Peer-to-Peer Optimal Solar Energy Trading using Proof-of-Authority Blockchain Technology

Mohammad Rafaquat Alam , Teoh Jin Yang

***Abstract*—With Malaysia's growing populations and improved lifestyles, the demand for energy would only increase posing new challenges and problems to the energy sector. This project proposes a peer-to-peer web-based solar energy trading platform that is to be built utilizing Blockchain technology. The aim of this project is to optimize energy usage by homes, such that those that require higher amounts, can trade energy instead of paying higher energy bills. The back end of the platform is to be built using the Golang, while using the Angular framework for the front end. Energy forecasting and double auction mechanism is also included for optimal energy allocation and they are written in Python. Moreover, the platform utilizes the MongoDB NoSQL cloud database to securely and reliably store its data.**

***Index Terms*—Solar Energy, Energy Forecasting, Double Auction, Blockchain, Frontend and Backend**

I. Introduction

THIS demo file is intended to serve as a "starter file" for IEEE Transactions on Magnetics journal papers produced under LaTeX using IEEEtran.cls version 1.8b and later. I wish you the best of success.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

II. Methods

A. Energy Forecasting

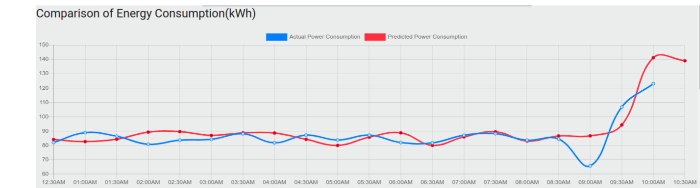


Fig:1 Prosumer specific energy consumption forecast vs time

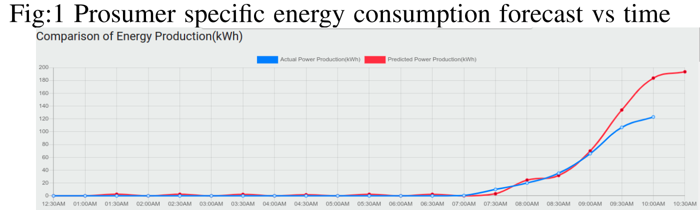


Fig:2 Prosumer specific energy production forecast vs time

1) Why Energy Forecasting: For our implemented web application, we are assuming that the solar energy a prosumer generates is stored in a battery and the energy balance is stored in a smart metre in their homes. Using this stream of data from the smart metre, we forecast the amount of energy consumed as well as produced for a given day. By predicting the energy consumption, we can limit the prosumer to a max energy value that they can make an order for. This ensures, that no prosumer orders extremely large energy amount that even the grid cannot produce. By predicting the energy production, we can limit the prosumer to a max energy value that they can use to make a bid on an energy request. This ensures, that no prosumer makes a bid that exceeds the max predicted amount they can produce.

2) How Energy Forecasting is done: The energy forecasting uses Triple Exponential Smoothing[4] with Additive Trend and Seasonality and relying on the latest two data pounts for seasonal period[5]. The training also includes Trend Damping to prevent the rise of unrealistic trends and thereby account for Seasonal Irregularity[5]. Once training is done using all the data points upto a certain time of the day a forecast is generated for the next 30 minutes from that point in time. This is the maximum energy forecast depending on what feature the prosumer is making use of. If the prosumer wants to make an order, they are shown their forecast graph for energy consumption. If the prosumer wants to make a bid on an energy request, they are shown their forecast graph for energy production.

B. Optimal Energy Allocation

1) Social Welfare Maximisation, SWM: The SWM problem is the objective function that hopes to fulfill the energy request of prosumers by maximising the energy provided by the bidders by considering the cost to generate the energy for each bidder. The amount of energy that would satisfy a prosumer based on their smart metre data and energy consumption forecast is defined by the satisfaction function in [1] as:

$$U_i(E_i^n) = w_i[\ln(n\sum_{j=1}^{J}(e_{ij}^n - e_i^{n,min}) + 1)] \tag{1}$$

where E^n is the energy receviable for prosumer that would satisfy them, w_i is the charging willingness, J is the total number of bids on the request, e_{ij}^n is the energy the prosumer can receive from the bidder as per their consumption forecast,

The above equation makes sure that the prosumer(buyer) has enough energy even if their whole demand is not satisfied as per their consumption forecast. Next we need to consider the cost the bidding prosumer has to incur when they trade the energy they produced as per the forecast. The cost function is given in [1] as :

$$L_i(S_j^n) = c_1\sum_{i=1}^{I}(s_j^n)^2 + c_2\sum_{i=1}^{I}(s_{ji}^n) \tag{2}$$

where S^n is the energy that bidder can provide, I is the request made by the prosumer, c_1 and c_2 are cost factors for the bidder, s_j^n is the energy the bidder can produce as per their forecast, s_{ji}^n is the energy the bidder wants to trade from the total produced

Now that we know how much energy would satisfy the prosumer as shown in Eq.(1) and the cost the bidder has to bear to trade energy as in Eq.(2), we can determine the objective function for SWM as:

$$SWM: \max_{E_n,S_n}\sum_{i=1}^{I}U_i(E_i^n) - \sum_{j=1}^{J}L_j(S_j^n) \tag{3}$$

$$\begin{aligned} &\textit{Subject to}: e_i^{n,min} \leq \eta\sum_{j=1}^{J}e_{ij}^n \leq e_i^{n,max}, \forall_i \in E, \\ &\sum_{i=1}^{I}s_{ji}^n \leq S_j^{n,max}\forall_j \in Z, \\ &\rho s_{ji}^n = e_{ij}^n, \forall_i \in E, \forall_j \in Z, \\ &e_{ij}^n \geq 0, \forall_i \in E, \forall_j \in Z, . \end{aligned} \tag{4}$$

2) Optimal Allocation Problem, OAP: The OAP, ensures that there is no one prosumer who can out bid everyone else, meaning all bidders are guaranteed to receive some compensation as long as bidders bid on the request. Moreover TNB, as the main electricity providers in Malaysia, are also guaranteed to receive a reward since the system is designed to use their grid lines to distribute the energy. The OAP is adapted to serve prosumer-to-prosumer energy allocation from one that uses Electric Vehicles and Service Providers[1]. It is defined in [1] as :

$$OAP: \max_{E^n,S^n}\sum_{i=1}^{I}\sum_{j=1}^{J}[b_{ij}^n \ln e_{ij}^n - p_{ji}^n s_{ji}^n] \tag{5}$$

where b_{ij}^n is the price the bidder wants from the prosumer, p_{ji}^n is the price the prosumer is willing to pay to bidder,

Both 'OAP' and 'SWP' have same subject to constraints as described in Eq. (4). So, here we carry out constraint relaxation through Lagrangian method[1] to find the local maxiumim[7]. So, energy receivable by a buyer and energy tradable by a bidder, after applying Lagrangian method is:

$$e_{ij}^n = \frac{b_{ij}^n[(n\sum_{j=1}^{J}e_{ij}^n - e_i^{n,\min}) + 1]}{nw_i} \tag{6}$$

where n is the battery charging efficiency, w_i is the charging willingness, $e_i^{n,\min}$ is the minimum energy the buyer needs, and:

$$s_{ji}^n = 2c_1 s_{ji}^n + c_2 \tag{7}$$

where, $c_1 and c_2$ are cost factors for the bidder,

3) Double Auction Mechanism:

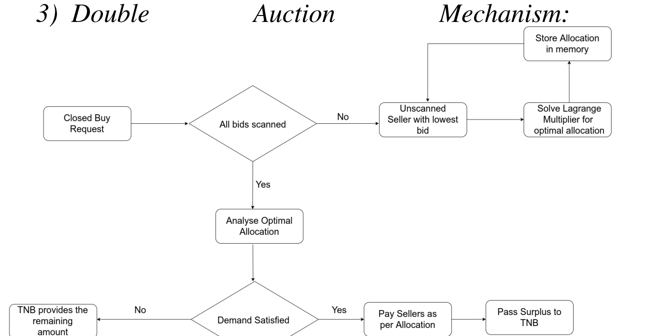
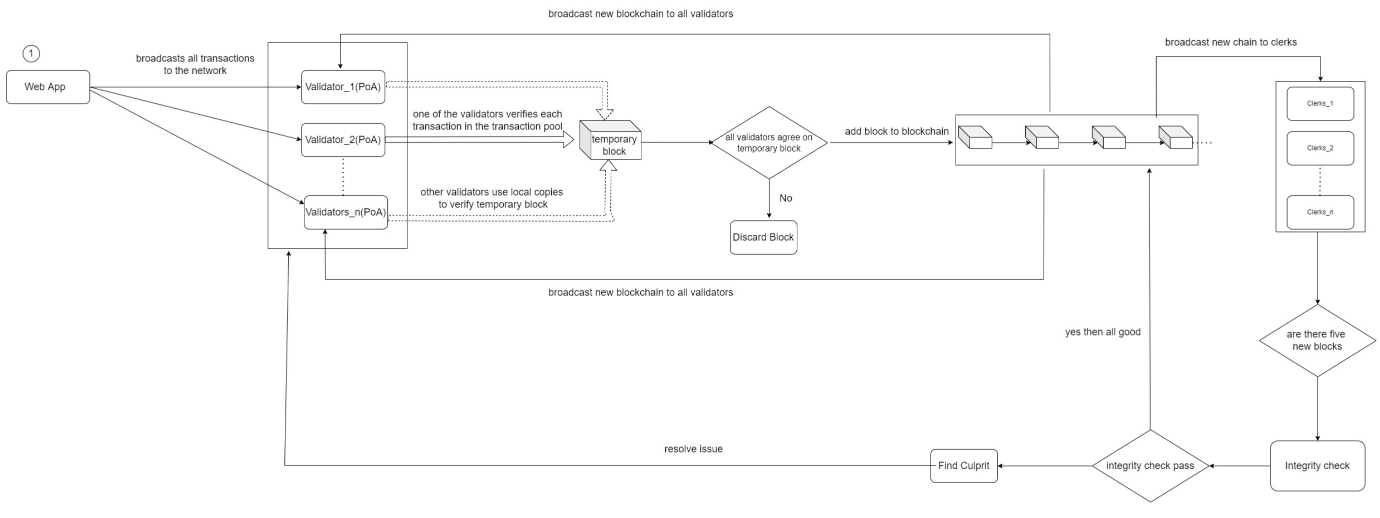


Fig:3 Flowchart for double auction with demand and supply response

The Double Auction Mechanism, DAM is what implements the OAP solution via Lagrange method. The DAM is implemented to run every 30 minutes on the web application where it takes all closed requests, loops through them. The Langrange equations, Eq.(6). and Eq.(7). are applied for each bidder for the current request in the loop. The Langrange always scales down the bidders energy amount by 10 to 15% on average to prevent one bidder to always dominate. This also means that the amount payable by prosumer to each bidder reduces. However, the prosumer still pays the full price of the energy demand they made. So, once all the bidders are considered, the system checks whether the total energy demand made by the prosumer in the request is satisfied or not. If demand is satisfied, the remaining fiat amount that is not received by an bidder is transferred to TNB. If demand is not satisfied the surplus energy is taken from TNB. The rate of energy is capped at RM 0.20 per kWH $^+_-20\%$ for all bidders as well as TNB. At the end of the DAM,:

- All bidders who have made a bid on the request receive a reward in fiat amount in exchange for their traded energy.
- The buyer pays for the energy amount they are asked for at a rate of RM 0.20.
- TNB as the grid provider can provide the surplus energy if demand not satisfied by bidders or receive the surplus fiat amount if demand is satisfied.



C. Blockchain

As shown in Fig.1, the blockchain uses a Proof of Authority(PoA) consensus mechanism where only validators are given the authority to mine new blocks and add new ones to the blockchain. We also have a new kind of users called clerks who are there for validator accountability checks. This is to ensure that validators do not conspire to work against public interest.

1) Roles in the Blockchain:

- *Validator*: A validator is one who has been granted the right to verify transactions, mine new blocks, add and discard blocks. Since we are using a Proof of Authority consensus mechanism, the validators undergo a rigorous registration process where they need to reveal their identities. Their reputation is at stake which means if they go against the interest of the normal nodes on the chain, then their status as validators will be revoked and made known to the greater community.
- *Clerk*: Clerks provide an additional layer of integrity check on the blockchain to ensure that validators do not conspire against the community. After the addition of every new block, they receive an updated local copy of the blockchain and user accounts. They will use their local copies to check whether the nonce of the last block from the central blockchain provides the same hash when they use it on the transactions in their local blockchain copy. If the match does not happen for more than 50% of the clerks, then an integrity check is triggered. Unlike validators, any normal node can be made a clerk and they do not need to be rigorously identified.

2) Blockchain pipeline:

- *Transaction Verification and Signing*: After the double auction is run every 30 minutes, the new pool of transactions are broadcasted over the network to all the validators. The validator who receives the transaction pool first, will verify each transaction where they check whether the buyer has sufficient balance or not. If so, then that transaction is marked as verified and made part of a temporary block. If a buyer does not have the required balance then the transaction is marked invalid. Once all the transactions have been checked, and added to the temporary block, it is then broadcasted to all the remaining validators. These validators use their local copy of user accounts to verify each transaction in the temporary block. They then use the nonce of the temporary block to hash the transactions from the latest block in their local blockchain. If this hash matches with that of the temporary block for all validators then the temporary one is made permanent and added to the central blockchain.
- *Discarding Blocks*: If there is a validator who does not find a match for the hash, then their local copies of user accounts and blockchain is updated. Then the check is done again. If the hash fails to match a second time, then that block is discarded. The non-match signifies that a transaction was manipulated in the central blockchain and so the block discarding is justified.
- *Integrity Check*: After the formation of 5 new blocks, an accountability check is triggered where the clerks verify each transaction in the latest permanent block in the central blockchain. They use their local copy of user accounts to verify each transaction in the latest block, then use the nonce of the latest block to hash the transactions from the latest block in their local blockchain. If the hash matches that in the central blockchain for more than 50 percent of the clerks, then there is no issue but if the match is less than 50 percent, then an integrity check is issued. This goes the through the local blockchain copy of each validator and compares the hash of the latest block. The validator(s) whose hash has a mismatch is then flagged. In this case the validator access may be revoked. This guarantees that validators do not work against public interest.

References

[1] The consortium paper
[2] The demand and supply one by Dr. Tan
[3] Proof of Authority
[4] Time-series Forecasting - Complete Tutorial 1
[5] Time-series Forecasting - Complete Tutorial2

[6] A Gentle Introduction to Exponential Smoothing for Time Series Forecasting in Python
[7] A Gentle Introduction To Method Of Lagrange Multipliers