Blockchain for peer-to-peer energy exchanges: A novel approach for the consensus algorithm

David Vangulick
University of Liège,
Liège, Belgium
david.vangulick@skynet.be

Bertrand Cornélusse, Damien Ernst
University of Liège,
Liège, Belgium
{bertrand.cornelusse, dernst}@uliege.be

***Abstract*—Energy communities and peer-to-peer energy exchanges are expected to play an important role in the energy transition. In this context, the blockchain approach can be employed to foster this decentralized energy market. The purpose of this paper is to propose a novel consensus approach in order to use blockchain as a Meter Data Management System (MDMS) and create a link between the local peer-to-peer market and the general energy market.**

***Index Terms*—Blockchain, energy community, Meter Data Management**

I. Introduction

Blockchains can be regarded as decentralized and distributed ledgers that keep track of any type of transaction. Since the arrival of Bitcoin [1] and its subsequent success as a cryptocurrency, the blockchain has emerged as a disruptive factor in many areas, starting with banking transactions. With blockchain 2.0 and the future version 3.0 allowing the use of automated transactions (e.g. smart contracts), the energy sector is probably the next sectors to be impacted by this new way of performing verification and authentication of transactions between parties.

One of the most promising use case for the blockchain in the energy sector is the peer-to-peer (P2P) energy exchange. A peer is a member of a group of local energy customers (also called Local Energy Community or LEC), including generators, consumers and prosumers that exchanges energy directly with other members of the group without intermediation by conventional energy suppliers. In western countries as European Union, USA, New Zealand or Australia but also in China, these LEC are considered as a mean to integrate more renewable energy sources (RES).It can be demonstrated that , from a financial and energy availability point of view, when generating units in the LEC are mostly RES, only a part of the energy needed by each customer can be provided by P2P energy exchange. The rest part (and in most of the case, the main part) has to be provided by the classical market.

add ref to winter package

To guarantee the rights and duties of each party and to make the necessary link to the wholesale market, these exchanges must be supervised by a neutral metering party such as the distribution system operators (DSOs) as provided for in French law [2] on collective self-consumption or in the E-Cloud project [3]).

find a ref

We will endeavor to demonstrate that a novel consensus algorithm has to be used in order to make the necessary logical and data links between P2P Energy exchange in the LEC and the classical Market. This paper is structured as follows: the first following section will set the definitions and conventions used in this paper. Section III clarifies the functionalities needed to run a P2P energy exchange and the link with the classical market. After summarizing the work done so far, Section V then states the problem of interest in this paper. Section VI, we will describe the consensus algorithm to run a blockchain ables to cope with the problem stated in Section V. Section VII concludes and provides directions of further work.

II. Definitions and conventions

For the remainder of this paper, the following definitions and conventions shall apply:

- **Local Energy Community** (LEC): We will use a generalized local energy community definition. It is defined by:
  - a limited geographical area (e.g. same street, or same residential block, same business area);
  - at least, one connection point between the community and the public grid (in an extreme case, each participant is connected to the public grid);
  - generations units that are installed in the same geographical area as the community and are considered as common asset(s) to the community (virtual power plant);
  - consumption and generation are metered separately. These meters provide data for each classical market period (e.g. 1/4h).
- **Meter Data Management System (MDMS)** is used to collect and store meter data. This MDMS transforms these data into information that is used by the market including billing and clearing houses (settlement). The data in the MDMS are the source of truth for all the market(s) processes. For the purpose of this paper we will not cover the process of managing asset (meter) changes such as automatic recognition of new devices that some commercial MDMS softwares propose.
- **Classical Energy Market** is a commodity market that deal with the trade and supply of electrical energy. For

the sake of simplicity, per convention, we only consider that it is a forward day-ahead market. The prices for each market periods (e.g. 1/4h; 1/2h;1h ...) are fixed in day-ahead.

- **Generator**: is corresponding to power-generating facilities means a facility that converts primary energy into electrical energy and which consists of one or more power-generating modules connected to a network at one or more connection points ([4]). The energy produced is sold on the Classical Energy Market or in case of LEC also on the P2P Market.
- **Consumers**: use electricity.
- **Retailers**: procure the energy, that their consumers will use, at wholesale price on the Classical Energy Market. He/she is in charge of the billing to consumer of the energy bough on the Classical Energy Market but also of the all the taxes and network fees (cascade principle). This role covers the balancing responsible party (BRP) and is also accountable towards the Classical Energy Market for the good balance between the volume of energy buy and sell at each market period.
- **P2P market**: By extension of [5], we select the P2P model in which consumers and local generators located in a LEC are market-interconnected directly with each other, buying and selling energy services, regardless the network topology. It is a intra-day market meaning that the prices for one market period are set at the latest during the previous period. We will consider that the P2P market model is the one described in [6] with some adaptations as explained at Section III. Off course there are many other ways to cope this (e.g. [7] or [8])
- **Network operator**: deals with two tasks: the operation of the network (distribution or transport) and meter reading and processing for the market. For both activities and regardless the market's origin of the energy, regulated tariffs are set.
- **Clearing House**: As the speed of trades could be faster than the cycle time for completing the underlying transaction, clearing house is a service that ensures the process to settle the different transactions. For instance, a supplier switch or an end of the energy delivery contracts are managed by such clearing house.
- Notes :
  - prosumer: we do not consider the prosumer as an agent because for each market period it can be split between a generator role and a consumer role.
  - Figure 1 illustrates the different market period as defined in the present section

III. Functionalities of a P2P energy exchange connected to the classical market

A. *Adaption to the paper "consensus bases approach to peer-to peer electricity market"*

[6] describes a forward market. As mentioned in Section II, we consider that P2P Market is a sort of intraday forward market. This convention doesn't change the treatment of the optimization process proposed by [6]. As mentioned in this reference, the differences between what has been agreed regarding consumption or generation energy volume and the reality are treated during the balancing stage. This stage, in our use case, is settled in the clearing houses. Considering the local characteristic of the P2P, we also neglected the effect of network's constraints.

market time line.JPG

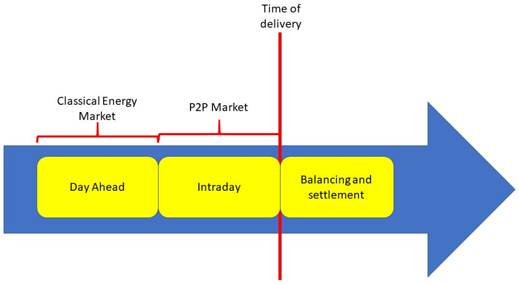


Figure 1. Simplified market time line.

In our use case, the energy required by a consumer could come from peers (one of multiple neighbor) or from its retailer. For the purpose of this paper, we will consider that retailers (one for each consumer) are also peer to peer agents with their own : (see [6]) :

- bilateral trading coefficient
- trade characteristics

For these agents-retailers, the "vectors of price estimates" (λ) are fixed because prices have been already defined during the day-ahead market. It is to note that the bilateral trading coefficient could be used to mimic the penalties that the retailer may apply when the consumer differs too many from its normal consumption profile.

Bertrand, pouvez vous valider cette approche ? merci

B. *Functional blocks of a combined P2P and classical markets*

To quote [9] applied to our use case, when a validating instance is required to create a bridge between a blockchain-based asset and its representation in the physical world, then the key aspect is not whether or not an entity can inscribe itself in a blockchain but rather how the control coupling works between the on-chain and off-chain entities. One implication is that tighter coupling is required between on-chain and off-chain assets when the smart assets concept is to be adopted as a virtual marshaling and control mechanism for assets in the physical world.

Like other energy blockchain (e.g. Pylon, PowerLedger,...), it is a good practice to have two parallel chains, one for the energy exchange ledger it self (the main chain) and a parallel for financial flows. The first energy exchange ledger can be a public chain while the second can be a private one. In P2P exchange, it is in the interest of the peer to integrate some flexibility (on the generation or consumption side) in flow P2P and MKT.JPG

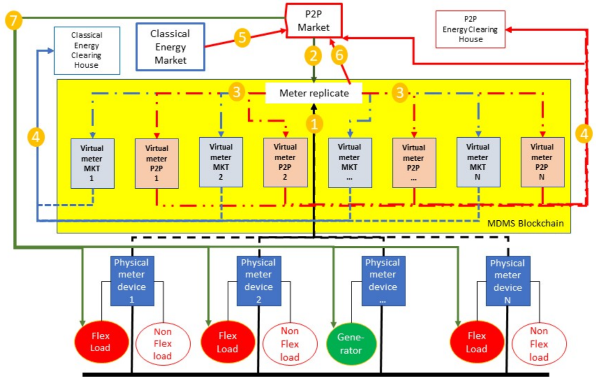


Figure 2. Information/commands flow linking the physical world and markets.

their decision sets in order to maximize their revenue/cost coming from local and wholesale markets. We think it could be possible to attach the control mechanism decision on the same (private) parallel chain. We will take this architecture model as assumption for this paper. The main chain will be used as the MDMS system, while the parallel chain will be used for the P2P Market platform.

The tight coupling between main (MDMS) and parallel (P2P Market) chains and physical world mentioned here above is, in our perspective, the cryptometer as described in [10].

Figure 2 illustrates information/command flows between cryptometer devices to market clearing houses through the MDMS and P2P Market.

The energy information from cryptometer to MDMS are represented with black doted lines (market with "1"). This information is encompassed in a first set of transactions recorded in the MDMS blockchain as a replicate of the physical meter. Based on information flow "2" that defines the repartition, this energy is split in the MDMS between energy related to P2P market and energy related to Classical Energy Market (represented by mark "3" with red lines for information flow for P2P market and blue lines for Classical Energy Market).

The repartition itself can be seen as a "smart contract" [11] using off chain services like ORACLIZE, TRUEBIT or iEXEC on Ethereum to connect an off chain optimization tool (e.g. Energy Management Service(EMS)). This EMS can use day-ahead price (link "5"), and transactions data (red lines "4" and "6"). It can also trigger some actions to flexible load or generation facilities (green line marked "7").

The information "3" is integrated in a second set of transactions recorded in the MDMS blockchain as virtual meters (one for the P2P market, and another for Classical Energy Market).

Finally, the information of both virtual meters can be used as data for the clearing houses (blue and red line marked "4").

C. Benefits of a MDMS based on blockchain

The main benefit of the MDMS based on the blockchain is the interoperability of the data source. Nowadays, network operators (as we defined it in Section II) read the data from the meters once a day (even with smart meter). There are also multiple steps of validation. For instance, in Belgium, the data are considered as pre-validated the day after. The second validation is done at after a month (to be precise at M+10 days). The final validation occurs months after the delivery time. This is low frequency data exchange is not suitable for near to real time market as P2P market is.

TABLE I
BLOCKCHAIN CONSENSUS ALGORITHM.

1) New transactions are broadcast to all nodes;
2) Each node creates a block with all the valid new transactions;
3) At each market period T_i a node is randomly selected and broadcasts its block;
4) Other nodes check the validity of the block and, if they agree, increment their chain;
5) If the majority of nodes agree, the block is definitively approved.

But these meters have a local data output (see description in [10]) that can be used for this type of side market. It is obvious that these data are not even pre-validated and some dissimilarities may occur between the classical and the P2P markets. These errors impose huge efforts in post-markets settlement.

A MDMS base on the blockchain could solve this issue, speed up validated data exchange and empower near to real time markets, because of:

- At the transactional level: The energy transactions are created at a very small scale are broadcast at high frequency.
- At the validation level: These transactions are (pre) validated when a block is mined.
- At the information source level: The same asset (the cryptometer) is used to provide unique energy information for both markets (classical and P2P).

IV. RECALL OF THE PREVIOUS WORKS

In [10], the authors described the general concept of a blockchain dealing with the interoperability by introducing the notion of block related to market period. This paper introduced the specific consensus algorithm summarized at Table I.

The selection of the miner (see step 3 at Table I) is described in Table II.

The simulations made in [12] demonstrated that:

- The issue "Nothing to stake" is correctly addressed by the voting process
- The main risks of the proposed consensus algorithm are
  - the generation of the random [1] number U_k for each candidate
  - the concentration of power/wealth which is related to the good choice for the value of α, β and γ

[1] We use the word "random" for the sake of simplicity, although the process is not purely random as it will be described in Section VI-B

TABLE II
PROPOSED MINER SELECTION ALGORITHM.

Let $\mathcal{K}$ be the set of nodes willing to support the chain at a specific time.

1) **Determine the wealth of each candidate miner.** The simplest definition of stake or wealth is the relative value of a node compared to the other nodes. This value can be derived from different criteria. In our use case, we choose the following wealth criteria to define the wealth of a node k, for a given $\mathcal{K}$ and for a time step T_i, as

$$W^k_{T_i} = \alpha E^k_{T_{i-1}} + \beta A^k_{T_{i-1}} + \gamma R^k_{T_{i-1}} \tag{1}$$

where we define

- E as the voting token corresponding to a subset of the volume of kilowatt-hours in the previous transactions (more kilowatt-hours increase the probability to generate the next block)
- A is an age measure of the previous block: how old is the last block created by a miner, how big is the probability to create the next one.
- R is a reputation measure: miners that have already created blocks than the other nodes will have a highest probability to be selected for the next block creation.

2) **Randomize.** Generate of a random number U_k for every candidate k with a uniform distribution in $]0, 1]$.

3) **Output.** The selected node has the maximum ratio W_k/U_k:

$$k^s_{T_i} = \arg\max_{k \in \mathcal{K}} \frac{W_k}{U_k} \tag{2}$$

V. PROBLEM STATEMENT

In [12], we have introduced the relation between *voting token*, *age of the last block* and *reputation* in $W_{t,k}$, namely $\beta A^k_{T_i-1} > \alpha E^k_{T_i-1} > \gamma R^k_{T_i-1}$. As such this formulation is incorrect because it does not take into account the volume of voting tokens that can be much higher than *age of the last block* or *reputation* for candidate. Upon that, if the volume of token sends by a candidate is an absolute value,the *age of the last block* and *reputation* have to be seen relatively with other candidates. It is to note that $E^k_{T_i-1}$, $A^k_{T_i-1}$ and $R^k_{T_i-1}$ are also related the full ledger's height. Indeed, it is obvious for *age of last the block* and *reputation*, though voting tokens are created with transactions, a block is a concatenation of transactions and the height of the ledger is the chain of blocks.In consequence, the amount of token that agents could have is limited by the amount of transaction contained in the full ledger. In other words, α, $\beta$and γ have to change at every market period in order to maintain the good balance between these parameters.

The objective of avoiding the concentration of wealth is still very key and the main idea to give priority to age of the last block remains valid. But still, the voting token is also important because it is the way that the *nothing to stake issue* is solved [10]. To find the good balance for these parameters, we simulate different possible configurations to define α, β and γ:

- Take the total volume of voting tokens
- Consider the total height of the ledger
- Limit the maximum voting token to the height of the ledger
- Consider only a subset of the full ledger (this is equivalent to give a maximum value for A and R)
- Combination of last two

To analyze these configurations, we record the results in function of which parameter determines the winning candidate. It could be $E^k_{T_i-1}$ (i.e. the winning candidate is the one that send the most voting tokens),$A^k_{T_i-1}$ (i.e.the winner is the one with the oldest minded block among the candidates),$R^k_{T_i-1}$ (i.e. among the candidates, the winner is the one who was most frequently a miner) but also U_k (i.e.none of the other criteria, the winner is only the consequence of the value of U_k). As this factor intervenes in the denominator of formula 1 and can take any value between $]0 : 1]$, it appears that, whatever the chosen configuration, in at least 80% of cases, it is this random number which determines the winner. This consideration rises another issue. For a rational candidate, what would be the number of voting tokens it should send to become a miner. In game theory, we can consider that this problem is equivalent to...The conclusion of this analyze is that rational candidates should send as few token as possible which is incompatible with the very principle of solving the *nothing to stake* problem. Therefore, we propose to adapt the proposed miner selection algorithm (see Table II). In this new method, we will still need to generate random number if there is a tie in the selection of the candidate. This will be the aim of the second problem to solve.

reprendre formule

The third issue is related to all public blockchain: Scalability. The scalability issue is the capacity of a blockchain to handle a growing amount of transactions, but also to maintain a comprehensive ledger with all the needed historical data. This is a very important topic in the blockchain eco-system. For instance, the Bitcoin's blockchain, the size of the block is often referred as the main issue in order to accept the same amount of transactions that classical financial services deliver (e.g. Visa or Master Card). Several attempts does not succeed to increase the block size from 1 Mb to 2 Mb [13].

When a blockchain growths and that a new node wants to join the chain, it has to load the full ledger. For Bitcoin or Ethereum, respectively launched in 2008 and 2014, it takes hours with a normal Internet connection to upload the full historic prior to be completely up and running. It is clearly a important drawback for P2P Market. To speed it up, there are some techniques or shortcuts proposed by both mentioned blockchain communities but these need to be correctly defined in order to maintain the same level of safety as having a full ledger. We will describe how reference points could be use to tackle this issue.

In the next Section we will treat the issue in the following order:

- New miner selection algorithm
- Providing ramdom number U_ω
- Reference points

VI. Blockchain consensus design

A. New miner selection algorithm

In place of computing a wealth as proposed in [10], we will consider that the selection of the miner is equivalent to a vote of the criteria *voting token*, *age of the last block* and *reputation* (the three criteria are the voters) among candidates miner. Though the theory of social choices and voting procedures is fare beyond the purpose of this paper, the reader can find some theoretical notions in [14] (chapter 4, paragraph 4). It has been proven by [15] that a voting mechanism cannot be both strategy proofness and non dictatorial. For our purpose the strategy-proofness principle is more important than non-dictatorial. This is why we do not consider voting mechanism like majority voting or Borda count ([16]). The voting mechanism that is the most suitable for the selection of the miner is Condorcet Voting System (CVS). Condorcet ([17]) finds that any alternative that is preferred to any other by the majority of voters should be selected. But there are cases where a Condorcet winner does not exist, the infamous Condorcet paradox. There are multiple way to solve this paradox. to cite only a few: Black method, Schulze, oriented grap, etc. To solve the paradox, we will use a simplified method from [18] that, if there is no deterministic Condorcet winner, the winner is randomly selected among the candidates who are tied. Giving the fact that there are potentially much more candidates than voters, the risk of tie is important. The question about the weight of α, β and γ presented in [10] is now transfered to the number of votes that corresponds to each parameter. In other words, we will have:

- voter $\mathcal{E}$ with α votes for the criteria *voting token*
- voter $\mathcal{A}$ with β votes for the criteria *age of the last block* and
- voter $\mathcal{R}$ with γ votes for the criteria *reputation*

We have a finite set of $N = |\mathcal{K}|$ candidates. Every voters express its preference on a set $\| = \{1, .., g, h, ..., k, ..., n\}$.

For g and h, two distinct elements of $\mathcal{K}$, let's define $M_{gh}^{P} = +1$ if the candidate g beats candidate h for the voter P, $M_{gh}^{P} = -1$ if candidate h beats g, $M_{gh}^{P} = 0$ if there is an equality between candidates g and h. To give an illustration of this rules, if candidate g has send E_g voting tokens and candidate h has send E_h tokens, then $M_{gh}^{\mathcal{E}} = 1$ if $E_g > E_h$,$M_{gh}^{\mathcal{E}} = -1$ if $E_g < E_h$ and $M_{gh}^{\mathcal{E}} = 0$ if $E_g = E_h$.

We can create a voting matrix $\mathcal{M}^P$, of dimensions NxN, with all the combinations for the voter P :

$$\begin{bmatrix}
0 & M_{21}^P & M_{31}^P & ... & M_{g1}^P & M_{h1}^P & ... & M_{k1}^P & ... & M_{n1}^P \\
M_{12}^P & 0 & M_{32}^P & ... & M_{g2}^P & M_{h2}^P & ... & M_{k2}^P & ... & M_{n2}^P \\
M_{13}^P & M_{23}^P & 0 & ... & M_{g3}^P & M_{h3}^P & ... & M_{k3}^P & ... & M_{n3}^P \\
... & ... & ... & 0 & ... & ... & ... & ... & ... & ... \\
M_{1g}^P & M_{2g}^P & M_{3g}^P & ... & 0 & M_{hg}^P & ... & M_{kg}^P & ... & M_{ng}^P \\
M_{1h}^P & M_{2h}^P & M_{3h}^P & ... & M_{gh}^P & 0 & ... & M_{kh}^P & ... & M_{nh}^P \\
... & ... & ... & ... & ... & ... & 0 & ... & ... & ... \\
M_{1k}^P & M_{2k}^P & M_{3k}^P & ... & M_{gk}^P & M_{hk}^P & ... & 0 & ... & M_{nk}^P \\
... & ... & ... & ... & ... & ... & ... & ... & 0 & ... \\
M_{1n}^P & M_{2n}^P & M_{3n}^P & ... & M_{gn}^P & M_{hn}^P & ... & M_{kn}^P & ... & 0
\end{bmatrix}$$

We make a new matrix called *total vote matrix* $\mathcal{M}^{tot} = \alpha.\mathcal{M}^{\mathcal{E}} + \beta.\mathcal{M}^{\mathcal{A}} + \gamma.\mathcal{M}^{\mathcal{R}}$.

It is possible to define that g is strictly preferred for every voters to h if and only if $M_{gh}^{tot} > 0$. By extension, k is an unique Condorset winner if it is the only row with all values $M_{kx}^{tot} > 0$ with $x = [1, ..., k[U]k, ..., n]$ (obviously, the value of $M_{kk} = 0$). If there are more than one candidate that fulfills this condition, it is not possible to determine an unique Condorset winner and all the candidates in this situation are put in a ballot, called $\mathcal{B}$. The winner is then drawn from this ballot. For each candidate ω in $\mathcal{B}$, we will provide a random number U_ω. The procedure is described in the next section.

B. Providing random number

The purpose of U_ω is to select a miner "randomly" between all candidate nodes that are in a tie situation. To be more precise, the aim is to have a process of selection for which there is no way to know exactly, in advance, which node will be selected. Upon that objective, this process has to be transparent in order to avoid manipulation coming from a malicious node.

The unbiased nature U_ω is only met if it is based in an exact and predetermined way on random information to be revealed in the future and therefore can not be known in advance by the party specifying the algorithm neither by the node that will used it (the actual miner as described in [10]). The random information must be such that it will be publicly and unambiguously revealed in a timely manner and can be controlled by other nodes.

Random sources should not include anything that any reasonable stakeholder could believe to be under the control or influence of the actual miner. As source for U_ω, we selected the number of candidates, the public address of the candidate, the amount of transactions and the energy volume in the proposed block.

For obvious reasons, any reference to temporal or chronological data (such as timestamps of votes) has been avoided.

The pseudo code to generate U_ω for each candidate is shown at table III. The selected candidate is the one with the highest ratio $\omega_{Ti}^{s} = \arg\max_{\omega \in \mathcal{B}} U_\omega$. Giving the collision free properties of the hash function, there is no risk to have a tie.

C. References points

To solve the scalability issue, we have to admit that, for a node, keeping the whole ledger from the origin is not always worth. For instance, on a legal point of view, the European regulation imposes that meter data must be kept for at least five years.There is less benefit to keep the transaction older than five years. But one of the biggest advantage of the blockchain is traceability. A non malicious node can always prove the link between all blocks to the genesis block. The question is than,for a non malicious node, how is it possible to create a strong reference point as the genesis block is .

find reference

To achieve this goal, we propose to create multiple references points separate by an epoch e. This method is inspired from Byzantine Fault Tolerance and Casper. For each refer-

add reference

TABLE III
PSEUDO-CODE TO GENERATE RANDOM NUMBER.

1) The following sources are captured;
   - $Z = |\mathcal{B}|$: amount of candidates in a tie situation
   - H^a_ω : public key of each candidate $\omega \in \mathcal{B}$ (address hash)
   - V^e_{Ti}: sum of the energy trades in all transactions for the proposed block
   - Q^t_{Ti}: amount of transactions for the proposed block
2) Calculate for each candidate $\omega \in \mathcal{B}$:
   - Hash function of the random sources:

$$H^h_{\omega_{Ti}} = hash.sha256(Z; H^a_\omega; V^e_{Ti}; Q^t_{Ti}) \qquad (3)$$

   - Transform the hash function into digit number:

$$H^d_{\omega_{Ti}} = H^h_{\omega_{Ti}}.hexdigest() \qquad (4)$$

3) Sum all the digit numbers of Z candidates:

$$H^t_{Ti} = \sum_{j=1}^{Z} H^d_{j_{Ti}}; \qquad (5)$$

4) Calculate U_ω for each candidate ω

$$U_\omega = \frac{H^d_{\omega_{Ti}}}{H^t_{Ti}} \qquad (6)$$

ence points, there are six phases: request,pre-prepare, prepare, commit, reply and fix. This point of reference corresponding to a time period T_i is initiated by the miner that has been designed for the block at T_i. We will name this miner M_i. The process of the reference point is done at the same time as the communication of the winner of the block T_{i+1} (see step 4 in table IV "Transaction for candidate selection" in [10]). We make the assumption that the communication network is asynchronous or may fail to deliver proper message (not deliver at all, delay it, or duplicate it). By convention, all messages will have the following structure: $< phase, T_i, t, (m) >_{\sigma_i}$ with :

- Phase: the corresponding phase (request, pre-prepared, ...)
- T_i: the height of the reference point to validate (target) corresponding to the number of the block of the last reference point added with the epoch e.
- t: time stamp corresponding to the market period of the block N
- (m): the message to be exchanged
- σ_i: signed by i. It should be noted that the key is well known because the parties involved in creating a reference point are all old miners and their public key is part of the information in their block header. In consequence, the signatures can always be checked.

The specific process is described in Table IV.

Figure 3 illustrates this process with $e = 4$ and if M_{i-3} does not answer.

Some slashing rules need to be set. First of all, M_j (also called *validator*) must not send more than one reply message for one given t and n. Secondly, considering the targeted reference point, M_i must send a request message with the correct T and t to the right M_{i-e}. And finally, the selection rules ($f+1$ and $g+1$) has to be correctly apply by every miner during the commit phase and reply phase.

TABLE IV
REFERENCE POINT PROCESS

1) **Request phase**: M_i sends the message $< request, T_i, t, (n) >_{M_i}$ to the block miner of the previous reference point M_{i-e} with n equal to the sequence number (more on that later)
2) **Pre-prepare phase**: M_{i-e} replicates the request by sending it to all x block's miners in the interval $[T_{i-e}, T_i[$ (M_x with $x \in [i-e; i[$) with the message $< pre-prepared, T_i, t, (n) >_{M_{i-e}}$. This message is also send to M_i as acknowledge.
3) **Prepare phase**: Every x miners replicate the request to all miners from the same interval with the message $< prepared, T_i, t, (n, AM_{i-e}) >_{M_x}$ with AM_{i-e} the public key of the M_{i-e}.
4) **Commit phase**: each miner that has received a *pre-pare* message (named M_j)sends back the following $< commit, T_i, t, (h(T^\theta)_j, n, AM_{i-e}) >_{M_j}$ with $h(T^\theta)_j$ is the hash function of block corresponding to the target reference point computed by j considering that the target block was mined by θ.
5) **Reply phase**: each j miner sends to M_{i-e} a reply message $< reply, T_i, t, (h(T^\iota)_j, n) >_{M_j}$.
   The hash functions $h(T^\iota$ is determined by the most frequent committed hash $h(T^\theta)$, but at least $f+1$.
6) **Fix phase**:M_{i-e} fixes the reference point by selecting $h(T^\tau)_{def}$ among the most frequent hash $h(T^\iota$ included in *Reply* message but at least $g+1$. This information is send back to M_i. If $h(T^\tau)_{def}$ is equal to the hash of it own block, it puts this information together with the transaction winner for the market period T_{i+1} (see step 4 in table IV "Transaction for candidate selection" in [10]).

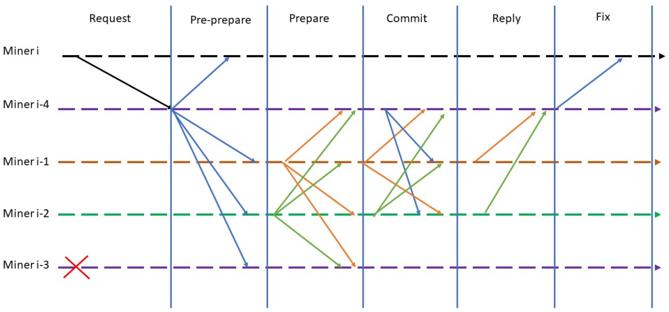


Figure 3. Illustration of reference point algorithm.

When one of these rules are broken, the guilty miner's reputation is put to an high negative number.

Given the assumption about asynchronous network, there is a need to define some priority rules. For the different stages of the reference point algorithm, there is a maximum time latency associated. Let's call this maximum latency t_y with y standing for the phases: rq for request, pp for pre-prepare, pr for prepare, co for commit, re for reply, fx for fix. We define:

$$t_{tot} = t_{rq} + t_{pp} + t_{pr} + t_{co} + t_{re} \qquad (7)$$

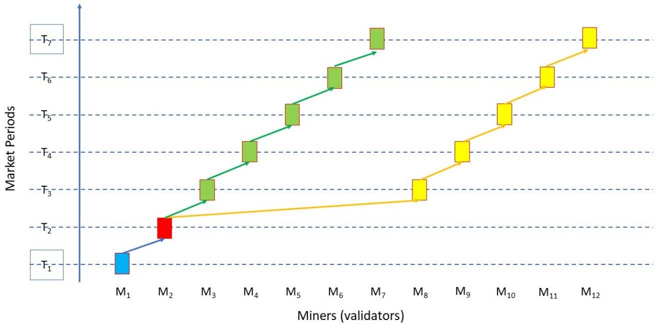


Figure 4. Example of reference point algorithm process dealing with fork.

The start time of the process is the gate closure time (see step 3 in table IV "Transaction for candidate selection" in [10]) that we name t_{gc} and is corresponding to $T_i - t_e$.

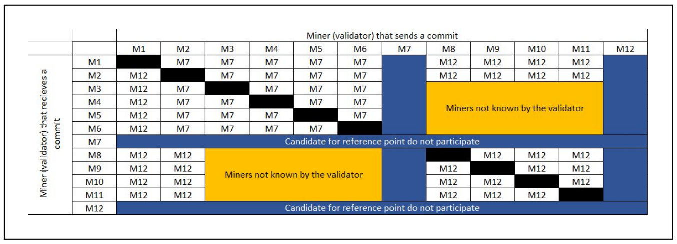
At $t_{gc} + t_{rq}$, if M_{i-e} do not receive a request message, it initiates the process itself by sending a pre-prepared message to all miners between $i-1$ to $i-x$.

If M_i did not receive at $t_{gc} + t_{rq} + t_{pp}$ the acknowledge from M_{i-e}, it starts a back up process. It will send the following message $< pre-prepared, T_i, t, (n, M_i) >_{M_i}$ to all *validators*. This message has the priority all other messages coming from M_{i-e}. At $t_{gc} + t_{tot}$, if M_{i-e} receives less than d reply messages, it will restart the process. In this case, the value of n is incremented. All the other *validators* will than consider that the message with the higher n has the priority. Finally at $t_{gc} + t_{tot}$, if M_i do not response at all (even for the block mining process), M_{i-e} will replace it. In this case, the block will be broadcast with a delay regarding the market period but it will be mined correctly.

The process of reference point strengthens the whole consensus mechanism, particularly in the occurrence of a fork. To explain this, we will take an example (see Figure 4)

Let's have 12 miners (form M_1 to M_{12}). At every market period T_i, all candidates propose their block and participate to the voting process. One of then is selected and become the miner and its block is the canonical block (colored block in figure 4). The winning transaction is represented by arrows. There is a reference point every six blocks (epoch e=6) (squared time period T_1 and T_7 in Figure 4). At T_2 (just after reference point T_1), M_2 becomes malicious and does not select one miner but select M_3 and M_8. We assume that some nodes will follow the branch initiated by M_3, and others support the branch of M_8. Even if now M_2 cannot make any other canonical block (it has been slashed), there is a fork.It is important to note that it is not because there is a fork that transactions in blocks in one of the branch are not correct. For the first branch, the process is initiated by M_7 at T_7. For the second branch, the process starts with M_{12}. Miners from a branch do not recognize miners from another branch. Table V shows the commit messages exchange. The columns are miners sending the message, while the rows miners receiving the message e.g. M_6 sends to M_1 to M_5:

TABLE V
ILLUSTRATION OF RESOLUTION OF A FORK: COMMIT PHASE.



		Miner (validator) that sends a commit											
		M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
Miner (validator) that recieves a commit	M1	■	M7	M7	M7	M7	M7		M12	M12	M12	M12	
	M2	M12	■	M7	M7	M7	M7		M12	M12	M12	M12	
	M3	M12	M7	■	M7	M7	M7		Miners not known by the validator				
	M4	M12	M7	M7	■	M7	M7						
	M5	M12	M7	M7	M7	■	M7						
	M6	M12	M7	M7	M7	M7	■						
	M7	Candidate for reference point do not participate											
	M8	M12	M12	Miners not known by the validator					■	M12	M12	M12	
	M9	M12	M12						M12	■	M12	M12	
	M10	M12	M12						M12	M12	■	M12	
	M11	M12	M12						M12	M12	M12	■	
	M12	Candidate for reference point do not participate											

TABLE VI
ILLUSTRATION OF RESOLUTION OF A FORK: REPLY PHASE.

Miner	Reply
M_2	M_{12}
M_3	M_{12}
M_4	M_7
M_5	M_7
M_6	M_7
M_8	M_{12}
M_9	M_{12}
M_{10}	M_{12}
M_{11}	M_{12}

$< commit, T_7, t, (h(T_7^{M_7})_{M_6}, n, AM_1) >_{M_6}$
While M_{11} sends to M_1; M_2 and from M_8 to M_{10} the message
$< commit, T_7, t, (h(T_7^{M_{12}})_{M_{11}}, n, AM_1) >_{M_{11}}$
The Table VI gives results of the reply messages to M_1. Giving the fact that the majority of the *Reply* points M_{12} as the reference point, the message *fix* for M_1 will be:
$< Fix, T_7, t, (h(T_7^{M_{12}}), n) >_{M_1}$
This message is send to both M_7 and M_{12}. M_7 and all the miners of block between T_3 and T_7 have to converge to the yellow branch of Figure 4. Miners M_3 to M_7 can keep the voting token that they earned but loose some reputation. The fork is now reduce to only one branch, and is solved.

Nodes may choice how many reference points is needed to consider subset of the full ledger as strong as the genesis block. Until now, we do not speak about the blocks'size. It is clear that this parameter impacts the choice of the node.

The block size is related to the size of one transaction, the amount of transactions that a block could contain and the size of the block header. We ran some simulations to assess different structures. For a MDMS for residential customers, we used the following parameters:

- Size of transactions
  - size of 1 transaction: 200 bytes
  - 3 transactions for each P2P market participant: replication of the physical meter (the basic transaction), separation between P2P market and Classical Energy Market.
  - In [19], we consider that the MPE for residential customer is 2%
  - As explained in [10], a transaction is created at each $2\%.E_{\frac{1}{4}}$ with $E_{\frac{1}{4}}$ the average energy consumed per quarter hour. So there are $\frac{1}{MPE}$ basic transactions

per market period (50 with our parameters) and in total 150 transactions per market period for P2P market participants.
  - For each P2P market participant, 30 kBytes is needed for each market period.
- Size of header: 300 bytes

With 100.000 P2P market participants to start with, the maximum block size can be put at 3.5 MBytes. If, this first amount of participants is reached, the block size can be updated accordingly. On one hand, with these numbers, for a node that wants to spend maximum 32 GBytes of memory (e.g. standard micro SD card), this is corresponding to 95 days of transactions. If there is a reference point at each 30 days (epoch, $e = 2880$), this node will have three reference points in its subledger that can be considered as good as the genesis block.On the other hand, parties like DSO could be obliged (e.g. by regulation) to keep the ledger length for at least five years, that corresponds to 365 GBytes storage space. The reader may agree that this space is not an issue at all for the nowadays technologies.

comparer ceci à la mémoire d'un smart meter type linky ou AMR

We have also to consider the influence of reference points on the algorithm about selection of the miner.Indeed, if a node that becomes a miner doest not have the full ledger from the genesis block, how is it possible to determine for each candidate the value of $A^k_{T_{i-1}}$ and $R^k_{T_{i-1}}$?

to be further analyzed

VII. Conclusion and further work

conclusion: different type of node (full - light) structure of the block head further work use of side-blockchain connected to MDMS BC to support P2P market How can we insentivize node that kept more than 3 reference point ?

VIII. Acknowledgment

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[10] D. Vangulick, B. Cornélusse, and D. Ernst, "Blockchain for peer-to-peer energy exchanges: design and recommendations," in *Proceedings of the XX Power Systems Computation Conference (PSCC2018)*, 2018.

[2] "Loi 2017-227 du 24 février 2017 ratifiant les ordonnances 2016-1019 du 27 juillet 2016 et 2016-1059 du 3 août 2016," February 2017.

[3] D. Vangulick, B. Cornélusse, T. Vanherck, O. Devolder, and D. Ernst, "E-cloud, the open microgrid in existing network infrastructure," in *Proceedings of the 24th International Conference on Electricity Distribution*, 2017.

[4] EU, "European commission regulation 2016/631/eu," April 2016, (Official Journal L 112, 29 April 2016).

[5] Y. Parag and B. K. Sovacool, "Electricity market design for the prosumer era," *Nature Energy*, vol. 1, no. 4, p. 16032, 2016.

[6] E. Sorin, L. Bobo, and P. Pinson, "Consensus-based approach to peer-to-peer electricity markets with product differentiation," *arXiv preprint arXiv:1804.03521*, 2018.

[7] M. Mihaylov, S. Jurado, N. Avellana, K. Van Moffaert, I. M. de Abril, and A. Nowé, "Nrgcoin: Virtual currency for trading of renewable energy in smart grids," in *European Energy Market (EEM), 2014 11th International Conference on the*. IEEE, 2014, pp. 1–6.

[8] F. Olivier, D. Marulli, D. Ernst, and R. Fonteneau, "Foreseeing new control challenges in electricity prosumer communities," in *Proc. of the 10th Bulk Power Systems Dynamics and Control Symposium – IREP 2017*, 2017.

[9] M. Swan and P. de Filippi, "Toward a philosophy of blockchain: A symposium: Introduction: Introduction," vol. 48, pp. 603–619, 10 2017.

[11] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 254–269.

[12] D. Vangulick, B. Cornélusse, and D. Ernst, "Blockchain for peer-to-peer energy exchanges: Probabilistic approach of proof of stake," in *Proceedings of the CIRED WORKSHOP (CIRED 2018)*, 2018.

[13] "2x called off: Bitcoin hard fork suspended for lack of consensus."

[14] S. J. Brams and P. C. Fishburn, "Voting procedures," *Handbook of social choice and welfare*, vol. 1, pp. 173–236, 2002.

[15] K. Arrow, "Individual values and social choice," *Nueva York: Wiley*, vol. 24, 1951.

[16] J. C. de Borda, "Mémoire sur les élections au scrutin," 1781.

[17] J. A. N. de Caritat Condorcet, "Essais sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix," 1785.

[18] L. N. Hoang, "Strategy-proofness of the randomized condorcet voting system," *Social Choice and Welfare*, vol. 48, pp. 679–701, 2017.

[19] EU, "European commission directive 2014/32/eu," March 2014, (Official Journal L 96, 29 March 2014).