Abstract: Transactive energy systems (TES) are modern mechanisms in electric power systems that allow disparate control agents to utilise distributed generation units to engage in energy transactions and provide ancillary services to the grid. Although voltage regulation is a crucial ancillary grid service within active distribution networks (ADNs), previous work has not adequately explored how this service can be offered in terms of its incentivisation, contract audibility, and enforcement. Blockchain technology shows promise in being a key enabler of TES, allowing agents to engage in trustless, persistent transactions that are both enforceable and auditable. To that end, this study proposes a blockchain-based TES that enables agents to receive incentives for providing voltage regulation services by (i) maintaining an auditable reputation rating for each agent that is increased proportionately with each mitigation of a voltage violation, (ii) utilising smart contracts to enforce the validity of each transaction and penalise reputation ratings in case of a mitigation failure, and (iii) automating the negotiation and bidding of agent services by implementing the contract net protocol as a smart contract. Experimental results on both simulated and real-world ADNs are executed to demonstrate the efficacy of the proposed system.

1 Introduction

Under the smart-grid paradigm, legacy power systems are evolving from a centralised, fossil-fuel based architecture into a decentralised, highly flexible system that is served primarily by distributed generation units (DGs). Particularly in active distribution networks (ADNs), DGs are being deployed in close proximity to the end consumer, significantly reducing line losses and increasing grid resiliency by allowing end consumers to draw backup power from the DGs when an outage occurs within the main grid. With traditional end consumers now beginning to own and operate DGs, the role of the end consumer has evolved to that of an autonomous control agent that is capable of contributing ancillary grid services to improve and optimise power system operation [1]. Several grid services in ADNs include: black start, load balancing, congestion management, as well as voltage regulation [2]. Voltage regulation, in particular, is a critical grid service because the intermittent power output of renewable DGs (mostly solar) can push the voltage outside of allowable limits, thereby causing damage to existing infrastructure [3].

This shift to service-oriented smart grids is best exemplified by the concept of transactive energy systems (TES), which seeks to combine the economic objectives of all agents with distributed control techniques to improve grid reliability and efficiency. The guiding principles of TES place special emphasis on enabling non-discriminatory participation by qualified agents, providing observable and auditable interfaces, and ensuring that all agents are accountable for standards of performance [4]. While many of the aforementioned grid services have been explored in the context of transactive energy [5], voltage regulation has not been adequately explored and a review of previous work reveals two main shortcomings. First, the incentive calculation is typically controlled by a centralised entity and the incentive amount is kept constant, regardless of the level of contribution of the agent in mitigating a voltage violation [6–8]. Second, the concept of contracting, auditing, and enforcement of contracts is not mentioned, where prosumers must be held accountable when they accept a contract to resolve a voltage violation and should be penalised if they do not comply. Both these shortcomings prevent the realisation of a TES where a competitive marketplace can be established to enable agents to bid and negotiate for the right to procure services, which is how many deregulated energy and ancillary markets are executed in present times [9].

Considering these shortcomings, blockchain technology has emerged as a suitable platform for the implementation and auditable governance of decentralised marketplaces in the context of TES [10]. Blockchains are a type of distributed ledger, where each agent maintains a local copy of the ledger and a consensus algorithm is used to ensure consistency amongst all agents. Transactions submitted to the blockchain network during a given time period are collected in a discrete block of data, verified by agents against a set of rules that the network is governed by, and then appended to the end of the blockchain in an immutable, tamper-proof fashion. Transactions can be automatically enforced by the use of smart contracts, which are software applications that are deployed to the blockchain and auto-execute based on the state of the ledger. As such, the ethos of TES align well with the ethos of blockchains, and address the two aforementioned shortcomings in previous work. Namely, blockchains allow for the incentive scheme and bidding process for the voltage regulation service to be decentralised by executing it as a smart contract that is verified by all participating agents, thereby enabling non-discriminatory governance. The immutability property of the blockchain ledger enables auditability and trust. Finally, the smart contract can enforce the terms of a service contract by penalising an agent for not resolving a voltage violation in an acceptable manner.

Blockchain-based TES has proven popular in both academic and real-world initiatives. However, its main use case has been applied to the implementation of peer to peer (P2P) energy trading systems, which is the focus of ~33% of all active projects [10]. The objective of such systems is to exemplify how the implementation of local energy markets can result in the mitigation of large power imbalances while also unlocking new revenue streams for agents. The most publicised example is the Brooklyn microgrid (developed by LO3), where a neighbourhood of ten homes can directly sell their excess solar energy to each other at a price and time of their choosing. The system is implemented via Ethereum-based smart contracts [11]. Further variations in academia can be found in [12–14], where all of the work utilises smart contracts to facilitate transactions amongst agents as well as...
to implement a mathematical function to ensure that the market price of electricity is fair and equitable (social welfare maximisation, double auction). However, the aforementioned work does not mention enforcement of unresolved contracts, nor does it discuss how blockchains or smart contracts could be utilised as a mechanism to penalise such occurrences.

To that end, this paper proposes a novel, blockchain-based TES that enables agents to participate in a decentralised marketplace and offer voltage regulation as a grid service within ADNs. It is worthwhile to note that this paper considers grid-connected microgrids to be miniature versions of ADNs as in [15], and thus, without loss of generality, the proposed system can be applied to grid-connected microgrids as well. The implementation of the system is exemplified within an ADN that is divided into several zones, where each agent must regulate the voltage in its own zone. In the event of a voltage violation that the agent cannot resolve on its own, the agent can initiate the process of soliciting bids from neighbouring agents and awards a service contract to the most suitable bid. The responding agent must then resolve the voltage violation of the initiating agent using control actions of available DGs within its zone. The proposed system maintains a reputation rating for each agent on the ledger that is analogous to a credit score, and is increased or decreased proportionately for each successful or unsuccessful mitigation of a voltage violation, which directly impacts the ability of the agent to procure bids in the future to earn revenue. The agent negotiation process is implemented as a smart contract and is based on a modified version of the contract net protocol (CNP), where all service contracts are enforced by the smart contract through checking the latest power measurement values on the ledger to determine if the voltage violation has been resolved successfully. To validate the system as a proof of concept, two sets of experiments are presented. The first experiment utilises a modified IEEE-33 bus feeder as the system-under-test, and executes the proposed system to ensure that both overvoltage and undervoltage violations can be mitigated. The second experiment involves a real-world implementation at a small, low-voltage, grid-connected Canadian microgrid to mitigate overvoltage violations. Both experiments utilise the hyperledger fabric (HLF) framework for the implementation of the blockchain.

The contributions of this paper can be summarised as follows:

(i) Developing a novel implementation of a distributed voltage regulation algorithm using blockchain technology. To the best of the authors’ knowledge, only one previous paper has explored voltage regulation in this context, but neglects to explore how smart contracts can enforce service contracts, nor does it explore incentive schemes for voltage regulation services provided. [16].

(ii) Modifying the existing CNP by adding an enforcement stage to be utilised as a smart contract in order to handle the initiation, negotiation, and finalisation of all service contracts between agents.

(iii) Execution of a real-world experiment at a Canadian, grid-connected microgrid to demonstrate how voltage violations can be mitigated by the deployment of the proposed system.

The organisation for the remainder of the paper is as follows. Section 2 provides a detailed overview of blockchain technology, while Section 3 presents the system modelling of a typical power distribution system and defines the mathematical relationship between a power injection of an DG and the system voltage. Section 4 introduces the design principles and implementation of the proposed TES. Section 5 presents the results of the simulated and real-world experiments, while Section 6 is reserved for conclusions.

2 Review of blockchain technology

2.1 Blockchain components

At its core, a blockchain is a decentralised, transaction-based state machine that immutable tracks P2P transactions from different agents and utilises distributed consensus techniques to ensure that all agents are in agreement with the current system state [17]. All transactions are stored on a ledger that is shared and maintained collectively by all agents within the blockchain network, where transactions are any pieces of data or commands that might alter the current state of the system. Examples of transactions may be a money transfer between agents, or a measurement of a unit of renewable energy from the DG of an agent. All transactions are collected together into discrete blocks of data that are chained together in chronological order, such that the evolution of system state is transparent to all agents on the network over time.

Fig. 1 shows a block diagram of a general blockchain system, where a ledger of discrete blocks (Bk) store a set of transactions (Tik), where k is the discrete timestep and i is the index of the transaction within the block. A block has a maximum block size Bmax that defines the total number of transactions that can be stored within it, characterised by \( T_{ik} \), i.e. \( i \in \{1...B_{max}\} \). The validity of each transaction is checked by agents within the blockchain network using distributed consensus techniques [18], and once verified, each block is timestamped and encoded with a unique identifier using a cryptographic hash function that is referred to as a block hash. The block hash is uniquely composed of all the data stored within the block \( Y(B^i) \), and as such, any tampering of the block content will result in a completely different block hash that can be detected. Each previous block hash \( Y(B^{i-1}) \) is also stored within the subsequent block of the ledger and chained together in sequence, resulting in ledger data that is tamper-proof and immutable.

Also depicted in Fig. 1 are smart contracts, which are software applications that automatically execute transactions amongst agents based on ledger data and an associated set of terms and conditions. Smart contracts are deployed to the blockchain in much the same manner as regular transactions and must be validated by agents on the network. As the smart contracts have access to the shared ledger, they have the ability to enforce the terms and conditions of the contract. An example of a smart contract could be the transfer of ownership of an asset contingent upon the transfer of a sum of money, or the checking of power measurements on the ledger to determine if an agent operated a DG correctly to resolve a voltage violation. The computational burden of the execution of the smart contract, as well its participation in the consensus procedure is taken on by a computing resource referred to as a node. An agent
transactions based on the smart contract after other agents have also verified the transaction using their respective nodes. It is worthwhile to mention that each agent must fully authenticate their identity within the blockchain network by utilising public/private key pairs to digitally sign their transactions, such that each individual transaction can be uniquely identified as belonging to the correct agent. These public/private key pairs (PK, SK), along with the current account balance of the agent (whether in fiat currency or cryptocurrency) is stored in a data structure referred to as a wallet.

2.2 Consensus mechanisms
As mentioned earlier, consensus mechanisms play a vital role in blockchains to verify the authenticity and validity of all transactions between agents. Due to the fact that the ledger is shared amongst all network participants and must reflect a single point of truth, there must be general consensus amongst all agents with respect to the current system state. In general, consensus mechanisms aim to be byzantine fault tolerant (BFT), which refers to the ability for a network of agents to reach consensus even if a number of agents experience communication loss, or intentionally send incorrect responses to other agents during the validation process [19]. To establish BFT, it has been mathematically proven that a minimum of two-thirds of the number of agents must return the same result when verifying a block of transactions (or the anticipated result of the execution of a smart contract) [19].

The most popular consensus algorithm in use today is proof of work (PoW), used in platforms such as Bitcoin and Ethereum [20]. In PoW, all agents are incentivised to participate in the consensus process, where agents leverage computational resources within their node to validate transactions for a monetary reward (often referred to as mining). The validation of an upcoming block of transactions involves solving a cryptographic puzzle that is composed of a unique combination of Y(B^2i) and Y(B^(P+1)), and the first miner to solve the puzzle receives a reward. Once a miner solves the puzzle, it broadcasts the solution to all other miners that verify its correctness and the new block is added to the chain. However, PoW relies on brute force methods to solve the puzzle, resulting in significant scalability concerns due to the computational complexity and energy expenditure required by the consensus process [21]. The energy consumed from the PoW consensus process on the Bitcoin network in the year 2017 was equivalent to the annual energy consumption of Ireland, which is in excess of 30 TWh [22].

In contrast, an example of a consensus algorithm that does not use mining techniques is practical byzantine fault tolerance (pBFT), which is used by platforms such as HLF [23]. Within pBFT, one agent is designated as a leader, and the other agents are designated as followers. During the validation process, the leader agent broadcasts the block of transactions (or proposed smart contract to be executed) to all other followers, who in turn, validate the transactions. The transactions are successfully confirmed only when more than two-thirds of the agents return an identical confirmation result. Compared to PoW, pBFT is minimally computationally intensive, however, its communication complexity is O(n^2) because every agent must communicate with every other agent during the validation process [24]. As a result, pBFT is ideally suited when there are a limited set of agents within the network, which is further discussed in the next subsection.

2.3 Permissioned and permissionless architecture
Generally, blockchains can be divided into two categories: permissionless (public), or permissioned (private or hybrid) [23]. Permissionless blockchains, such as Bitcoin and Ethereum, are open to the public, and any agent can begin to make transactions after registering on the network. In contrast, permissioned blockchains, such as HLF, require an invitation to join the network.

As a result, the identities of the agents within the network can be known. Permissioned blockchains have found success in use cases where there are shared business networks in which agents (i) might have conflicting incentives or (ii) do not necessarily trust each other, yet must still conduct transactions for a shared business objective [25]. Since an invitation is required to participate in a permissioned blockchain, the consensus algorithm does not require all agents to participate in the consensus process, thereby alleviating the concerns about excessive energy use associated with permissionless blockchains [26]. The requirement of the invitation also mitigates concerns of Sybil attacks, where a malicious agent creates fake identities, registers the identities onto the network, and attempts to undermine the consensus process [27].

Recent work has benchmarked permissioned blockchains (HLF) and permissionless blockchains (Ethereum) and has revealed that HLF consistently outperforms Ethereum within scalability metrics such as transaction throughput (transactions per second), average latency of transactions (total time required for transaction to be committed to the ledger) and computational burden (peak memory usage) [28]. At 16 nodes, HLF is capable of processing 1000 transactions per second at 50 s latency, while Ethereum processes ~200 transactions per second at 75 s latency [28]. In terms of peak memory usage, one benchmark involved the sorting of an array of 10 million elements, which resulted in a peak memory usage of 473 MB for HLF and over 22,000 MB for Ethereum [28]. It should be mentioned, however, that HLF cannot support more than 16 nodes, while Ethereum does not have a theoretical limit [29].

3 Problem of voltage regulation in ADNs and its application in distributed TES
This section introduces the problem of voltage regulation in ADNs by presenting the mathematical modelling of ADN components, as well as the impact of DG control actions on the voltage within ADNs. This is followed by a subsection that explores how the voltage regulation problem can be addressed by blockchain-based systems.

3.1 Mathematical modelling of ADNs
Fig. 2 shows a typical radial distribution feeder within an ADN, where DGs and loads are connected. Renewable DGs, primarily based on solar and wind energy, are typically preset to operate at maximum power point tracking mode [30]. However, the inverter of the DG is capable of setting operating points for active and reactive power to provide support for voltage at its point of common coupling (PCC) with the main grid. As such, active power (P) and reactive power (Q) represent the control variables that an agent can set when operating the DG. In general, at time instant t, the active and reactive power flow in branch i, and the voltage at bus i, could be represented with respect to bus j as follows [31]:

\[ P_i(t) = P_j(t) + \sum_{n=1}^{n-1} (P_{L_i}(t) - P_{BESS_j}(t) - P_{DG_i}(t)) \]

\[ Q_i(t) = Q_j(t) + \sum_{n=1}^{n-1} (Q_{L_i}(t) - Q_{BESS_j}(t) - Q_{DG_i}(t)) \]
where $n$ is the line index, $(P_n, Q_n)$ are the output active/reactive powers, $(P_{DG,i}, Q_{DG,i})$ are the active/reactive loads, $(P_{in,j}, Q_{in,j})$ are the injected active/reactive powers from the DGs, $V_i$ is the voltage at each bus, and $(r_{in,j}, x_{in,j})$ is the resistance/reactance of each line. From (1)–(3), it can be seen that changes in the injected absorbed active and reactive power play a dominant role in shaping the voltage profile of the overall network. Assuming the system is in steady state allows the neglecting of the change in load demand, and further linearisation of (3) by neglecting the change in load demand can be expressed as

$$V_i(t) = V_i(t) - 2 \sum_{n=1}^{N} \left( r_{in,j} P_{in,j}(t) + x_{in,j} Q_{in,j}(t) \right)$$

(3)

The voltage at the substation is typically very stiff given the overall impedance of the main grid. Given this, the substation voltage can be expressed as

$$\Delta V_i(t) = \frac{1}{{\sqrt{(t-1)}}} [V_i(t) - \Delta V_i(t)]$$

(4)

$$\Delta P_{DG,i}(t) \sum_{n=1}^{N} r_{in,j} + \Delta Q_{DG,i}(t) \sum_{n=1}^{N} x_{in,j}$$

(5)

It is now possible to determine the sensitivity of each bus to any $P/Q$ power injection/absorption as follows:

$$SP = \frac{\partial V_i}{\partial P_j} = \sum_{n=1}^{N} \left( \frac{-1}{V_i(t-1)} \right) S_{Qj} = \frac{\partial V_i}{\partial Q_j} = \sum_{n=1}^{N} \left( \frac{-1}{V_i(t-1)} \right)$$

(6)

Further, the impact of any control action (changes in $P/Q$) at bus $j$ on the voltage of bus $i$ can be expressed by

$$\Delta V_i(t) = \frac{1}{{\sqrt{(t-1)}}} [V_i(t-1) - \Delta V_i(t)]$$

(7)

$$\Delta P_{DG,i}(t) \sum_{n=1}^{N} r_{in,j} + \Delta Q_{DG,i}(t) \sum_{n=1}^{N} x_{in,j}$$

From (7) it can be seen that the penetration of large amounts of DGs within power systems can both cause and mitigate voltage violations. Large injections of active power may result in an overvoltage violation, whereas the curtailment of active power and/or absorption of reactive power have the potential to reduce the voltage and resolve the violation.

### 3.2 Potential application of blockchain for voltage regulation

Conventional approaches for voltage regulation within ADNs have followed two approaches: centralised and distributed (non-blockchain based, however) [32]. A centralised approach relies on a single, trusted authority to unilaterally control all agents within the entire ADN to globally regulate the system voltage. On the other hand, a distributed approach obviates the need for a central authority by spatially partitioning the ADN into several zones, where each zone is governed by an agent that regulates the zonal voltage by controlling the DGs in its own zone. As shown in Fig. 3, zonal partitioning requires agents to strategically coordinate their control actions via P2P exchanges to ensure that the overall system voltage remains within normal operating conditions.

Yet, both approaches do not provide explicit mechanisms for defining and maintaining trust among agents, where the notion of trust within the application of voltage regulation is composed of (i) validating that the incentive received by an agent for taking a control action is fairly calculated and delivered; (ii) validating that the agent indeed took the expected control action to mitigate a voltage violation; and (iii) maintaining an impartial, auditable interface to track the performance of the agent over time. A centralised approach can easily use a central controller to validate all control actions of the agents and track agent performance, but since its authority is unchallenged, the controller cannot be fully trusted to make equitable economic decisions for all agents [33]. A conventional distributed approach enables agents to formulate their own incentive scheme and make rational economic decisions via P2P negotiation, however, these approaches do not address how the agents will impartially track the performance of other agents.

Blockchain technology can address these trust issues by using the immutability principles of shared ledgers and smart contracts. All agents would be required to store measurements of zonal voltage and power output of their DGs to the shared ledger, where the measurements would be stored as transactions as shown in Fig. 1. Smart contracts would enable agents to offer voltage regulation services to other agents in the form of a digital service contract, which would include terms and conditions such as: the bidding price of the service, the voltage deviation to be corrected, as well as the time expiry of the contract. Crucially, the smart contract would be able to enforce the terms and conditions of the service contract by querying the power quality measurements on the ledger to ensure that the voltage deviation was corrected within the specified time. In addition, the agents would collectively maintain an agent reputation rating on the ledger, which the smart contract would autonomously update when a voltage violation is resolved appropriately. In this way, the functions of calculating agent incentives, validating agent control actions, and maintaining auditable metrics of agent performance are performed collectively by all agents, since the agents must reach consensus to maintain the state of the ledger and execute smart contracts.

### 4 Blockchain implementation of proposed TES

#### 4.1 Blockchain architecture design of proposed TES

When considering design principles for designing blockchain-based systems, careful thought is given to its design architecture in terms of security and scalability [34]. In the context of TES, especially with the application of voltage regulation, a permissioned blockchain seems to be more suitable than a permissioned blockchain for two main reasons. First, participation of the agents within the system should be qualified, invite-only, and agreed upon by all agents within the network to decrease cybersecurity concerns with respect to malicious agents and Sybil attacks [35]. Thus, an invited agent would be assigned a specific ID ($A_n$), as well as a public/private key pair ($PK_n$, $SK_n$) such that the agent would be able to digitally sign transactions within the blockchain network to prove their authenticity. Second, the scalability of the overall system in terms of transaction speed and computational burden should be kept as low as possible. As discussed earlier, a permissionless blockchain architecture achieves higher transaction throughput, lower latency, and significantly less computational burden when compared to permissioned blockchain
architectures. In the case of HLF, a maximum constraint of 16 agents within a network is indeed a limiting factor, however, this limit may be overcome by assigning larger spatial zones of operation to an agent as in [30].

To further improve the scalability of blockchain-based systems, a common strategy is to minimise the data stored on the ledger, as well as to reduce the complexity of the logic executed by the smart contracts [36]. Since both these blockchain components require consensus, and therefore, heavy computational burden, only critical pieces of data that are required by the smart contract to enforce any service contracts between the agents should be stored on-chain. Any data, or logic that does not require intervention by other agents can be executed off-chain. For the application of voltage regulation, the required logic is defined in (7), where the data items include: measurements of the active/reactive power of the DG, a consensus, and therefore, heavy computational burden, only critical details such as the IDs of the agents within the contract, bid price, voltage deviation to be corrected, and time expiry of the contract.

It should be noted, however, that obtaining voltage measurements by deploying smart meters at every bus within the system is infeasible. Instead, smart meters can be strategically dispersed throughout the spatial zones of the agents, and the agents can use state estimation techniques to estimate the voltage at every bus within its zone. Methods for optimal allocation and placement of smart meters in distribution systems, as well as distributed state estimation techniques can be found in [37]. Given this, the agent can query smart meters locally and execute the state estimation process off-chain. Upon obtaining the estimate of the voltage profile of its zone, the agent can readily determine if there is a voltage violation, and if so, submit the voltage deviation measurement to the ledger. The resultant agent bidding process and granting of service contracts would then commence on-chain. In this way, the amount of data stored on the ledger is reduced, the contribution of each DG towards the violation mitigation is explicitly known, and automatic enforcement can be performed by the smart contract by querying the latest measurements on the ledger.

4.2 Agent coordination process

The agent coordination process involves the detection of a voltage violation by an agent within its zone, the corresponding solicitation for neighbouring agents to offer bids for mitigating that violation via control actions of their DGs, and finally, the negotiation, assignment, and enforcement of the final service contract between agents. This subsection will provide more detail on each of the three stages, along with a flowchart that is provided in Fig. 4 to support the description of the process. Additionally, a summary table of on-chain and off-chain activities within each stage is provided in Table 1. It should be noted that for the purposes of this paper, and without loss of generality, the assumption is that each zone has only one DG, each agent does not especially trust each other, and that the configuration of the zonal boundaries is pre-decided to satisfy the power adequacy constraint as in [30].

4.2.1 Stage I: detection of zonal voltage violation: Referring back to Fig. 3, each zone is defined as zone $z_i$, and governed by an agent with unique ID $A_{z_i}$. All zones are coupled with each other such that they have one bus in common, referred to as the point of zone coupling $PZC_z$. A small deployment of smart meters is placed at specific buses within the zone, which the agent can query in real time to obtain measurements of power generation, demand, and voltage at specific buses. As discussed earlier, state estimation can be performed by the agent to take these measurements and subsequently estimate the voltage at each bus within the zone. Upon computing the estimate of all voltages within its zone, the agent can determine the presence of any voltage violation and attempt to resolve it locally and/or initiate negotiations with neighbouring agents to resolve the violation. In the latter case, the agent must determine the voltage required at the PZC of its neighbouring zones that would effectively mitigate the violation in its own zone. This can be determined from the following sensitivity equation:

$$\frac{\partial V_{PZC_z}}{\partial V_i} = \frac{V_{PZC_z}(t) - V_{PZC_z}(t-1)}{V(t) - V(t-1)}$$

(8)

where $V_{PZC_z}$ is the voltage at the PZC of zone $z_i$, and $V_i$ is the voltage at the violated bus $i$, and $V_{PZC_z}(t)$ is the target setpoint required to mitigate the voltage violation.

From the perspective of the blockchain implementation, the querying of real-time data from the smart meters, as well as the state estimation process can be done off-chain to reduce the amount of data stored to the ledger. The estimate of the zonal state of the agent does not hold value for the rest of the agents, nor is there a trust issue that requires consensus from other zones. Agents may query the smart meters through standard communication protocols such as MODBUS, and use the measurements to execute the state estimation process using its own computational resources.

4.2.2 Stage II: determination of agent bid prices: A realistic implementation of a TES involves a competitive market that allows disparate agents to offer bids for services in return for economic compensation. Therefore, agents are self-interested and rational,

---

**Fig. 4 Flowchart for the agent coordination process**

**Table 1 Summary of on-chain/off-chain tasks in Fig. 4**

| Stage | Task | Execution | Justification |
|-------|------|-----------|---------------|
| I     | State estimation | Off-chain | Voltage estimation does not require ledger data |
| I     | Issue CFP | On-chain | CFP enforcement requires on-chain data |
| II    | CFP evaluation | Off-chain | Pricing strategy does not require on-chain data |
| III   | Bid evaluation | On-chain | Requires ledger data $G_{A_{z_i}}$ |
| III   | Enforcement | On-chain | Requires ledger data $G_{A_{z_i}}$ |

IET Smart Grid, 2020, Vol. 3 Iss. 5, pp. 646-656

This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)
focused on earning as much revenue as possible while also maintaining their zonal voltage. The agent can earn revenue via the strategic operation of the DGs in two ways: selling excess active power back to the grid at the wholesale price of electricity, or modulating the $P/Q$ output of the DG to mitigate a voltage violation as a grid service. An agent must ensure that the control actions it selects for its DGs are economically and operationally viable. The selection of control actions is reminiscent of a classic optimisation problem, wherein the agent finds optimal values for its control actions $\{\Delta P, \Delta Q\}$, depending on device and system constraints, while seeking to be as profitable as possible. It is important to note that the optimisation is run only when there is an instance of a voltage violation, and as such, it is an online process that has time-varying constraints. The control actions available to each zonal DG are its active and reactive power output

$$X_z = \{\Delta P_{DG,z}(t), \Delta Q_{DG,z}(t)\}$$

(9)

The constraints revolve mainly around the physical operational limits of the zonal DGs as well as the physical constraints of the distribution system infrastructure of the zone itself. The constraints for the DGs are as follows:

$$C_{DG,z} = \begin{cases} P_{DG,z}(t) & \leq P_{MAX,z}(t) \\ Q_{DG,z}(t) & \leq Q_{MAX,z}(t) \\ & \land \left( P_{DG,z}(t) + Q_{DG,z}(t) \leq S_{MAX,z}(t) \right) \end{cases}$$

(10)

where $\{P_{MAX,z}, Q_{MAX,z}, S_{MAX,z}\}$ are the maximum ratings of the active, reactive, and apparent power settings of the DG. It is worth noting that $P_{MAX,z}$ is further constrained by the availability of the input fuel of the DG (whether gas, diesel, solar irradiance, or wind speed).

The physical constraints of the zone are to obey the maximum current carrying capacity of each distribution line and to maintain the voltage within specified limits for each bus within the zone $C_z$

$$C_z = \begin{cases} I_{i,z}(t) & \leq I_{CAP,i}^{\text{MAX}} & \forall i \in F^z \\ & \land \left( V_{i,z}(t) \leq V_{i,z}^{\text{MAX}} \right) & \forall i \in M^z \end{cases}$$

(11)

where $I_{CAP,i}$ is the maximum carrying capacity of a distribution line $f$, $\{V_{i,z}^{\text{MIN}}, V_{i,z}^{\text{MAX}}\}$ are the minimum and maximum voltage limits at each bus $i$, $F^z$ is the set of all lines within zone $z$, and $M^z$ is the set of all buses within zone $z$.

Finally, the objective function of each agent can be formulated as

$$F = \max \rightarrow R_i(t)$$

$$R_i(t) = R_{DG,i}(t) + R_{SRV,i}(t)$$

(12)

where

$$R_{DG,i}(t) = PR_{DG,i}(t) \times (P_{DG,i}(t) \times \Delta t)$$

$$R_{SRV,i}(t) = PR_{DG,i}(t) \times (Q_{DG,i}(t) \times \Delta t) + R_{DG,i}(t) \times \alpha$$

(13)

result in a loss of revenue as per (13). As such, a markup factor can be added to the grid service to ensure that the agent receives some profit instead of simply meeting its marginal costs. The value of the markup factor can be arbitrarily decided by the agent.

Solving the above optimisation problem results in the precise values for the $\Delta P$ and $\Delta Q$ setpoints of each DG in the zone that can potentially mitigate a local or neighbouring voltage violation, as well as the final price of the agent bids. Agents may still solicit bids from neighbouring agents while being fully capable of resolving the voltage violation locally if the bids are less expensive than the local operation would cost. It is worthwhile to note that an agent may encounter a scenario where the optimisation problem in Stage II does not return a feasible result, and/or other CFPs are not received by the agent. In this case, the agent will return to Stage I to determine if a voltage violation had occurred in its own zone which was functioning in Stage II, as shown in Fig. 4. If the agent does not return to Stage I, the agent will only be able to respond to other CFPs without ever resolving voltage violations in its own zone.

As discussed earlier, a reputation rating is established amongst all agents to reflect both the ability and trustworthiness of the agent to provide voltage regulation services for the rest of the agents in the system. It is a crucial metric that heavily influences the procurement of voltage regulation services that earn revenue for the agent. The reputation rating of the agent is increased with each successful mitigation of a voltage violation request, with the increase being proportional to the severity of the voltage violation as follows:

$$G_{\text{NOW}} = G_{\text{PREV}} + \gamma \Delta V$$

(15)

where $G_{\text{NOW}}$ is the reputation rating for an agent, the superscripts NOW and PREV are representations of the current and previous reputation ratings, respectively, $\Delta V$ is the magnitude of voltage deviation, and $\gamma$ is a scaling factor that can be arbitrarily set in consensus with all agents within the system. If an agent fails to resolve a voltage violation as per the contract, harsh penalties are imposed on its reputation rating by setting $\gamma$ to a negative value. Reputation ratings are taken into account when agents compete in the bidding process to procure grid services, and a negative reputation rating greatly affects the ability of an agent to generate revenue for itself. This is explored in further detail in the next subsection.

Returning to the blockchain implementation of the proposed system, the execution of the optimisation problem can be done off-chain, as it does not require any of the existing data from the ledger. It is assumed that the sensitivity factors that are stored on the ledger are also available in the local memory of the agent in order to solve the optimisation problem. However, the maintenance of the reputation rating metric is something that must be stored and updated on-chain via consensus, particularly since it is a sensitive metric that affects the revenue generating ability of the agents. As such, allowing a smart contract to automatically adjust the reputation rating of an agent after validating the successful mitigation of a voltage violation is a feasible solution to this problem. This obviates the need for one of the agents, or a central authority, to manage the reputation rating of all agents.

4.2.3 Stage III: agent negotiation process: To lend structure to the negotiation process between the agents, a modified version of the CNP is utilised, which is a negotiation protocol used in multi-agent systems to allow for efficient task allocation between agents [38]. Agents using the CNP can be ‘initiators’ or ‘responders’, in which initiators request assistance from responding agents to complete a particular task. The CNP has three stages, which are: announcement, bidding, and assignment. In the announcement stage, initiators that require assistance for a task send a call for proposal (CFP) to other agents that request them to submit bids to obtain a prospective service contract. In the bidding stage, responding agents compute and submit bids to the initiator. Finally, in the assignment stage, the initiating agent evaluates the contracts and awards them to the agent(s) that have the best proposal. It is
worthwhile to note a responding agent can choose to subcontract a task to another agent, thereby becoming the initiating agent for the subcontract.

An example of the three-stage CNP is described below as an negotiation process between initiating agent $A_{IN}$ and responding agents $A_R$ that are neighbours of agent $A_{IN}$. It should be noted that the magnitude of voltage violation will be addressed symbolically, however, it is assumed that they are represented in the per unit style (p.u.), which represents all system quantities as fractions of a base unit value:

1. **Announcement:** Agent $A_{IN}$ encounters a voltage violation and determines the magnitude of voltage deviation needed at its PZC with agent $A_R$ as $\Delta V_{PZC}$. This is computed using (8). Agent $A_{IN}$ initiates a CFP to agents $A_R$ in the form of a four-tuple $\{A_{ID}, CFP_{ID}, \Delta V_{PZC}, \Delta T\}$, where $A_{ID}$ is the initiating agent ID, $CFP_{ID}$ is the ID of the CFP, and $\Delta T$ is the timeout period (in seconds) that specifies how long the bid will remain valid.

2. **Bidding:** Agents $A_R$ evaluate the CFP by determining its feasibility using (9)-(11). If successful, each $A_R$ sends back a reply in the form of $\{A_{ID}, CFP_{ID}, PR_{BID}\}$, where $PR_{BID}$ is the price of the bid and is computed by (12).

3. **Assignment:** Agent $A_{IN}$ collects all bids and multiplies the bid price with the reputation rating of each agent to determine the final bid price. The lowest bid of the agents $A_R$ is awarded the service contract in the form $\{A_{ID}, CFP_{ID}, DEC\}$, where $DEC$ is a binary decision variable (1 for assignment, 0 for rejection).

It can be seen that the determination of the best proposal is a function of the bid price, as well as the reputation rating of the agent. Therefore, it is vital that the agent maintains a positive reputation rating to enhance its chances of generating revenue for itself. Within a blockchain implementation, it can be seen that all agent transactions should be executed on-chain since each transaction contains all the pertinent information that can be used to enforce the awarded service contract. This information includes: the bid price agreed upon by the agents, the timeout period until the bid is valid, and the magnitude of voltage violation the agent agrees to mitigate. Using this information, a smart contract implementation of the CNP would be used to (i) validate that the initiating agent has enough money to pay the responding agent, (ii) verify that all bids received past the timeout period are invalid, and (iii) enforce the service contract post-assignment by checking the $P/Q$ measurements of the responding agent on the ledger along with the voltage measurements at the PZC to determine that the violation was mitigated by the rightful service provider.

The enforcement of the service contract between agents warrants a modification to the existing CNP by adding an enforcement stage after the assignment stage. If the smart contract determines that the service contract was not successfully resolved by an agent, penalties are levied upon the reputation rating of the responding agent and the announcement stage is recommenced with a new voltage deviation magnitude as in (15).

### 4.3 Summary of blockchain implementation of TES

As a summary, Fig. 5 is presented to recap the complete implementation of the proposed blockchain-based TES, which is a mirror image of the abstract, generalised blockchain architecture in Fig. 1. As seen in Fig. 5, the ledger data includes: voltage measurements at the shared PZC of all agents, measurements of the power output of each DG, a sensitivity matrix, an array of reputation ratings for the agents, as well as a data structure that represents all service contracts between agents. Implementation details with respect to the smart contract are included in the following subsection.

**4.3.1 Smart contract implementation:** Recall that the primary reasons for the development of the smart contract are to (i) automate the negotiation/bidding process of voltage regulation service contracts using CNP; (ii) enforce the terms and conditions of each service contract to ensure that any voltage violations are actively mitigated; and (iii) maintain a reputation rating for all agents that reflect their trustworthiness and ability to resolve voltage violations. Achieving the aforementioned functionality via a smart contract obviates the need for a centralised, trusted authority to manage such sensitive agent interactions. As such, the implementation of the smart contract requires the development of logic/code that enables the initialisation of an agent account, the maintenance of the CFP lifecycle (including CFP creation and assignment), as well as the final enforcement of the CFP. Thus, the smart contract proposed in this paper is implemented as a set of functions that have the ability to access/update data on the ledger and use the data to provide the requisite functionality. The pseudo-code of each function within the smart contract is described in Algorithm 1 (Fig. 6), while a generic description is provided below.

It is worth noting that for an agent to utilise a smart contract, it must invoke the specific function of the smart contract it wishes to execute. The smart contract function will execute locally for the agent, however, any resulting changes to the shared ledger will only be finalised after other agents have validated the transactions during the consensus process (as discussed in Section 2.2).

For the remainder of the text, $A_{ID}$ will describe the unique ID assigned to an agent, but to differentiate between initiating agents and responding agents, the terms $A_{IN}$ and $A_R$ will be used, respectively.

**initAccount():** This function initialises an account for the agent and assigns it an ID based on its zone number. To verify the identity of the agent calling this function, an input message that defines the Zone ID and is digitally signed by $SK_{ID}$ must be provided to the function. Upon verification, the functional initialises the number of dollars in the agent wallet to zero ($B_{ID}$).

**createCFP():** This function is called by an initiating agent, $A_{IN}$, and creates a CFP that stores the Agent ID, a unique ID for the CFP, the required change in voltage needed at the PZC ($\Delta V_{PZC}$), as well as a timestamp for when the bid will expire ($expiryCFP$). The $A_{IN}$ signs the CFP with $SK_{IN}$ and this generates a new CFP on the blockchain.

**replyCFP():** This function is called by agents $A_R$ that respond to an active CFP with a bid price. The function validates the incoming bid by checking (i) if the price of the bid is less than the balance of
function initAccount()
Input: ZoneNum→Z, Msg(Z)
Verify identity of agent using Msg(Z)
Initialize account with $A_{IP} ← Z$ and $B_{ID} ← 0$
function createCFP()
Input: $A_{IN}, ΔV_{PZC}, c, expiryCFP$
Generate CFPID and deploy new CFP
Output: $A_{IN}, CFPID, ΔV_{PZC}, c, expiryCFP$

function replyCFP()
Input: [$A_{ID}, CFPID, PR_{BID}, bidTime] → bid
Create list to hold valid bids→validBids
if $PR_{BID} ≤ B_{IN}$ and $bidTime ≤ expiryCFP$
append bid to validBids

function assignCFP()
Input: validBids
if currentTime ≥ bidExpiry then
Multiply each $G_{A_{ID}}$ with $PR_{BID}$ in validBids to finalBids
Return bidWinner ← argmin(finalBids)

function enforceCFP()
Get P/Q measurements from ledger for $A_{ID}$ over $ΔT$
Use (7) to calculate $ΔV_{PZC}$
compare to $ΔV_{PZC}$ stated in service contract → success
if success then
Update reputation rating of $A_{ID}$ using (15)
Transfer $PR_{BID}$ from $A_{IN}$ wallet to $A_{ID}$ wallet
else
Decrement reputation rating of $A_{ID}$ using (15)

Fig. 6 Algorithm 1: Smart contract implementation

Table 2 Settings of DGs within studied IEEE 37-bus system

| DG index | $P_{MAX}$, p.u. | $Q_{MAX}$, p.u. | $P_{DG}$, SMVAR | $α$ |
|----------|-----------------|-----------------|-----------------|-----|
| 1        | 1.0             | 0.6             | 1050            | 1.1 |
| 2        | 1.5             | 0.0             | 0               | 1.3 |
| 3        | 1.0             | 0.5             | 1200            | 1.2 |
| 4        | 1.0             | 0.5             | 1100            | 1.1 |

5 Experimentation results

Two sets of experimentation results have been prepared to demonstrate the efficacy of the proposed system. The first set of results is conducted on a modified IEEE 33-bus system that is depicted in Fig. 3. The second set of results is from a real-world experiment conducted at a low-voltage, Canadian microgrid. The real-world experiment was conducted using HLF as the blockchain framework, with agents being implemented on myRIO and Raspberry Pi microcontrollers. Agents were able to submit measurements, bids, and generate smart contracts to HLF by using standard HTTP web services. MATLAB was used to simulate the ADN in the first experiment, while the second experiment was conducted at the microgrid and used the MODBUS protocol to control the DGs.

5.1 Case study: IEEE 37-bus power system

The settings for the DGs within the system are shown in Table 2, where the pricing for reactive power follows the ranges of real-world power system operators as reported in [39]. The markup factors are set arbitrarily. The reputation ratings for all agents are initialised to 1, while its scaling factor is set to 10. The DG type is assumed to be solar.

The experimental results are depicted in Figs. 7–10, based upon a 12 h snapshot between the hours of 6:00 and 18:00. To improve the readability of the agent interaction process in Figs. 9 and 10, a reference table for the definition of the tuple items within the message is provided in Table 3. Fig. 7 shows the active power generated from the DGs, as well as the corresponding price for active power in the wholesale electricity market, which was gathered from the Ontario transmission system operator data archives.

This work assumes that the DGs use the wholesale price of electricity as the price of active power, $PR_{A}(t)$, which varies hourly. As can be seen in Fig. 7, DGs 1–3 have steady active power production throughout the time period, however, DG 2 experiences an outage from the hours of 7:30 to 9:30. The outage causes an undervoltage violation in Zone 2 at bus 33, where the voltage profile of bus 33 can be seen in Fig. 8. This results in Agent 2 issuing a CFP to both Agents 1 and 3 for voltage support. This is depicted in the announcement stage of Fig. 9, where $CFP_{ID}$ 1 is issued to both Agent 1 and Agent 3 with the required change in voltage magnitude at the PZC for both agents is specified at 0.0033 and 0.0034 p.u., respectively. As seen in the figure, the CFP then enters the bidding stage, where Agent 2 computes a total reactive power injection of $\Delta Q = 0.5691$ p.u. and responds to the CFP with a bid of $704$ using (12). Agent 3 follows a similar process that results in a setpoint of $\Delta Q = 0.1296$ p.u. and a bid of $257$.

The smart contract evaluates the proposals based on price, since the reputation ratings are equal at this stage, and assigns the service contract to Agent 3. This can be seen in Fig. 9, where the third element in the tuple of the message sent to Agent 3 is 1 (signifying a winning bid). The smart contract then enters the enforcement stage, querying the ledger to receive the measurements of voltage from the PZC. When the time limit of the contract expires, the latest measurement reveals that the violation was not mitigated. Subsequently, the smart contract decrements the reputation rating of Agent 3 by 0.534 using (15), resulting in a lower reputation rating of 0.966. After updating the ledger, the smart contract issues a fresh CFP (ID 2) to Agent 1. Agent 1 successfully resolves the voltage violation as shown in Fig. 8, and the smart contract validates the mitigation within the enforcement stage. This can be noticed in Fig. 9, where the third element in the tuple of the message sent to Agent 1 is ‘S’; this signifies the successful completion of the service contract. Subsequently, Agent 1 collects its payment and also obtains an increase in reputation rating (1.0033). The service contract is honoured for 2 h until DG 2 comes back online (9:30), resulting in a net earnings of $1408 for Agent 1.

DG 2 comes back online at 9:30, and reaches maximum power capability at 10:30. As seen in Fig. 8, this causes an overvoltage violation at bus 37 in Zone 2. Agent 2 again generates a smart contract (ID 3) for both neighbouring agents as seen in Fig. 10. Agent 2 also computes the total loss to its revenue if it were to curtail the power locally, which is a total curtailment of $\Delta P = 0.5$ p.u., or $\Delta P = 0.38$ p.u. with a corresponding bid price of $750$. Agent 3 executes (12), and returns with setpoints of $\Delta Q = -0.50$ p.u. and $\Delta P = -0.28$ p.u., however, seeking to reduce the reactive power needed in an effort to reduce the cost of the bid, Agent 2 issues a CFP to Agent 4 downstream. This can be seen in Fig. 10, where Agent 3 generates a new CFP (ID 4), which specifies a voltage adjustment at the PZC between Agents 3 and 4 to be $-0.0106$ p.u.. Agent 4 computes the setpoints of its DGs to be...
ΔQ = −0.50 p.u and ΔP = −0.2 p.u, and returns a bid price of $458 to Agent 3. Agent 3 rejects this bid since the price of its own control actions would itself result in an aggregated bid of over $1000. Seeking to improve its reputation rating, Agent 3 reduces its bid by 90% by applying a negative markup factor, essentially providing the service at marginal cost. After finalising the bids, the smart contract applies the reputation rating of Agent 2 to compute an aggregate bid price of $70, and assigns the service contract to Agent 3 since it is less than the bid of Agent 1 and also less than the $77 it will cost Agent 2 to solve the problem locally. The smart contract then executes the enforcement stage and validates that the voltage violation has been mitigated as seen in Fig. 8. After the enforcement stage, the smart contract generates a successful completion message to Agent 3 (as shown in Fig. 10). The smart contract upgrades the reputation rating of Agent 2 to 0.99. The service is provided for 5 h, resulting in a net savings of Agent 2 of $35.

5.2 Case study: real-world microgrid

The proposed system was validated at a low-voltage microgrid referred to as the Kortright Centre Microgrid (KCM). The KCM is a grid-connected microgrid that is connected directly to a medium voltage feeder of a larger distribution network. A simplified single-line diagram of the KCM is shown in Fig. 11, where two agents are deployed at buses 3 and 4. At bus 3, there is a 20 kW solar DG (SolarEdge) that is under the command of Agent 1, and at bus 4 downstream, there is a 75 kWh battery bank (Xantrax 6848) and 5 kW electronic load (e-load) that is under the command of Agent 2. In terms of measurements, there is a smart power quality meter deployed at the PCC between the KCM and the main grid. All agents control the devices via the MODBUS protocol. The KCM suffers from regular overvoltage violations due to excessive solar generation and minimal loading.

The objective of this case study is to demonstrate the execution of a smart contract between Agent 1 and Agent 2 in order to mitigate an overvoltage violation in Zone 1 within a real-world setting. In particular, the testing and validation of end-to-end communication between the agents and the smart contract/ledger is of particular importance. To that end, a block diagram of the system implementation is provided in Fig. 12, where the blockchain is developed within the HLF framework using the Hyperledger Composer tool.

A REST API is configured within HLF to provide agents with the ability to invoke specific functions of the smart contract. Thus, the REST API is used to create, assign, and enforce CFPs, as well as to submit measurements from DERs onto the ledger. Once a CFP is issued to the correct agent, a push notification is sent by the smart contract to the intended agent, where the agent translates the contract details into specific control actions to control their DERs. Agent 1 is implemented on a National Instruments PXI controller, Agent 2 is implemented on a Raspberry Pi 2 B+ microcontroller, while the blockchain network is hosted on a Linux-based laptop.

Table 3 Description of tuple messages in Figs. 10 and 11

| Stage   | Tuple                  | Notes                  |
|---------|------------------------|------------------------|
| Announcement | \(A_{IN}, CFP_{ID}, ΔV_{PCC, \text{c}}, \text{expCFP}\) |                        |
| Bidding | \(A_{B}, CFP_{ID}, PR_{BID}\) |                        |
| Assignment | \(A_{AS}, CFP_{ID}, DEC\) | DEC accept(1) or DEC fail(0) |
| Enforcement | \(A_{EN}, CFP_{ID}, RES\) | RES success(S) or RES failure(F) |

Fig. 7 Plot of wholesale price of active power and DG active power generation

Fig. 8 Voltage profiles of buses 33 and 37 with (C) and without (U) distributed voltage regulation

Fig. 9 Agent message log for a scenario of undervoltage violation

Fig. 10 Agent message log for a scenario of overvoltage violation

Fig. 11 Simplified single-line diagram of the KCM

Fig. 12 Block diagram of the system implementation

Fig. 13 Description of tuple messages in Figs. 10 and 11

Table 3 Description of tuple messages in Figs. 10 and 11

| Stage   | Tuple                  | Notes                  |
|---------|------------------------|------------------------|
| Announcement | \(A_{IN}, CFP_{ID}, ΔV_{PCC, \text{c}}, \text{expCFP}\) |                        |
| Bidding | \(A_{B}, CFP_{ID}, PR_{BID}\) |                        |
| Assignment | \(A_{AS}, CFP_{ID}, DEC\) | DEC accept(1) or DEC fail(0) |
| Enforcement | \(A_{EN}, CFP_{ID}, RES\) | RES success(S) or RES failure(F) |
As seen in Fig. 13, Agent 1 detects an overvoltage violation of 1.0541 p.u. at 12:31:16. This is primarily due to the excess active power generation by the solar DG of Agent 1, which can be seen in Fig. 14. The net demand of the KCM is positive, indicating that 0.18 p.u. of active power is flowing back towards the main grid. Agent 1 invokes the createCFP function of the smart contract using the REST API and generates a CFP that requires the voltage at the PZC between Agent 1 and Agent 2 to be $\Delta V = -0.02$ p.u. Agent 2 does not have inverter based devices that have reactive power modulation ability, and the execution of (12) returns back $\Delta P = -0.06$ p.u. Note that since there are only two agents available, this experiment does not involve bidding. Nevertheless, Agent 2 receives a notification that a CFP has been generated and utilises the replyCFP function to submit a bid for the CFP. Since this is the only bid, the smart contract assigns the CFP to Agent 2 using the assignCFP function and Agent 2 takes action by charging its battery bank at 12:31:26. The change in voltage can be seen in Fig. 13, where the voltage at the PZC of Agent 1 and Agent 2 reduces by the setpoint of $\Delta V = -0.02$ p.u. Fig. 14 correspondingly shows a reduction of $\Delta P = -0.06$ p.u. in the net demand of the KCM, which reflects the action taken by Agent 2. Thus, Agent 1 does not have to curtail its solar output.

6 Conclusion

This paper proposed a blockchain-based TES to allow disparate agents to provide voltage regulation services to the grid in return for economic compensation. The implementation of these services were facilitated by the implementation of a smart contract that impartially audits service contract between agents and enforces its validity by directly observing power measurements on the ledger provided by smart meters throughout the ADN. The solution proposed a reputation based rating system that was increased and decreased based on the successful/unsuccessful execution of service contracts between agents, and significantly affected the agents ability to generate revenue in future transactions. The efficacy of the system was tested on both a large, simulated model of an ADN as well as a real-world Canadian microgrid using HLF, where experimental results showed the ability for distributed agents to resolve voltage violations.

7 References

[1] Tushar, W., Zhang, J.A., Smith, D.B., et al.: ‘Prioritizing consumers in smart grid: A game theoretic approach’, *IEEE Trans. Smart Grid*, 2014, 5, (3), pp. 1429–1438.

[2] Pirbazari, A.M.: ‘Ancillary services definitions, markets and practices in the world’. 2010 IEEE-PES Transmission and Distribution Conf. and Exposition: Latin America (T&D-LA), Sao Paulo, Brazil, November 2010, pp. 32–36.

[3] Mahmud, N., Zahedi, A.: ‘Review of control strategies for voltage regulation of the smart distribution network with high penetration of renewable distributed generation’, *Renew. Sust. Energy Rev.*, 2016, 64, pp. 582–595.

[4] Renani, Y.K., Ehsan, M., Shahidehpour, M.: ‘Optimal transaction market operations with distribution system operators’, *IEEE Trans. Smart Grid*, 2018, 9, (6), pp. 6692–6701.

[5] Mohy-ud-din, G., Muttagi, K.M., Sutanto, D.: ‘Transaction energy-based planning framework for vpps in a co-optimised day-ahead and real-time energy market with ancillary services’, *IET Gener., Transf. Distrib.*, 2019, 13, (11), pp. 2024–2035.

[6] Zhou, X., D’All’Anese, E., Chen, L., et al.: ‘An incentive-based online optimization framework for distribution grids’, *IEEE Trans. Autom. Control*, 2018, 63, (7), pp. 2019–2033.

[7] Kekatos, V., Wang, G., Conejo, A.J., et al.: ‘Stochastic reactive power management in microgrids with renewables’, *IEEE Trans. Power Syst.*, 2015, 30, (6), pp. 3386–3395.

[8] Shigenobu, R., Kinjo, M., Mandal, P., et al.: ‘Optimal operation method for distribution systems considering distributed generators imparted with reactive power incentive’, *Appl. Sci.*, 2018, 8, (8), pp. 1–23. Available: http://www.mdpi.com/2076-3417/8/1/1141.

[9] Mylrea, M., Gourtsetti, S.N.G.: ‘Blockchain for smart grid resilience: exchanging distributed energy at speed, scale and security’, *2017 ResilienceWeek (RWS)*, Wilmington, DE, USA, September 2017, pp. 18–23.

[10] Andoni, M., Robu, V., Flynn, D., et al.: ‘Blockchain technology in the energy sector: a systematic review of challenges and opportunities’, *Renew. Sust. Energy Rev.*, 2019, 100, pp. 143–174. Available: http://www.sciencedirect.com/science/article/pii/S096642401831858X.

[11] Mengelkamp, E., Garttner, J., Rock, K.: ‘Consortium blockchain for secure energy trading in industrial internet of things’, *IEEE Trans. Ind. Inf.*, 2018, 14, (8), pp. 3690–3700.

[12] Kang, J., Yu, R., Huang, X., et al.: ‘Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains’, *IEEE Trans. Ind. Inf.*, 2017, 13, (6), pp. 3154–3164.

[13] Aitken, N.Z., Svetinovic, D.: ‘Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams’, *IEEE Trans. Dependable Secur. Comput.*, 2018, 15, (5), pp. 840–852.

[14] Hatziargyriou, N., Asano, H., Irawan, R., et al.: ‘Microgrids’, *IEEE Power Energy Mag.*, 2007, 5, (4), pp. 78–94.
[16] Danzi, P., Angelichinoski, M., Stefanovic, , et al.: ‘Distributed proportional-fairness control in microgrids via blockchain smart contracts’. 2017 IEEE Int. Conf. on Smart Grid Communications (SmartGridComm), Dresden, Germany, October 2017, pp. 45–51

[17] Wood, D.D.: ‘Ethereum: a secure decentralised generalised transaction ledger’ (Ethereum Project Yellow Paper, 2014)

[18] Bach, L. M., Mihaljevic, B., Zagar, M.: ‘Comparative analysis of blockchain consensus algorithms’. 2018 41st Int. Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 2018, pp. 1545–1550

[19] Jalalzai, M. M., Busch, C.: ‘Window based bft blockchain consensus’. 2018 IEEE Int. Conf. on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, Canada, July 2018, pp. 971–979

[20] Chan, W., Olmsped, A.: ‘Ethereum transaction graph analysis’. 2017 12th Int. Conf. for Internet Technology and Secured Transactions (ICITST), Cambridge, UK, December 2017, pp. 498–500

[21] Manshaei, M. H., Jadliwala, M., Maiti, A., et al.: ‘A game-theoretic analysis of shard-based permissionless blockchains’, IEEE Access, 2018, 6, pp. 78 100–78 112

[22] O’Dwyer, K. J., Malone, D.: ‘Bitcoin mining and its energy footprint’. 25th IET Irish Signals Systems Conf. 2014 and 2014 China-Ireland Int. Conf. on Information and Communications Technologies (ISSC 2014/CIICT 2014), Limerick, Ireland, June 2014, pp. 280–285

[23] Cash, M., Bassioni, M.: ‘Two-tier permission-ed and permission-less blockchain for secure data sharing’. 2018 IEEE Int. Conf. on Smart Cloud (SmartCloud), New York, NY, USA, September 2018, pp. 138–144

[24] Thakkar, P., Nathan, S., Viswanathan, B.: ‘Performance benchmarking and optimizing hyperledger fabric blockchain platform’. 2018 IEEE 26th Int. Symp. on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Milwaukee, WI, USA, 2018, pp. 264–276

[25] Sidirov, M., Tze Ong, M., Vikneswaran, R., et al.: ‘Ultralightweight mutual authentication rfid protocol for blockchain enabled supply chains’, IEEE Access, 2019, PP, pp. 1–1

[26] Cheng, C., Xue, S., Dai, H., et al.: ‘An overview of blockchain technology: architecture, consensus, and future trends’. 2017 IEEE Int. Congress on Big Data (BigData Congress), Honolulu, HI, USA, June 2017, pp. 557–564

[27] Patel, S. T., Mistry, N.H.: ‘A review: sybil attack detection techniques in wsn’. 2017 4th Int. Conf. on Electronics and Communication Systems (ICECS), Coimbatore, India, February 2017, pp. 184–188

[28] Dinh, T.T.A., Wang, J., Chen, G., et al.: ‘Blockbench: A framework for analyzing private blockchains’. Proc. of the 2017 ACM Int. Conf. on Management of Data, Chicago, IL, USA, 2017, pp. 1085–1100

[29] Nasir, Q., Qasse, I., Talib, M., et al.: ‘Performance analysis of hyperledger fabric platforms’, Sec. Commun. Netw., 2018, 2018, pp. 1–14

[30] El-Taweel, N. A., Farag, H. E.: ‘Voltage regulation in islanded microgrids using distributed constraint satisfaction’, IEEE Trans. Smart Grid, 2018, 9, (3), pp. 1613–1625

[31] Abdelaziz, M.M.A., Farag, H.E., El-Saadany, E.F., et al.: ‘A globally convergent trust-region method for power flow studies in active distribution systems’. 2012 IEEE Power and Energy Society General Meeting, San Diego, CA, USA, July 2012, pp. 1–7

[32] Elkhatib, M. E., El-Shatshat, R., Salama, M.M.A.: ‘Novel coordinated voltage control for smart distribution networks with dg’, IEEE Trans. Smart Grid, 2011, 2, (4), pp. 598–605

[33] Münsing, E., Mather, J., Moura, S.: ‘Blockchains for decentralized optimization of energy resources in microgrid networks’. 2017 IEEE Conf. on Control Technology and Applications (CCTA), Mauna Lani, HI, USA, August 2017, pp. 2164–2171

[34] Cai, C., Duan, H., Wang, C.: ‘Tutorial: building secure and trustworthy blockchain applications’. 2018 IEEE Cybersecurity Development (SecDev), Cambridge, MA, USA, September 2018, pp. 120–121

[35] Liang, G., Weller, S.R., Luo, F., et al.: ‘Distributed blockchain-based data protection framework for modern power systems against cyber attacks’, IEEE Trans. Smart Grid, 2019, 10, (3), pp. 3162–3173

[36] Li, C., Palanisamy, B., Xu, R.: ‘Scalable and privacy-preserving design of on/off-chain smart contracts’, CoRR, abs/1902.06359, 2019. Available: http://arxiv.org/abs/1902.06359

[37] Xygkis, T.C., Korres, G.N.: ‘Optimal allocation of smart metering systems for enhanced distribution system state estimation’. 2016 Power Systems Computation Conf. (PSCC), Genoa, Italy, June 2016, pp. 1–7

[38] Smith, R.G., Davis, R.: ‘Frameworks for cooperation in distributed problem solving’, IEEE Trans. Syst., Man, and Cybern., 1981, 11, (1), pp. 61–70

[39] Kirby, B.: ‘Ancillary services: technical and commercial insights’ (Wärtsilä North America, USA, 2019)