

# Real-time Peer to Peer Energy Trade with Blockchain Offline Channels\*

Subhasis Thakur

National University of Ireland, Galway  
Galway, Ireland  
subhasis.thakur@nuigalway.ie

John G. Breslin

National University of Ireland, Galway  
Galway, Ireland  
john.breslin@nuigalway.ie

**Abstract**—Blockchain become a suitable platform for peer to peer energy trade as it facilitates secure interactions among parties with trust or a mutual trusted 3rd party. However, the scalability issue of blockchains is a problem for real-time energy trade to be completed within a small time duration. In this paper, we use offline channels for blockchains to circumvent scalability problems of blockchains for peer to peer energy trade with small trade duration. We develop algorithms to find stable coalitions for energy trade using blockchain offline channels. We prove that our solution is secure against adversarial prosumer behaviors, it supports real-time trade as the algorithm is guaranteed to find and record stable coalitions before a fixed time, and the coalition structure generated by the algorithm is efficient.

**Index Terms**—Peer to Peer Energy trade, Blockchains, Offline Channels

## I. INTRODUCTION

Blockchains are a suitable platform for peer to peer energy by eliminating 3rd parties to facilitate interactions among parties who do not trust each other. However, blockchains have scalability problems, i.e., the number of transactions processed per second is low. For example, Bitcoin processes 7 transactions per second while Mastercard processes 50000 transactions per second. We say a transaction is confirmed if the transaction is in a block that proceeds a few blocks. Figure 1 (a) explains the time it takes to process a transaction. There are three stages of transaction confirmation. First, it should be added to a new block, then the block should be accepted by the blockchain network and finally, the few new blocks should be published after the block containing the transaction in order to avoid forks. In the Bitcoin network, it takes approximately 10 minutes to publish a new block, hence it takes 40 minutes to confirm a transaction. We will use the term trade cycle to denote the sequence of events starting from the prosumers announcing their energy requirements, coalition computation, actual energy trade, and settlement of funds for the trade. Let the blockchain network acting as the platform for peer to peer energy trade include prosumers and

This project is funded by the Department of Business, Enterprise and Innovation, under the Government of Ireland's Project 2040 Plan and this publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289\_P2, co-funded by the European Regional Development Fund.

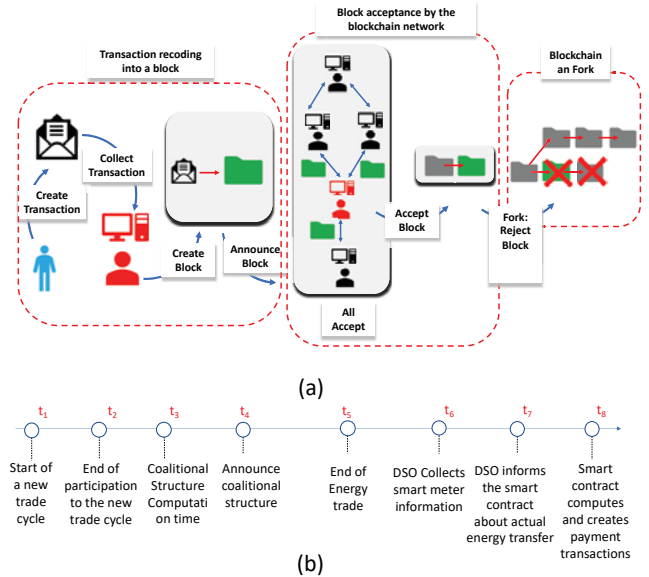


Fig. 1. (a) Transaction confirmation time. (b) Time line of events in a single trade cycle.

regulators (DSOs) as peers use smart contracts to facilitate energy trade. The total time needed for a trade cycle (shown in Figure 1 (b)) is as follows: [ $t_2$ :] Smart contract collects energy surplus or deficiency information from each prosumer. Each prosumer sends a transaction to the smart contract indicating its energy requirement. [ $t_3$ :] The smart contract computes the coalitional structure. [ $t_4$ :] The smart contract announces the coalitional structure by creating transactions. [ $t_5$ :] Prosumers trade energy as prescribed by the smart contract for a fixed duration. [ $t_6$ :] DSO collects meter reading from smart meters. [ $t_7$ :] DSO informs the smart contract about actual energy flow by creating a transaction with the smart contract as the recipient. [ $t_8$ :] The smart contract computes actual payment settlement and makes payment transactions.

The requirements of a real-time trade using blockchains are as follows: (1) A prosumer who wants to participate in the trade should be able to create a transaction within the time window [ $t_2, t_1$ ]. (2) The smart contract should be able to find

a coalitional structure by  $t_3$ . (3) The smart contract should be able to create transactions before  $t_4$  to announce the coalitional structure. (4) The DSO should be able to gather smart meter information within time  $t_6 - t_5$ . (5) The DSO should be able to inform the smart contract by creating a transaction before time  $t_7$ . (6) The smart contracts should be able to create transactions before  $t_8$ . In presence of the above time constraints to create and confirm transactions, it is likely that : (1) It may be possible that a transaction does not get a confirmed with a time limit and it will lead to exclusion of prosumer from the trade or the trade does not commence at all. (2) It may be possible that a transaction is initially accepted but later, it is discarded due to forks. In such a scenario, the trade will occur with discarded information or trade decisions.

In this paper, we propose a blockchain offline channel-based solution to the transaction confirmation problem. Our solution guarantees that information is recorded into the blockchain in real-time, i.e., transactions are recorded within a fixed time duration. Only time delay in receiving messages impacts the time to transfer tokens through offline channels. In a bi-directional channel, only 3 messages are needed to transfer tokens between two peers. We have the following results:

(1) We developed a Contractual Nash stable model of co-operative peer to peer energy trade. (2) We developed distributed algorithms using blockchain offline channels to find contractual Nash stable coalitions for energy trade. (3) We prove that the proposed trade algorithm is secure against adversarial prosumers who provide false information. (4) We prove that the proposed algorithm guarantees that information will be recorded in the blockchain. (5) Using experimental evaluation we analyse the efficiency of the proposed energy trade algorithm.

The paper is organised as follows: in section 2 we review related literature, in section 3 we describe contractual Nash stable coalitions for peer to peer energy trade, in section 4 we discuss an algorithm to find contractual Nash stable coalitions, in section 5 we prove that the proposed solution satisfies real-time energy trade requirement, in section 6 we present an experimental evaluation of the coalition formation algorithm, and we conclude the paper in section 7.

## II. RELATED LITERATURE

The Bitcoin lightning network was proposed in [1] which allows peers to create and transfer funds among them without frequently updating the blockchain. Similar networks are proposed for Ethereum [2] and credit networks [3]. Blockchain is a suitable platform for peer to peer energy trade. In [4] authors have analysed the suitability of blockchain network in terms of network size, communication delay, etc on recording transactions for the energy trade. In [5] authors have used coalitional game theory in peer to peer energy trade which also includes electric vehicles. In [6], [8] the authors used coalitional game theory to model blockchain-based energy trade. In [7] the authors used double auction for peer to peer energy trade. We advance the state of art in peer to peer

energy trade with blockchains by providing a real-time trading platform using blockchain offline channels.

## III. COALITIONAL PEER TO PEER ENERGY TRADE

$\{p_i\}$	$n$ prosumers
$\{t_i\}$	discrete time instances
$D_i^j$ and $S_i^j$	energy demand and energy supply (through its own energy generators, i.e., solar panels) of the prosumer $p_i$ at time $t_j$ until time $t_{j+1}$ or for time duration $dt$ . The energy requirement at $p_i$ at time $t_j$ is $E_i^j = D_i^j - S_i^j$ . A positive value of $E_i^j$ means prosumer $p_i$ has surplus energy (i.e., it is generating more than its own consumption) and a negative value of $E_i^j$ will mean the prosumer has an energy deficiency for next $dt$ time duration.
$d^{i,j}$	the distance between prosumer $p_i$ and $p_j$ w.r.t the distribution lines.

**Definition 1.** A coalitional structure over the prosumers at the time  $t_j$  is a non-overlapping partition over them and it is denoted as  $\pi^j = (\pi_1^j, \dots, \pi_k^j)$  where  $k$  is a positive integer and  $\pi_x^j \subset P$ . Each group  $\pi_x^j \in \pi^j$  is a coalition for the time duration  $dt$  at time  $t_j$ . Prosumers trade energy among themselves in each coalition.

We assume that coalitions are formed among prosumers in close proximity and we will use  $d$  as the maximum allowed distance between any two prosumer in a coalition. In each coalition, the total generation is shared among the prosumers with energy deficiency proportional to their energy demand. For the coalition  $\pi_i^j = (p_1, p_2, \dots, p_x)$ , total energy generation is  $\sum_{p_i \in \pi_i^j: E_i^j > 0} E_i^j$ . For any prosumer  $p_x$  with energy deficiency in the coalition  $\pi_i^j$  will get a share of

$$\theta_x^j = \begin{cases} \frac{(\sum_{p_i \in \pi_i^j: E_i^j > 0} E_i^j) \times E_x^j}{\sum_{p_i \in \pi_i^j: E_i^j < 0} E_i^j} & \text{if } \sum_{p_i \in \pi_i^j: E_i^j > 0} E_i^j < \sum_{p_i \in \pi_i^j: E_i^j < 0} E_i^j \\ E_x^j & \text{if } \sum_{p_i \in \pi_i^j: E_i^j > 0} E_i^j \geq \sum_{p_i \in \pi_i^j: E_i^j < 0} E_i^j \end{cases}$$

Similarly, for any prosumer  $p_x$  with energy surplus in the coalition  $\pi_i^j$  will be able to sell a fraction of its surplus

$$\delta_x^j = \begin{cases} \frac{(\sum_{p_i \in \pi_i^j: E_i^j < 0} E_i^j) \times E_x^j}{\sum_{p_i \in \pi_i^j: E_i^j > 0} E_i^j} & \text{if } \sum_{p_i \in \pi_i^j: E_i^j \geq 0} E_i^j \geq \sum_{p_i \in \pi_i^j: E_i^j < 0} E_i^j \\ E_x^j & \text{if } \sum_{p_i \in \pi_i^j: E_i^j < 0} E_i^j < \sum_{p_i \in \pi_i^j: E_i^j > 0} E_i^j \end{cases}$$

We assume that  $q^j \geq q_i^j$  where  $q^j$  and  $q_i^j$  are per unit cost of electricity from the grid at time  $t_j$  and from any prosumer with surplus energy in the coalition  $\pi_i^j$ . Using these parameters, we define the utility of prosumers as follows:

**Definition 2.** The utility of a peer  $p_x$  in the coalition  $\pi_i^j$  at time  $t_j$  is:

$$U_x^j(\pi_i^j) = \begin{cases} \theta_x^j \times q_i^j + q^j * \text{Max}(0, E_x^j - \theta_x^j) & \text{if } E_x^j < 0 \\ \delta_x^j \times q_i^j & \text{if } E_x^j > 0 \end{cases}$$

**Definition 3.** A coalitional structure  $\pi^j = (\pi_1^j, \dots, \pi_k^j)$  is contractually Nash stable if the following holds: For any  $p_x \in \pi_i^j$  there is no coalition  $\pi_y^j$  such that: (a)  $U_x^j(\pi_y^j) > U_x^j(\pi_i^j)$ , (b) Maximum distance from  $p_x$  to any member of  $\pi_y^j$  is less than  $d$ , (c) and, for all  $p_z \in \pi_y^j$ ,  $U_z^j(\pi_y^j \cup p_x) \geq U_z^j(\pi_y^j)$ , i.e., no existing member of the new coalition is not worse off as  $p_x$  joins the coalition.

#### IV. COOPERATIVE ENERGY TRADE WITH BLOCKCHAIN OFFLINE CHANNELS

First, we will present an informal description of how to use offline channels to compute a coalition structure, next we will explain the distributed algorithm to compute such a coalition structure, next we will discuss the adversarial behaviour of prosumers, next we will explain channel-based algorithm to compute coalitional structure.

##### A. Brief description of the solution

Briefly, our solution is as follows: *Step 1:* Prosumers follow a protocol to find coalitions by themselves as they follow a distributed algorithm. We define a coalition as one seller and multiple buyers. *Step 2:* After forming the coalitions, the prosumers record such a coalition using Hashed Time Locked Contracts (HTLCs)<sup>1</sup> in the channel network. *Step 3:* After forming the coalitions, prosumers trade energy. After the completion of electricity trade, buyers verify the meter reading of sellers before paying the sellers for electricity. PBTs used to transfer a token from each buyer to DSO. Successful PBT execution indicates that actual electricity flowed from the sellers to the grid. After completion of PBT executions, HTLCs are updated to record the transfer of funds from buyers to the sellers. First, in section 4.3 we discuss a distributed protocol for finding a coalitional structure. In section 4.5 we discuss HTLC creation to record coalitional structures into the channel network and in section 4.6 we discuss the execution of PBTs from buyers to the DSO and procedure to update the HTLCs to record the payment in the channel network.

##### B. Protocol for distributed computation of coalitional structure

The protocol for distributed computation is as follows: We classify the prosumers into the sets of buyers and sellers.

- (1) A buyer randomly selects another prosumer within distance  $d$  and checks if the prosumer is a seller. Assume that the buyer is  $Buyer_1$  and the seller is  $Seller_1$ .
- (2)  $Seller_1$  informs the available electricity it can sell to  $Buyer_1$ .
- (3) If  $Buyer_1$  agrees to buy the proposed amount of energy then it agrees with  $Seller_1$ .
- (4) If  $Seller_1$  successfully receives the message from  $Buyer_1$  and chooses it to sell its electricity then it sends a confirmation message to  $Buyer_1$ .
- (5) A buyer can be part of multiple coalitions. As shown in the example,  $Buyer_1$  also participates in another coalition where  $Seller_n$  is the seller.

- (6)  $Seller_n$  updates previous members of its coalition about joining of  $Buyer_1$  in its coalition.

In each coalition, there is only one seller and possibly more than one buyer. And, in each coalition, the seller only includes new buyers in its coalition if it has excess electricity after satisfying the energy deficiency of all buyers in its coalition. We assume that all communications between the prosumers are secure and finish within a finite delay.

##### C. Adversary models

We will assume that communications are secure among the prosumers. We assume that communication channels are not faulty and it takes a finite time to receive a message. We consider the following adversarial behaviours from the prosumers: *Fake Buyers:* Buyers can provide false information regarding their energy needs. A fake buyer may agree to buy a certain amount of energy but does not consume energy. *Fake Seller:* Sellers can provide false information regarding their surplus energy. A fake seller may agree to sell more electricity than it can possibly produce. *Faulty Buyer:* A buyer may become faulty and does not consume electricity despite agreeing with a least one buyer to buy its electricity. *Faulty Seller:* A seller may become faulty and does not produce electricity despite agreeing with a least one seller to sell its electricity.

##### D. HTLCs for recording coalition structure

The distributed coalitional structure finding protocol creates each coalition with one seller and multiple buyers. The first buyer is offered to buy all electricity provided by the seller. It may be more than the first buyer's requirement. Hence the second buyer buys the surplus electricity from the first buyer. This process is continued until the total demand from the buyers is more than the total electricity to be generated by the seller or no other buyer joins the coalition. We use HTLCs to record this hierarchy of buyers in a coalition. For example, if the coalition contains two buyers, where the first buyer buys electricity of value 10 tokens while it only needs electricity of value 7 tokens then the second buyer buys electricity of 3 tokens from the first buyer. It is needed that all prosumers in a coalition are within distance  $d$  from each other w.r.t electricity distribution lines. We assume that electricity flows from any seller to all buyers within distance  $d$ . We will use two HTLCs to denote the coalition where  $Buyer_1$  will pay  $Seller_1$  10 tokens and  $Buyer_2$  will pay  $Buyer_1$  3 tokens. Assume that there are offline channels between  $Buyer_1$  and  $Seller_1$  (and between  $Buyer_1$  and  $Buyer_2$ ). In such channels new confirmation transactions are created which pays  $Seller_1$  10 tokens from  $Buyer_1$  and pays 3 tokens from  $Buyer_2$  to  $Buyer_1$ . The process of creating a confirmation transaction is described in section 4.1. Next, we use PBT (as described in section 4.1) to transfer tokens from the DSO to each buyer as each buyer verifies whether the seller has actually produced the promised electricity. Successful execution of such PBT reveals certain keys to the seller which it can use to update its HTLCs with the  $Buyer_1$  so that it can get paid for the

<sup>1</sup>[https://en.bitcoin.it/wiki/Hash\\_Time\\_Locked\\_Contracts](https://en.bitcoin.it/wiki/Hash_Time_Locked_Contracts)

electricity whenever it publishes the HTLC to the blockchain network.

We assume that prosumers and the DSOs are peers of a blockchain network. The blockchain is assumed to a proof-of-work-based public blockchain (Bitcoin network). The blockchain offline channels are established according to the channel establishment rules described in [1]. Note that the proof-of-work-based consensus can be replaced by energy-efficient consensus protocols such as proof of authority if it allows the creation of multi-signature addresses in its blockchain. Prosumers are advised to open channels with their immediate neighbouring prosumers as it is most likely to trade energy with them. Each prosumer is also required to open a channel with the DSO. It is assumed that only one such channel between a prosumer and the DSO can be opened. All channels are assumed to be bi-directional and value in the channels represented by a cryptocurrency. First, we explain the HTLC creation in a coalition among a buyer and a seller.

(1) First,  $Seller_1$  asks the DSO to provide a set of locks (Hash of random string with a fixed Hashing function). Let  $Seller_1$  asks for  $k$  locks. These locks are shown using Black color in Figure 2(a).

(2) Let the total value of  $Seller_1$ 's excess energy is 4 tokens.  $Buyer_1$  agrees to pay  $Seller_1$  4 tokens as it creates confirmation transactions, i.e., HTLCs where it allocates 4 additional tokens to  $Seller_1$ . The conditions of the HTLCs are as follows: In the confirmation transaction created by  $Buyer_1$ :  
 (a)  $Buyer_1$  will immediately get 1 token from the channel between  $Buyer_1$  and  $Seller_1$ .  
 (b) After 13 Minutes (we assume that the trade window is 5 Minutes)  $Seller_1$  will get 9 tokens.  
 (c) But if  $Seller_1$  publishes this transaction despite being an old confirmation transaction then  $Buyer_1$  will get these 9 tokens if it can reveal key of  $Seller_1$  and  $k$  keys of the locks provided by the DSO.

(3) After the creation of this confirmation transactions, both parties exchange it as they now have a contract for energy trade.

(4) Next,  $Buyer_1$  creates a sequence of HTLCs to send a very small amount of tokens to the DSO. Locks of these sequence of HTLCs are the  $k$  locks provided by the DSO.

(5) The minimum time limit of these HTLCs is slightly more than the expected termination time of the energy trade, i.e., we assumed the energy trade window is 5 Minutes, hence we use the minimum time window 10 minutes. This means after  $Seller_1$  trades electricity to  $Buyer_1$  for the next 5 minutes, the contract can be executed by the DSO.

The protocol for extending a coalition is as follows (shown in Figure 2(b)): We show how to extend the coalition shown in Figure 2(a) with an additional buyer  $Buyer_2$ . It is as follows:

(1) Assume that,  $Buyer_2$  to be included in the previous coalition with  $Buyer_1$  and  $Seller_1$ . Assume that, remaining electricity from  $Seller_1$  after satisfying  $Buyer_1$  is of value 2 tokens. This means  $Buyer_2$  joins this coalition to buy energy

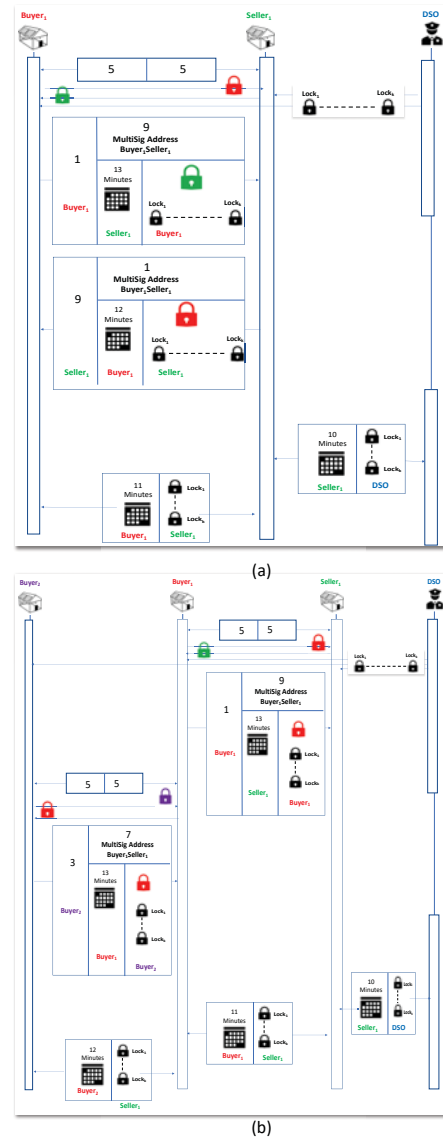


Fig. 2. (a) Protocol for coalition between a seller and first buyer, (b) Protocol for path based transfer

of value 2 tokens. Note that such augmentation of this coalition is Contractual Nash stable as  $Buyer_1$  does not get worse off by including  $Buyer_2$  in the coalition.

(2) Similar to previous methods of coalition formation, confirmation transactions are created and shared between  $Buyer_2$  and  $Buyer_1$ . The confirmation transaction created by  $Buyer_2$  is as follows:

(a) According to it if  $Buyer_1$  publishes it then  $Buyer_2$  will get 2 tokens and  $Buyer_1$  need to wait for another 13 minutes to get 7 tokens from this channel between  $Buyer_1$  and  $Buyer_2$ .  
 (b) But  $Buyer_2$  can claim these 7 tokens if it can produce the key to the lock created by  $Buyer_1$ .

(3) Next,  $Buyer_2$  creates a sequence of HTLCs to send a very small amount of tokens to the DSO via  $Buyer_1$  and  $Seller_1$ . Locks of these sequences of HTLCs are the  $k$  locks provided

by the DSO.

(4) The minimum time limit of these HTLCs is slightly more than the expected termination time of the energy trade, i.e., we assumed the energy trade window is 5 Minutes, hence we use the minimum time window 10 minutes. This means after  $Seller_1$  trades electricity to  $Buyer_1$  and  $Buyer_2$  for the next 5 minutes, the contract can be executed by the DSO.

It is possible that another Buyer can be added to this coalition shown in Figure 2(b). We can create another set of confirmation transactions between  $Buyer_1$  and  $Buyer_3$ . We can partially use the locks provided by the DSO.

### E. HTLC Executions

We will explain execution of the HTLCs as created in the previous section with the following example. As shown in Figure 3(a) three sets of confirmation transactions are created between  $(Seller_1, Buyer_1)$ ,  $(Buyer_1, Buyer_2)$  and  $(Buyer_1, Buyer_3)$ . Also, three sets of PBTs are initiated as shown in Figure 3(b) as follows:

- (1) The PBT path  $Buyer_2 \rightarrow Buyer_1 \rightarrow Seller_1 \rightarrow DSO$  uses the locks  $Lock_1, \dots, Lock_x$ .
- (2) The PBT path  $Buyer_3 \rightarrow Buyer_1 \rightarrow Seller_1 \rightarrow DSO$  uses the locks  $Lock_{x+1}, \dots, Lock_y$ .
- (3) The PBT path  $Buyer_1 \rightarrow Seller_1 \rightarrow DSO$  uses the locks  $Lock_{y+1}, \dots, Lock_k$ .

Note that these sequences of HTLCs are executed after the actual energy trade occurs, i.e., electricity flows from the seller to the buyers. These contracts are used to verify if actual electricity flow occurred as measured by the DSO using smart meters at the prosumers. If the DSO measures expected electricity flow then it initiates execution of such sequences of HTLCs by providing keys of the  $Lock_1, \dots, Lock_k$ . As the  $Seller_1$  and  $Buyer_1$  are included in all such PBTs, they notice the keys used by the DSO in executing these HTLCs. These keys will be used by them to claim funds for electricity transfer. Note that, the objective of channels is to keep updated confirmation transactions rather than publishing confirmation transactions as peers can reuse the channel with updated confirmation transactions. In order to update a channel keys of the last confirmation transaction must be revealed. Hence if an old confirmation transaction is published then the publishing party does get the tokens immediately as it has to wait for the time mentioned in its HTLC. While it is waiting, the other party of the HTLC can observe such transactions in the blockchain network and claim all tokens as it knows the keys to the locks created by the DSO and keys to the locks shared by the alternate parties.

## V. ANALYSIS

### A. Security

**Fake Buyers:** A fake buyer may agree to buy a certain amount of energy but does not consume energy. But it enters an HTLC with the seller and irrespective of whether it consumes electricity for the trading window or not, it will be

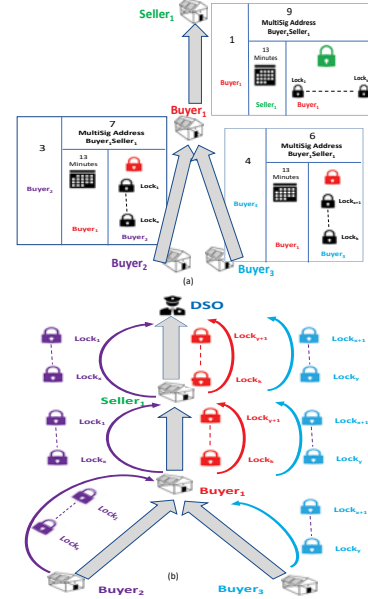


Fig. 3. (a) Set of confirmation transactions with HTLCs to be updated with keys from PBT executed from buyers to the DSO. (b) PBT protocol for token transfer from a buyer to DSO to verify the actual energy production of the seller

charged for the electricity as agreed in the HTLC if the DSO can confirm that the seller produced the required electricity.

**Fake Seller:** A fake seller may agree to sell more electricity than it can possibly produce. We assumed that DSO only operates one channel with each prosumer. In order to sell electricity to more than it can produce, a seller must use more than one set of PBTs with the first buyers in multiple trades in the same trading window. It is not possible to create multiple PBTs as the channel with the DSO is locked by one trade. Hence the seller can not create multiple PBTs.

**Faulty Buyer:** A buyer may become faulty and does not consume electricity despite agreeing with a least one buyer to buy its electricity. The seller does not get worse off as HTLCs are already created and it is executed irrespective of faults in the electricity distribution lines.

**Faulty Seller:** A seller may become faulty and does not produce electricity despite agreeing with a least one seller to sell its electricity. In this case, the DSO notices that the seller failed to produce the promised electricity and it does not execute PBTs. Hence the seller can not claim tokens for energy trade.

### B. Contractual Nash Stability

The proposed model of the coalition formation process creates coalitions with only one seller and buyer(s). Buyers are sequentially added in a coalition only if there is excess electricity after satisfying the energy deficiency of all current buyers in the coalition. We assume that the price of electricity bought from a prosumer is fixed and hence a prosumer will not leave the current coalition due to the lower cost of electricity.

### C. Real time execution

We claim that the proposed trade model fits real-time trade requirement because (illustrated in Figure 1 (b)):

(1) A prosumer needs 4 messages to find a coalition. After finding a coalition, it needs 4 messages to create and share confirmation transactions to record the coalition in the offline channels. We assume the prosumers are situated nearby and these messages will be received before time  $t_4$ .

(2) We will assume that a seller does not evaluate participation requests from a buyer after time  $t_2$  hence, a buyer will either join or not find a coalition before trade time begins at  $t_4$ .

(3) Note that, as a seller is not allowed to evaluate participation offer from a buyer after  $t_2$  its coalition may become inefficient, i.e., there will be unsold electricity in a coalition. But this ensures that at least the coalition computed before  $t_2$  is recorded with HTLCs in the channel network.

(4) The blockchain network will not control the information gathering infrastructure of the smart grid and hence communication delay from the smart meter to the DSO is beyond the control and scope of this solution.

(5) A DSO can verify whether the seller has produced the promised electricity by executing HTLCs in the PBTs from each buyer to the DSO. Each such PBT needs  $x$  messages to be received for successful execution where  $x$  is the length of the PBT, i.e., the number of buyers +2 (the seller and the DSO). We expect such a finite small number of messages will be received by the prosumers before the deadline  $t_7$ .

(6) Finally, the buyers and the sellers update their HTLCs with the keys received from PBT execution from buyers to the DSO. Such an update is instantaneous as it only needs to store the key in its internal database.

### VI. EVALUATION

We evaluate the performance of the coalition structure generation algorithm in terms of communication of finding coalitions and length of PBTs executed in each coalition as the length of the PBT (number of peers in a PBT execution path) indicates the number of messages needed to execute the energy trade operation. We use agent-based modelling [4] and Simpy library in Python to produce an asynchronous event simulation of the distributed coalition formation protocol described in section IV-D. We use 6 datasets of 50, 100, 150, 200, 250, 300 prosumers. In each dataset, 50% prosumers are identified as sellers and rest as buyers chosen uniformly at random. Energy deficiency and the surplus of each prosumer is a fraction between [0,1]. Distance between each pair of prosumers is a fraction between [0,1]. We assume mean distances between all pairs of the prosumers is the maximum allowed distance between any two prosumers in a coalition. Outcomes of the execution of the coalition formation protocols are shown in Figures 4(a) and 4(b). As shown in Figure 4(a) the average size of the coalitions remains approximately the same as we increase the number of prosumers. This means the length of PBT execution path for executing PBTs from buyers to DSO (as discussed in section IV-E) remains low and hence it is unlikely that the communication failure will lead to the failure

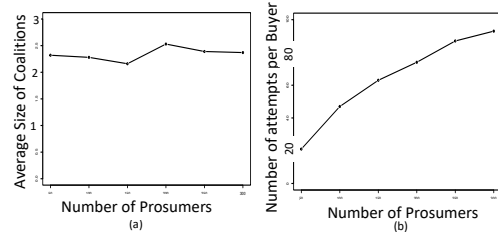


Fig. 4. (a) Average size of coalitions (b) Average number of attempts per buyer to join a coalition.

of such PBT executions. As shown in Figure 4(b), the number of attempts per buyer increases as the number of prosumers for energy trade is increased. This is expected as the number of potential coalitions to join for each buyer is increased because the number of prosumers is increased.

### VII. CONCLUSION

The long transaction confirmation time of public proof-of-work/stake based blockchains makes it impossible to use for energy trades with a small trade duration. We prove that the proposed trade model is secure against adversarial prosumers and it is efficient in terms of energy loss due to long transmission and unsold renewable energy.

### REFERENCES

- [1] J. Poon and T. Dryja, "The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments." [Online]. Available: <https://lightning.network/lightning-network-paper.pdf>
- [2] "Raiden network," <http://raiden.network/>, accessed 2018.
- [3] G. Malavolta, P. Moreno-Sanchez, A. Kate, and M. Maffei, "Silentwhispers: Enforcing security and privacy in decentralized credit networks," *IACR Cryptology ePrint Archive*, vol. 2016, p. 1054, 2016.
- [4] B. Hayes, S. Thakur, and J. Breslin, "Co-simulation of electricity distribution networks and peer to peer energy trading platforms," *International Journal of Electrical Power & Energy Systems*, vol. 115, p. 105419, 2020. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0142061519302972>
- [5] S. Thakur, "A unified model of peer to peer energy trade and electric vehicle charging using blockchains," *IET Conference Proceedings*, pp. 77 (6 pp.)–77 (6 pp.)(1). [Online]. Available: <https://digital-library.theiet.org/content/conferences/10.1049/cp.2018.1909>
- [6] S. Thakur and J. G. Breslin, "Peer to peer energy trade among microgrids using blockchain based distributed coalition formation method," in *Technol Econ Smart Grids Sustain Energy*, vol. 3, no. 4, 2018.
- [7] S. Thakur, B. P. Hayes, and J. G. Breslin, "Distributed double auction for peer to peer energy trade using blockchains," in *2018 5th International Symposium on Environment-Friendly Energies and Applications (EFEA)*, Sep. 2018, pp. 1–8.
- [8] W. Tushar, T. K. Saha, C. Yuen, M. I. Azim, T. Morstyn, H. V. Poor, D. Niyato, and R. Bean, "A coalition formation game framework for peer-to-peer energy trading," *Applied Energy*, vol. 261, p. 114436, 2020. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0306261919321245>