{th} EV travels $D_{i\to k}$ after paying p_k^{en} to charge e_i^{rq} from k^{th} CS, the updated energy level of i^{th} EVs battery is given as below.

$$e_i^{up} = (SoC_i^{pr} + SoC_i^{rq})\, e_i^{rt} - e^{ds}_{i\to k} \tag{9}$$

Similarly, the updated energy level of k^{th} CS is given as below.

$$e_k^{up} = \left(SoC_k^{pr} e_k^{rt}\right) + \left(SoC_i^{rq} e_i^{rt}\right) \tag{10}$$

The above described process of energy trading depends on the utility functions of EVs and CSs. Utility function depicts the profit function which depends on energy, price and revenue with an entity. Two types of utility functions are computed in the proposed framework; 1) Utility functions related to edge node selection, and 2) Utility functions related to energy trading process. Both types of utility functions are discussed as below.

4.1.1. Utility functions related to edge node selection

The proposed scheme utilizes edge-as-a-service for offloading entire computational workload on edge nodes. Therefore, it becomes a very important task to select an optimal edge node among the available array of edge nodes. To achieve this task, a utility function is defined for an edge node also which depends on the delay and anticipated throughput. So, the utility function ($U_i(j)$) of i^{th} EV with respect to j^{th} edge node is given as below.

$$U_i(j) = \frac{B_j^{rq} \times T_j^{an}}{(n+1) \times t_j^{an}} \times \frac{1}{D_{\frac{i \to k}{j}}} \tag{11}$$

where, B_j^{rq} is the bandwidth required to transmit the computational workload to j^{th} edge node, T_j^{an} is the average anticipated throughput, t_j^{an} represents the average anticipated delay, and $D_{\frac{i \to k}{j}}$ depicts the distance traveled by i^{th} EV to charge e_i^{rq} from k^{th} CS wherein the entire computational processing takes place at j^{th} edge node.

Similarly, the edge nodes have also to compute their utility function with respect to all requesting EVs. Therefore, the utility function ($U_j(i)$) of j^{th} edge node with respect to i^{th} EV is given as below.

$$U_j(i) = R_j - C_{j \to i} \tag{12}$$

where, R_j is the revenue available with j^{th} edge node and $C_{j \to i}$ is the cost involved to serve the i^{th} EV.

Now, $C_{j \to i}$ involved to serve the i^{th} EV is given as below.

$$C_{j \to i} = \frac{B_{j \to i}^{rq} \times T_{j \to i}^{an}}{t_{j \to i}^{an}} - C_{j \to i}^{of} \tag{13}$$

where, $C_{j \to i}^{of}$ is the cost involved to the offloading of computational load from i^{th} EV to j^{th} edge node.

4.1.2. Utility functions related to energy trading process

The utility function (U_i) of i^{th} EV participating in the energy trading mechanism is given as below.

$$U_i = x_i \sum ln \left(y_i + (e_i^{rq} p_k^{en} - e_{i\to k}^{ds} p_k^{en})\right) \tag{14}$$

where, x_i, y_i are constants and the ln is preference modeling function [31].

Similarly, the utility function (U_k) of k^{th} CS participating in the energy trading mechanism is given as below.

$$U_k = \sum_{k=1}^{K} \sum_{i=1}^{I} p_k^{en} e_i^{rq} \tag{15}$$

Now, the major objective of the proposed energy trading scheme using edge-as-a-service is to maximize the combined utility ($U_{\frac{i\to k}{j}}$).

$$U_{\frac{i\to k}{j}} = \sum_{i=1}^{I} \begin{bmatrix} 1,1,1 & 1,2,1 & . & . & 1,j,1 \\ 1,1,2 & 1,2,2 & . & . & 1,j,2 \\ . & . & . & . & . \\ . & . & . & . & . \\ i,1,k & i,2,k & . & . & i,j,k \end{bmatrix}. \tag{16}$$

Transforming Eq. 16 into simpler form, the objective function of the proposed scheme is shown as below.

$$max \left[\sum_{j=1}^{J} (U_{\frac{1\to 1}{j_1}})\chi_{\frac{1\to 1}{j_1}} + (U_{\frac{1\to 2}{j_2}})\chi_{\frac{1\to 2}{j_2}} + ... + (U_{\frac{i\to k}{j_n}})\chi_{\frac{i\to k}{j_n}} \right] \tag{17}$$

s.t.

$$0 < e_i^{rq} \leq e_i^{rt} \tag{18}$$
$$p_k^{en} > 0 \;\; \forall k \in \text{Ķ} \tag{19}$$
$$0 < e_k^{av} \leq e_k^{rt} \tag{20}$$
$$e_i^{rq} \leq e_k^{av} \tag{21}$$
$$\chi_{\frac{i\to k}{j_n}} \in [0,1] \tag{22}$$
$$U_i(k) > U_i(k^*) \tag{23}$$
$$U_k(t) > U_k(t-1) \tag{24}$$
$$D_{(\frac{i\to k}{j})} < D_{(\frac{i\to k}{j})^*} \tag{25}$$

An optimal $\frac{i\to k}{j_n}$ pair is selected on the basis of a decision variable ($\chi_{\frac{i\to k}{j_n}}$, $\forall t$), which is defined as below.

$$\chi_{\frac{i\to k}{j_n}} = \begin{cases} 1 & \text{for} \quad U_{\frac{i\to k}{j}} > U^{*}_{\frac{i\to k}{j}} \\ 0 & \text{for} \quad otherwise \end{cases} \tag{26}$$

where, $\frac{i\to k}{j}^{*}$ depicts the set of all pairs other than $\frac{i\to k}{j}$.

To achieve the above objectives, Algorithm I is designed wherein the first purpose of i EVs is to select an edge node for processing of their energy trading requests in the first phase. For this purpose, they compute $U_i(j)$ with respect to j edge nodes. On the basis of the utilities, EVs sort the list of edge node on the basis of their preference. On the other hand, edge nodes also compute their utility ($U_j(k)$) with respect to i EVs. After computing utilities, edge nodes also arrange then in the order of their preference. Finally, both EVs and edge nodes converge to an optimal solution.

Once an edge node is selected, in the second phase, the EVs compute e_i^{pr} which is compared to e_i^{rt}. If ($e_i^{pr} < e_i^{rt}$), then e_i^{rq} is computed and announced to k CSs. In a similar manner, k CSs check their SoC_k^{pr}. Using SoC_k^{pr}, the e_k^{av} is computed. If ($e_i^{av} < e_i^{th}$), then p_k^{en} is calculated (line 10-22). After this, the CSs compute their utility on the basis of e_i^{rt} and p_k^{en} and compare it with the previous instance. If ($U_k(t+1) > U_k(t)$), then the computed p_k^{en} is announce to i EVs. Now, i EVs compute their utility on the basis of the announced p_k^{en} from all k CSs and order then on the basis of preference. On the basis of this preference ordering, a CS is selected for each requesting EV. Hence, an optimal $\frac{i\to k}{j}$ pair confirmed for which $\chi_{\frac{i\to k}{j}}$ is set to 1 (line 23-34). Finally, e_i^{up} and e_k^{up} are computed for next round if the concerned EV charge the required amount of energy from the selected CS.

4.1.3. An example of the proposed framework

Let us consider an example of an EV which requires to charge energy from CS. Fig. 4 shows the layout of a smart city scenario which is divided into equal sized blocks. In this exemplar setup, 4 CSs (A, B, C and D) are located. In the same way, 4 edge nodes (1, 2, 3 and) are geo-located at various positions. The concerned i^{th} EV has to travel a distance of $D_{i\to A}$, $D_{i\to B}$, $D_{i\to C}$ and $D_{i\to D}$ from all four CSs.

Now, the EV has to choose an optimal edge node among the available which would handle its computational load. The decision is based on the

Algorithm 1 Edge-as-a-service for energy trading

```
Input: $E_i^{rt}$,$SoC_i^{prs}$, $E_j^{rt}$, $SoC_j^{prs}$
Output: $j$, $U_i$, $U_j$
for (i=1; i≤n; i++) do
    Compute $R_i^{rq}(k)$
    if ($R_i^{rq}(k) < R_i^{avl}(k)$) then
        Compute $U_i(j) = \frac{B_j^{rq} \times T_j^{an}}{(n+1) \times t_j^{an}} \times \frac{1}{D_{\frac{i \to k}{j}}}$
        Arrange $lnU_i(j)$
    end if
    Compute $U_j(i) = R_j - \left( \frac{B_{j \to i}^{rq} \times T_{j \to i}^{an}}{t_{j \to i}^{an}} - C_{j \to i}^{of} \right)$
    Arrange $lnU_i(j)$
    Converge to an optimal solution.
    Check $e_i^{pr} = SoC_i^{pr} e_i^{rt}$
    if ($e_i^{pr} < e_i^{rt}$) then
        Compute $e_i^{rq} = (SoC_i^{mx} - SoC_i^{pr})\, e_i^{rt}$
        Announce $e_i^{rq}$ to $k$ available CSs.
    end if
    for (k=1; k≤n; k++) do
        Check $SoC_k^{pr}$
        Compute $e_k^{av} = \left( SoC_k^{pr} - SoC_k^{th} \right) e_k^{rt}$
        if ($e_i^{av} < e_i^{th}$) then
            Compute $p_k^{en} = \alpha \left( \frac{SoC_k^{mx}}{SoC_k^{av} - SoC_k^{th}} \right)$
        else
            Opt out of energy trading scheme.
        end if
        Compute $U_k = p_k^{en} e_i^{rq}$
        if ($U_k(t+1) > U_k(t)$) then
            Announce $p_k^{en}$
        end if
        Compute $U_i(k) = x_i \sum ln \left( y_i + (e_i^{rq} p_k^{en} - e_{i \to k}^{ds} p_k^{en}) \right)$
        if ($U_i(k) > U_i(k^*)$) then
            Announce $p_k^{en}$                                  ▷ $k^*$ is set containing all except k
            Select $k$
        end if
    end for
    $\frac{i \to k}{j}$ pair confirmed.
    Set $\chi_{\frac{i \to k}{j}} = 1$ for $\frac{i \to k}{j}$ pair
    Compute $e_i^{up} = (SoC_i^{pr} + SoC_i^{rq})\, e_i^{rt} - e_{i \to k}^{ds}$
    Compute $e_k^{up} = (SoC_k^{pr} e_k^{rt}) + (SoC_i^{rq} e_i^{rt})$
    Start the next round.
end for
```

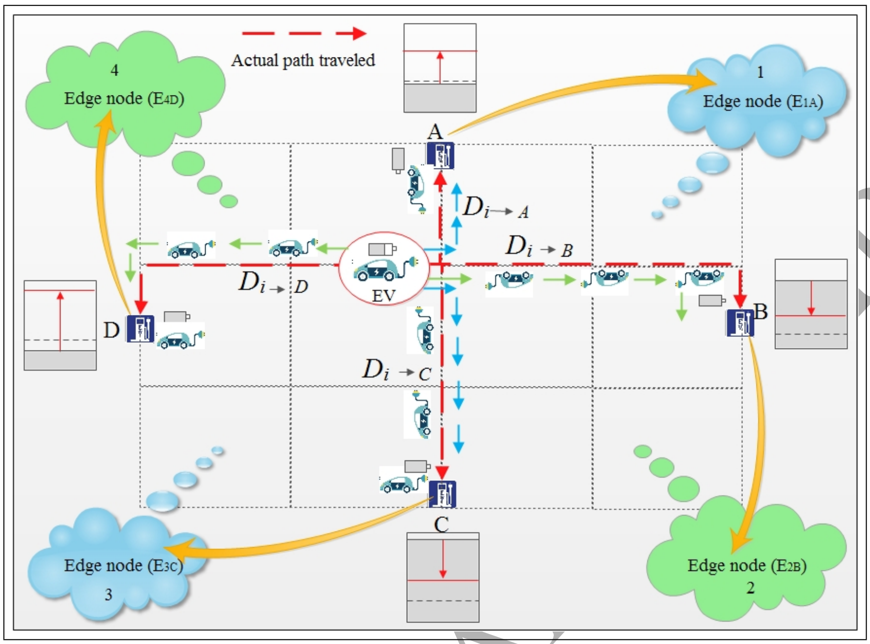


Figure 4: Edge-as-a-service in energy trading.

utilities of the both entities wherein distance is one of the important parameters. In the considered example, the utility of i^{th} EV with respect to each edge node is shown in Table 2. The utility of i^{th} EV is maximum for edge node 1 and therefore it is selected to handle its computational tasks.

Table 2: Values of utility (EV versus edge nodes)

EV/edge node	**E1**	**E2**	**E3**	**E4**
EV_i	1.834	1.678	1.456	1.105

Now, the utility of i^{th} EV is calculated on the basis of various parameters of which distance is the most important. Table 3 shows the values of various parameters considered and the utility computed. According to Table 3, the i^{th} EV achieves highest utility value with respect to D. In the similar way, the CSs also compute the value of their utilities if multiple EVs are requesting them (this case is not discussed in the example).

Table 3: Values of various parameters and computed utility (EV versus CSs)

CS	$D_{i\to k}$ **(in mtrs)**	e_i^{rq} **(in kWh)**	$e_{i\to k}^{ds}$ **(in kWh)**	p_k^{en} **(in Rs)**	$U_i(k)$
A	5500	8	7.9	5	0.5
B	1500	8	4.4	5	22
C	2800	8	7.1	5	4.5
D	1100	8	3.2	5	24

4.2. Blockchain for secure energy trading using edge-as-a-service

The blockchain scheme is used to provide the security for energy trading between the EVs and CSs. The consensus algorithm is used to validate the transaction after which it is added to the blockchain and its log is shared among the approver nodes. All the nodes that are selected as edge nodes serve as approver node expect the one which is used to compute the operations for an EV. This implies that the edge node which perform computations for a particular EV would not participate as its approver node to validate that transaction. The complete process of validating the transaction using approver nodes with the help of blockchain consensus algorithm is shown in Fig. 5 and the same is explained as follows.

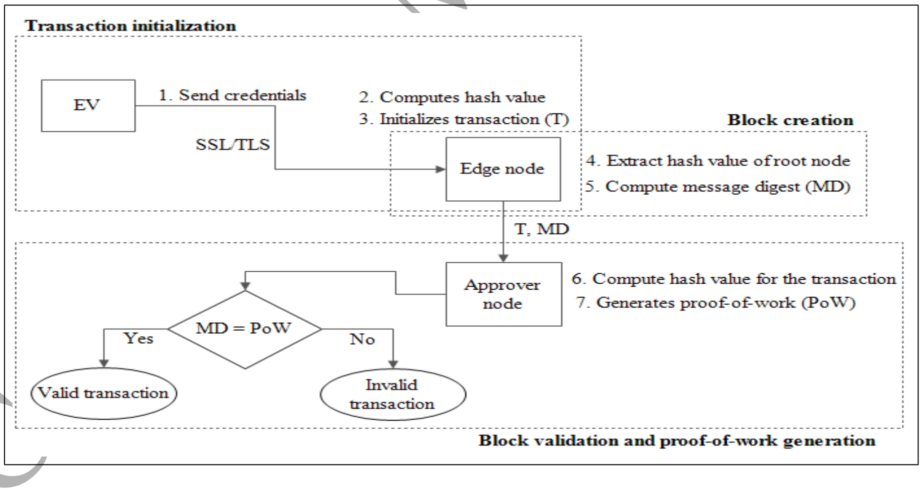


Figure 5: Blockchain process for energy trading.

(a.) Transaction initialization:
Initially, when the i^{th} EV which wants to trade the energy with CS, it sends its credentials (i.e., identity, price, energy) to the trusted edge node present

in its present cluster. This edge node then computes a hash function (H_i) and initializes the transaction (T_i) which is floated in the network. The calculation of these values is given as follows.

$$H_i = [ID_i \mid\mid E_{id} \mid\mid nonce] \tag{27}$$

$$T_i = Hash(H_i \mid\mid B_i) \tag{28}$$

where, ID_i is the identity of i^{th} EV, E_{id} is the edge nodes identity where the computations are performed, *nonce* is nonce value, and B_i is the data block (which contains information regarding excess/required energy, price, amount of energy).

(b.) Block header generation:

Once the transaction for i^{th} EV is finalized, the block headers are computed in the following way. The block header is generated by computing the hash of root in the Merkle hash tree. The basic premise behind using the Merle hash tree is that the hash value of the root node (H_R) in this tree can be easily computed by merging the hash values of the left (H_l) and right child (H_r) of the root node.

$$H_R = Hash(H_l \mid\mid H_r) \tag{29}$$

In general, as the tree would have the index of all the transactions in the network, the value of root hash can be computed as,

$$H_R = Hash\{[H_1 + H_2] + [H_3 + H_4] + \ldots + [H_{(n-1)} + H_n]\} \tag{30}$$

A message digest is now computed using SHA-256 as,

$$MD_i = SHA - 256\ (H_{pre} \mid\mid H_R \mid\mid T_i \mid\mid ts \mid\mid t_v \mid\mid padding) \tag{31}$$

where, H_{pre} is the previous hash value in the transaction, ts is the time-stamp, t_v is the time for which the transaction will remain valid, and *padding* is the padding bits added to make the fixed-sized message digest. The value of H_{pre} is extracted from the last hash value in the present blockchain. The message digest is sent to the approver nodes for validation purposes. The value of t_v depends on the time that the EV would remain connected to the edge node (it can be either specified by the EV owner or can be computed from the mobility pattern of the EV). The edge node sends the values of MD_i, ts, t_v, and T_i to the approver nodes for validation purposes.

(c.) Block validation and proof-of-work (PoW) generation:

Once the approver nodes receive the message digest from the edge nodes, they compute the PoW for each transaction which is to be added in the blockchain. It is to be noted that the edge node which performs computations for a particular EV would not take part in the validation process as it acts as an intermediate agent for the EV to process the transaction. All the other edge nodes present in the network would act as approver nodes for validating the transaction for that EV. For this purpose, all these approver nodes compute the PoW in the following way and if more than 50% reach to a general consensus (either valid/invalid transaction), then the transaction is added to the blockchain depending on the outcome of the consensus. The approver nodes have an access to the log of all the valid transactions in the blockchain at the present instance of time. The combined list of all such transaction can be represented as,

$$T_L = [T_1 \mid\mid T_2 \mid\mid T_3,\ \ldots,\ \mid\mid T_n] \tag{32}$$

The hash value of the root node in Merkle hash tree is computed as depicted in eq. (30). The value of H_{pre} is extracted from the previous block in the blockchain. Using these values an intermediate hash value is generated as,

$$H_{new} = [H_{pre} \mid\mid H_R] \tag{33}$$

This hash value is then appended with the values of time-stamp, valid time of transaction, and the transaction itself which are received from the edge node. However, before that, the valid transaction time is checked. If the valid time of the transaction is less then the present time, then the transaction is discarded and the EV would make another request after joining next edge node. Otherwise, the final hash value for i^{th} EV is computed as,

$$H_{final} = [H_{new} \mid\mid T_i \mid\mid ts \mid\mid t_v \mid\mid padding] \tag{34}$$

The approver node generates a PoW for i^{th} EV for this hash value using the following equation.

$$PoW_i = SHA - 256\ (H_{final}) \tag{35}$$

If the value of this PoW is matched with the received message digest, then the block is considered validated by the approver node. Now, all the approver nodes compute PoW for i^{th} EV and send their result to the transaction server.

If more than 50% of the nodes agree on a general consensus that the block is valid, then this block is added in the blockchain, otherwise, associated transaction is discarded. After the block is added in the blockchain, all the approver nodes as well as the edge node are informed about it and their ledger is updated to add the information of this newly added block.

5. Results and Discussions

The case study of a city of 5 km × 5 km is considered which is considered for evaluating the proposed scheme which comprises of 4 CSs placed at different geographical locations and 40 EVs which can charge or discharge at any of the CSs based on their requirements. The EVs of different types are considered for energy trading with their battery specifications given in Table 4 [32]. The CSs are considered to be available for 24 hours with ample charging points available for the EVs. The value of SoC_k^{th} is fixed at 50%.

Table 4: Type of EVs considered.

EV type	**Battery capacity (kWh)**	**Charging voltage (V)**	**Charging current (A)**
Type 1	35	240	32
Type 2	16	240	15
Type 3	24	240	40
Type 4	56	240	70
Type 5	24	115	12
Type 6	8	120	15
Type 7	4.4	120	15

5.1. Performance of energy trading scheme

Considering all these parameters, the energy trading in the smart transportation sector takes place in a secure manner. The primary factor for energy trading is the energy level in the EVs' batteries and the charging requirements of the CSs. The EVs which want to sell their extra energy will go to the CS which presently face energy deficit and the EV would get the charging from the CS which has excess energy with them. The energy level in the EVs' battery is dependent on the present SoC level in the batteries which is shown in Fig. 6.

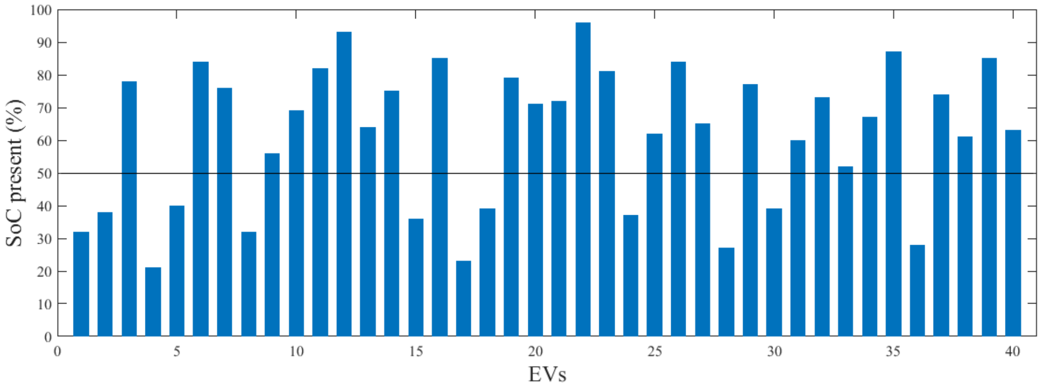


Figure 6: Present SoC level of the EVs' batteries.

Depending on this SoC level, the excess and required energy in the EVs' batteries is computed on the basis of the threshold SoC level which should be present in the batteries. This excess and required SoC level for all the EVs is depicted in Fig. 7.

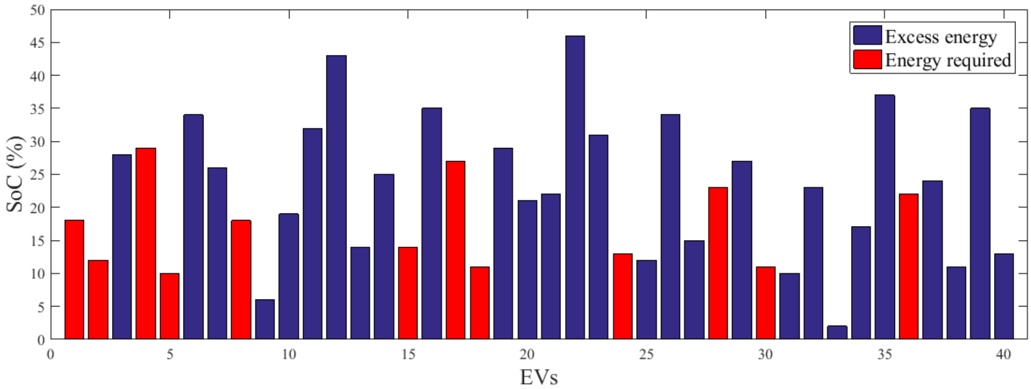


Figure 7: Excess and required SoC level of the EVs' batteries.

On the basis of excess or deficit SOC levels, the EV moves to the CS either for charging or discharging purpose which helps to stabilize the load profile of the CS. The change in its load profile before and after the energy trading is highlighted in Fig. 8. It can be inferred from this figure that the proposed scheme is effective to balance the load to a great extent on the CS by trading energy with the EVs. It is to be noted that the balanced load profile in Fig. 8 for CS is assumed to be fixed to depict a typical case of CS facing peak and valley in the load demand at different time intervals. During the peak shaving, the EVs which have excess energy travels to CS while for valley filling, the EVs which require energy would help to cater the load imbalance at the CSs.

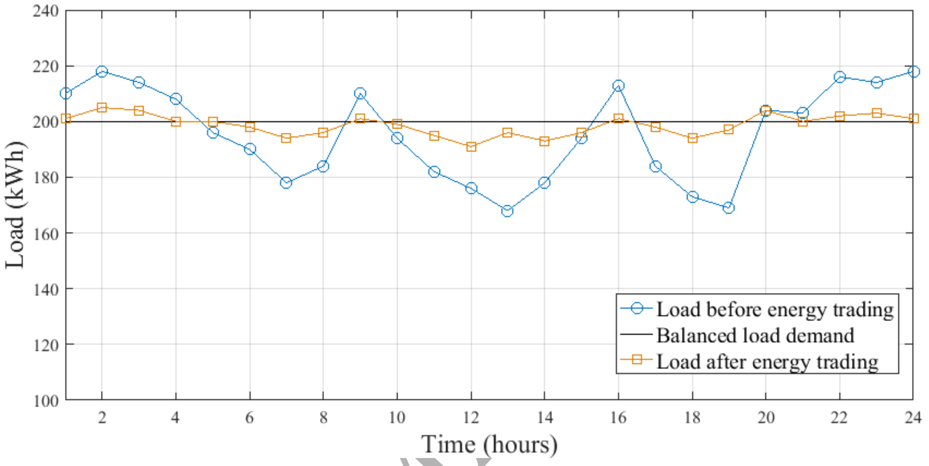


Figure 8: Load profile of the CS before and after energy trading.

The price announced by the EVs and CSs is a major factor to calculate their utilities based on which the energy trading decisions are taken. When the CS requires energy, it would follow the price announced by the EV and when the EV needs energy, it would adhere to the price announced by the CS. The price announced by one EV and one CS is illustrated in Figs. 9 and 10. It can be seen in Fig. 9 that this EV announced fixed price from 10 to 17 hours; it is because this duration falls within the office hours of the EV owner and the EV can easily be parked at the CS during this duration. However, the same is not true in the case of CS which announces the price on the basis of the energy requirements of the associated users, therefore, more fluctuations can be seen in Fig. 10. Whenever the gap between demand and supply is more, the CS would choose to buy/sell the energy at a higher cost as compared to the case when the gap is less.

5.1.1. Impact of edge-as-a-service platform and SDN

It is very important to analyze the impact of using edge-as-a-service platform in comparison to remote control center. It is pertinent to mention here

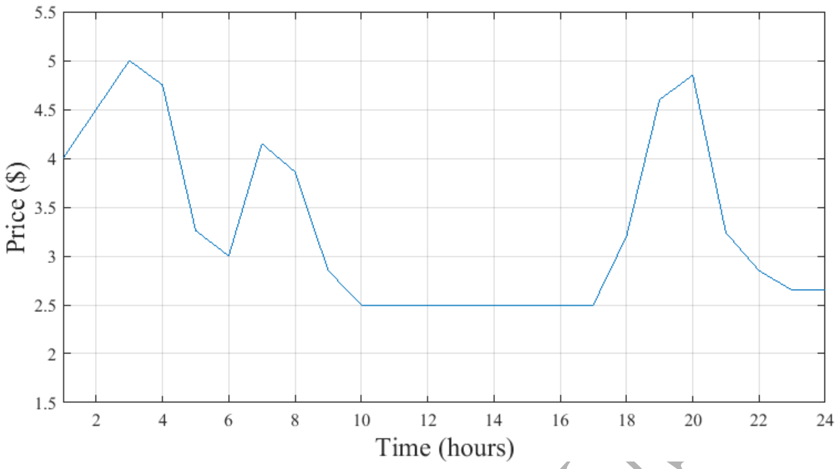


Figure 9: Price announced by the EV to discharge its battery.

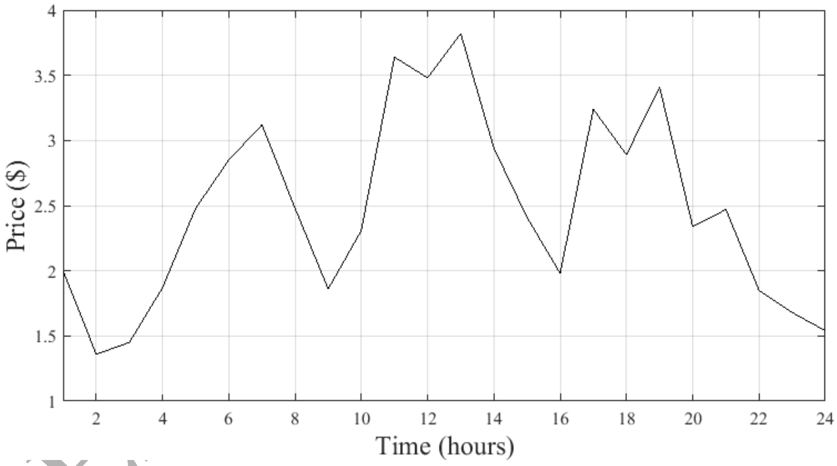


Figure 10: Price announced by the CS for charging EVs' batteries.

that serving requests of EVs closer to their location has strong impact on delay and throughput. The impact of this factor is clearly visible in results obtained in Figs. 11 and 12. In Fig. 11, the delay incurred for processing the EVs request using edge-as-a-service platform is far less as compared to remote control center scenario. Here, the remote control center refers to a centralized scenario where decision making is performed by a remote controller rather than an distributed edge-as-a-service platform. Moreover, the role of SDN is also accessed on the delay incurred. The result clearly shows that using dynamic network control through SDN further reduces delay incurred to handle EVs request. Moreover, the impact of proposed scheme is visible on throughput achieved also. Fig. 12 shows that when edge-as-a-service platform is used along with SDN, then throughput achieved is higher in contrast to other two cases.

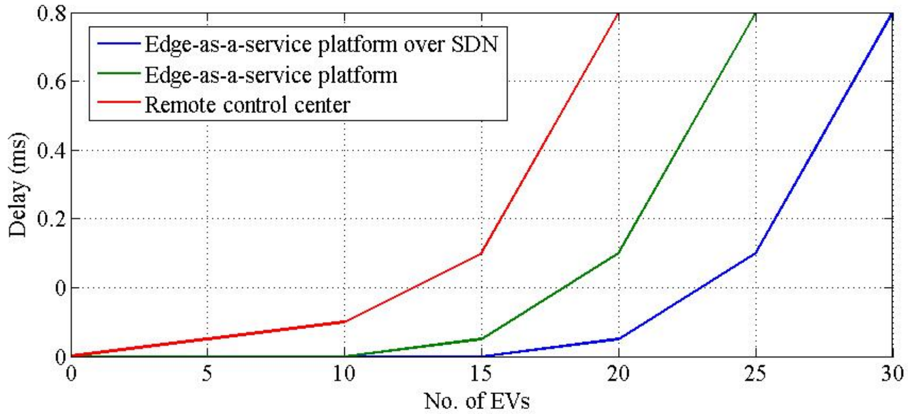


Figure 11: Delay.

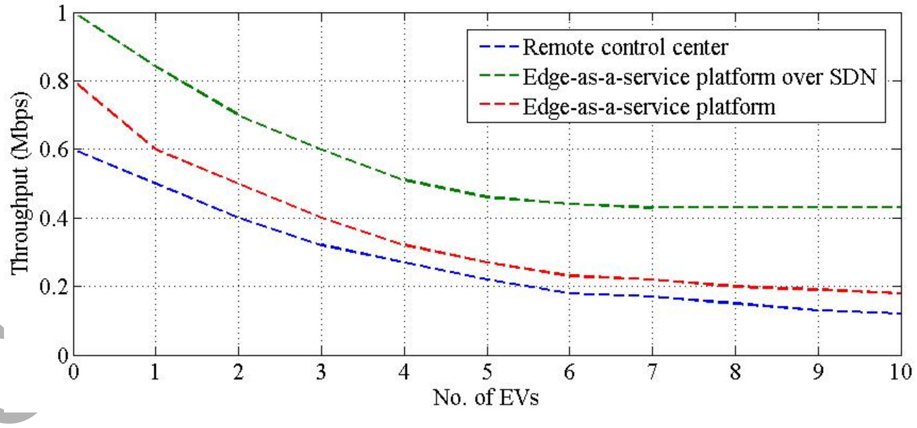


Figure 12: Throughput.

5.1.2. Performace of blockchain in energy trading

The proposed scheme is also evaluated in terms of providing robust security while energy trading on the basis of its communication and computation costs. To calculate these costs, the size of the various fields used in the blockchain should be known which is given in the Table 5.

Table 5: Information about size of various fields.

Field	***Size***	***Field***	***Size***
ID_i	32 bits	t_v	16 bits
E_{id}	32 bits	H_R	160 bits
nonce	16 bits	MD_i	256 bits
T_i	160 bits	PoW_i	256 bits
ts	16 bits	-	-

Communication Cost: The communication cost is computed in terms of bits transferred while initializing the transaction. Initially, the hash value H_i is computed at the edge node which takes 80 bits (32 bits of identity, 32 bits of edge identity, and 16 bit nonce value). After this, the transaction is initialized which uses the *hash* function which gives the output of 160 bits. Now, to generate the block header, the hash values from Merkle hash tree are computed which are of 160 bits and the final message digest for the EV is computed using SHA-256 which outputs the digest of 256 bits. Similar to this message digest, the PoW for each EV is generated at each approver nods which outputs a 256 bits value. As each EV communicates its message digest, time-stamp, valid time, and transaction, so, the overall communication cost for an EV in terms of message bits comes out to be [256+16+16+160]= 448 bits. For each approver node, it computes the PoW which takes 256 bits and sends another bit for computing the validity of the transaction, thereby, taking [256+1] = 257 bits.

Computation Cost: The computation operations required in the blockchain are addition, hashing, and append operations. If the total time taken for these operations is 1 ms, 3.8 ms and 0.35 ms respectively, then, the computation cost in terms of calculation time can be calculated as follows. For an EV, there are nine append operations, (n-1) add operations (with 'n' attributes in Merkle hash tree), and four hash operations. So, for a Merkle hash tree with 100 nodes, the computation cost for an EV comes out to be $[0.35\times 9 + 99\times 1 + 4\times 3.8] = 117.35$ ms. Similar to this, each approver node uses five append operations and one hash operation. So, the computation cost of each approver node is $[0.35 \times 5\ +\ 3.8 \times 1]= 2.75$ ms.

Analyzing the results into further detail, the impact of number of transaction on header generation, validation and block preparation time is also accessed. Fig. 13 shows the variation of header generation and validation time with respect to an increase in the number of transactions. It is clearly shown that the time required to generate header value and validate the transaction increase with an increase in the number of transactions. Similarly, as shown in Fig. 14, the block preparation time also increases with an increase in the number of transactions.

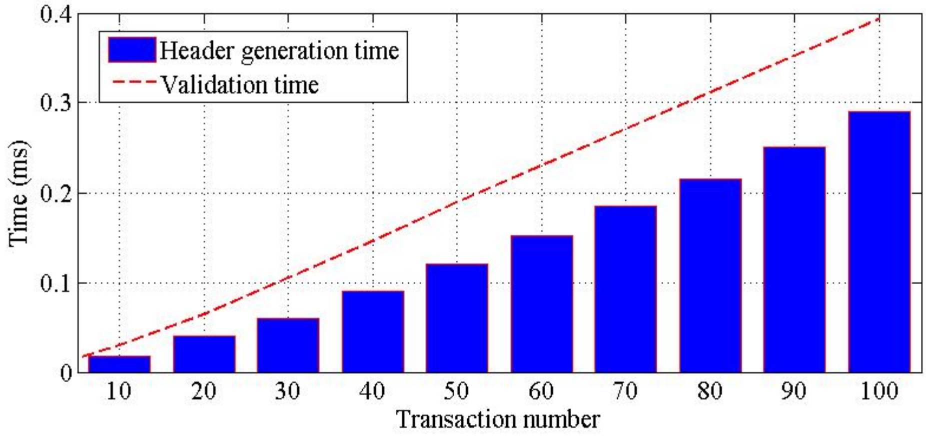


Figure 13: Computation time.

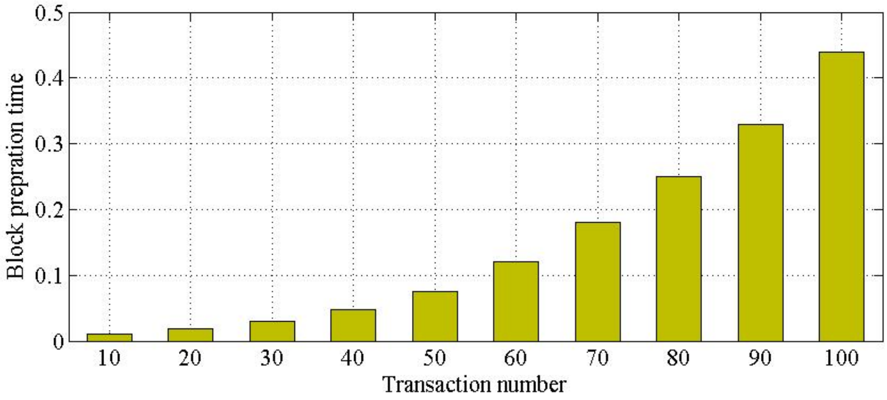


Figure 14: Key transfer time.

6. Conclusion

This paper presents SURVIVOR: a blockchain based edge-as-a-service framework for secure energy trading in SDN-enabled V2G environment. Edge nodes are used to serve the EV request closer to their physical location. The SDN-enabled communication architecture is used to provide communication backbone to the complete smart transportation sector. This increases the overall throughput of the network by decreasing the network latency for passing the information among various nodes. Apart from it, the computation of energy trading transactions for EVs are performed at the edge nodes which are secured using the blockchain consensus algorithm. The edge nodes are selected on the basis of utility function. The energy trading takes place such that the utility for the EVs as well as the CSs increases upon the successful energy trade. The results obtained prove that the proposed scheme effectively manages the energy trading requests by reducing the delay for communication. Moreover, the security of transactions is ensured using the blockchain mechanism by generating the proof-of-work. The results depict that it is lightweight in terms of computation and communication costs and does not put excess burden on the network. Moreover, the block preparation, header generation and validation times were also computed for SURVIVOR.

In future, the consortium blockchain mechanism can be used to analyze the performance of the proposed work. Moreover, the issues of content caching and vehicle mobility could be investigated to improve effectiveness of the proposed scheme.

References

[1] F. Cunha, L. Villas, A. Boukerche, G. Maia, A. Viana, R. A. Mini, A. A. Loureiro, Data communication in vanets: Protocols, applications and challenges, Ad Hoc Networks 44 (2016) 90–103.

[2] G. S. Aujla, A. Jindal, N. Kumar, EVaaS: Electric vehicle-as-a-service for energy trading in sdn-enabled smart transportation system, Computer Networks 143 (2018) 247 – 262.

[3] Emissions and charging costs-electric-cars, [Accessed: May 2018].
URL `https://www.ucsusa.org/clean-vehicles/electric-vehicles/emissions-and-charging-costs-electric-cars`

[4] Global Energy Outlook 2018, [Accessed: Aug. 2018]. [link].
URL `https://www.connaissancedesenergies.org/sites/default/files/pdf-actualites/globalevoutlook2018.pdf`

[5] R. Yu, W. Zhong, S. Xie, C. Yuen, S. Gjessing, Y. Zhang, Balancing power demand through ev mobility in vehicle-to-grid mobile energy networks, IEEE Transactions on Industrial Informatics 12 (1) (2016) 79–90.

[6] C. Lin, D. He, X. Huang, K.-K. R. Choo, A. V. Vasilakos, Bsein: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0, Journal of Network and Computer Applications 116 (2018) 42–52.

[7] Q. Feng, D. He, S. Zeadally, M. K. Khan, N. Kumar, A survey on privacy protection in blockchain system, Journal of Network and Computer Applications.

[8] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system.
URL `https://bitcoin.org/bitcoin.pdf`

[9] X. Xu, I. Weber, M. Staples, L. Zhu, J. Bosch, L. Bass, C. Pautasso, P. Rimba, A taxonomy of blockchain-based systems for architecture design, in: IEEE International Conference on Software Architecture (ICSA), 2017, pp. 243–252.

[10] G. S. Aujla, A. Jindal, N. Kumar, M. Singh, Sdn-based data center energy management system using res and electric vehicles, in: IEEE Global Communications Conference (GLOBECOM), 2016, pp. 1–6.

[11] D. Li, L. Deng, Z. Cai, B. Franks, X. Yao, Intelligent transportation system in macao based on deep self-coding learning, IEEE Transactions on Industrial Informatics 14 (7) (2018) 3253–3260.

[12] Y. Zhang, G. Zhang, R. Fierro, Y. Yang, Force-driven traffic simulation for a future connected autonomous vehicle-enabled smart transportation system, IEEE Transactions on Intelligent Transportation Systems 19 (7) (2018) 2221–2233.

[13] I. S. Bayram, M. Z. Shakir, M. Abdallah, K. Qaraqe, A survey on energy trading in smart grid, in: IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2014, pp. 258–262.

[14] B. Zhang, C. Jiang, J. Yu, Z. Han, A contract game for direct energy trading in smart grid, IEEE Transactions on Smart Grid 9 (4) (2018) 2873–2884.

[15] W. Shuai, P. Maillé, A. Pelov, Charging electric vehicles in the smart city: A survey of economy-driven approaches, IEEE Transactions on Intelligent Transportation Systems 17 (8) (2016) 2089–2106.

[16] C. Zhang, J. Wu, C. Long, M. Cheng, Review of existing peer-to-peer energy trading projects, Energy Procedia 105 (2017) 2563–2568.

[17] G. S. Aujla, N. Kumar, A. Y. Zomaya, R. Ranjan, Optimal decision making for big data processing at edge-cloud environment: An sdn perspective, IEEE Transactions on Industrial Informatics 14 (2) (2018) 778–789.

[18] A. Carvallo, J. Cooper, The advanced smart grid: Edge power driving sustainability, Artech House, 2015.

[19] G. S. Aujla, N. Kumar, Mensus: An efficient scheme for energy management with sustainability of cloud data centers in edgecloud environment, Future Generation Computer Systems 86 (2018) 1279 – 1300.

[20] N. Kumar, S. Zeadally, J. J. Rodrigues, Vehicular delay-tolerant networks for smart grid data management using mobile edge computing, IEEE Communications Magazine 54 (10) (2016) 60–66.

[21] W. Bao, D. Yuan, Z. Yang, S. Wang, W. Li, B. B. Zhou, A. Y. Zomaya, Follow me fog: Toward seamless handover timing schemes in a fog computing environment, IEEE Communications Magazine 55 (11) (2017) 72–78.

[22] Ö. Akgül, B. Canberk, Software defined things: a green network management for future smart city architectures, in: Smart cities and Homes, 2016, pp. 41–57.

[23] A. Rego, L. Garcia, S. Sendra, J. Lloret, Software defined network-based control system for an efficient traffic management for emergency situations in smart cities, Future Generation Computer Systems 88 (2018) 243 – 253.

[24] W. Zhong, R. Yu, S. Xie, Y. Zhang, D. H. Tsang, Software defined networking for flexible and green energy internet, IEEE Communications Magazine 54 (12) (2016) 68–75.

[25] Z. Lu, C. Sun, J. Cheng, Y. Li, Y. Li, X. Wen, Sdn-enabled communication network framework for energy internet, Journal of Computer Networks and Communications 2017 (2017) 13.

[26] K. Mannaro, A. Pinna, M. Marchesi, Crypto-trading: Blockchain-oriented energy market, in: AEIT International Annual Conference, 2017, pp. 1–5.

[27] S. Aggarwal, R. Chaudhary, G. S. Aujla, A. Jindal, A. Dua, N. Kumar, Energychain: Enabling energy trading for smart homes using blockchains in smart grid ecosystem, in: Proceedings of the 1st ACM MobiHoc Workshop on Networking and Cybersecurity for Smart Cities, 2018, p. 1.

[28] A. Dorri, F. Luo, S. S. Kanhere, R. Jurdak, Z. Y. Dong, Spb: A secure private blockchain-based solution for energy trading, arXiv preprint arXiv:1807.10897.

[29] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, E. Hossain, Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains, IEEE Transactions on Industrial Informatics 13 (6) (2017) 3154–3164.

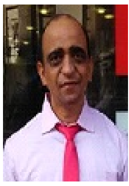
[30] C. Liu, K. K. Chai, E. T. Lau, Y. Chen, Blockchain based energy trading model for electric vehicle charging schemes, in: International Conference on Smart Grid Inspired Future Technologies, Springer, 2018, pp. 64–72.

[31] G. S. Aujla, S. Garg, S. Batra, N. Kumar, I. You, V. Sharma, Drops: A demand response optimization scheme in sdn-enabled smart energy ecosystem, Information Sciences 476 (2019) 453–473.

[32] K. Kaur, A. Dua, A. Jindal, N. Kumar, M. Singh, A. Vinel, A novel resource reservation scheme for mobile phevs in v2g environment using game theoretical approach, IEEE Transactions on Vehicular Technology 64 (12) (2015) 5653–5666.

Anish Jindal received his Bachelor of Technology degree from Punjab Technical University, India in 2012 and Master of Engineering degree from University Institute of Engineering and Technology, Panjab University, Chandigarh, India in 2014, both in Computer Science and Engineering. He received his Ph.D. degree in Computer Science and Engineering Department from Thapar University, Patiala (Punjab), India in 2018. He is working as a Senior Research Associate in the School of Computing and Communications, Lancaster University, UK. Prior to this, he was a Senior Research fellow of Council of Scientific and Industrial Research, India. He is member of the IEEE, ACM, and IAENG.

Gagangeet Singh Aujla is working as an Associate Professor in Computer Science and Engineering Department, Chandigarh University, Mohali, Punjab, India. He received the B.Tech degree and the M.Tech degree in Computer Science and Engineering from Punjab Technical University, Jalandhar, Punjab, India, in 2003 and 2013, respectively. He received his Ph.D. in Computer Science and Engineering from Thapar Institute of Engineering and Technology, Patiala, Punjab, India in 2018. He received IEEE TCSC Outstanding Ph.D. Dissertation Award at Guangzhou, China in 2018. He has many research contributions in the area of smart grid, cloud computing, software defined networks, and security. He is member of the IEEE, and ACM.



Neeraj Kumar received his Ph.D. in CSE from Shri Mata Vaishno Devi University, Katra (J & K), India, and was a postdoctoral research fellow in Coventry University, Coventry, UK. He is working as an Associate Professor in the Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala (Punjab), India. He has published more than 200 technical research papers in leading journals and conferences. He is an Associate Technical Editor of IEEE Communication Magazine and an Associate Editor of IJCS, Wiley, JNCA, Elsevier, and Security & Communication, Wiley. He is senior member of the IEEE.