Business-driven Blockchain-Mempool Model for Cooperative Optimization in Smart Grids

Marius Stübs[1], Wolf Posdorfer[2], and Julian Kalinowski[2]

[1] Security and Privacy Working Group
[2] Distributed Systems Working Group,
University of Hamburg, Germany
`{stuebs,posdorfer,kalinowski}@informatik.uni-hamburg.de`

Abstract. Smart Grid control can be represented as a set of optimization problems. Aggregators, called Virtual Power Plants, are using dispatch schedules as solutions to these optimization problems. Power generation of distributed energy resources has to match the consumption of the electrical load distributed over the grid. An established method of optimizing Smart Grid control is to calculate an optimal solution as a schedule vector over all controllable generators and loads.
In this paper, we describe a distributed way of verifying and agreeing upon a solution for this optimization problem. In order to meet the high standards regarding authentication and accountability, we incorporate blockchain technology and propose a Mempool Model for benevolent selection criteria.

Keywords: Blockchain, Mempool, Smart Grid, Virtual Power Plant

1 Introduction

Communication systems for Smart Grids have high standards for trusted authentication and accountability. Scientific publications indicate that electricity generation is expected to become more decentralized, and several researchers propose the use of Multi Agent Systems for distributed Smart Grid control [7]. Especially Smart Grid control can heavily profit from incorporating distributed decision verification as used in blockchain technology, and recent results show that energy trade can be executed based on blockchain transactions [13].

Literature in Power Systems Optimization has established two general approaches for scheduling: Competitive and Cooperative Scheduling [3]. Since competitive scheduling schemes are mainly based on single transactions like bilateral trade, simple blockchain adaptations exist [12]. For cooperative scheduling however, the interdependability of all power plants' schedules is generally higher. Thus, these approaches are harder to implement as blockchain transactions.

We propose to organize Virtual Power Plants (VPPs) in a distributed blockchain-based system. Each Distributed Energy Resource (*DER*) is equipped with a blockchain node and acts as an independent node. Nodes try to solve an optimization problem to come up with an optimal dispatch schedule for the VPP.

Since nodes are free to choose the utilized solving strategies, each node can potentially provide a different schedule.

VPPs act as single entities towards the electricity grid, but on the inside they orchestrate the scheduling of all connected physical power plants [18, 17]. VPPs are highly suitable for cooperative scheduling schemes, since they represent an isolated system with a clear goal. Further, blockchain based scheduling fits the distributiveness and the need for verification of scheduling decisions.

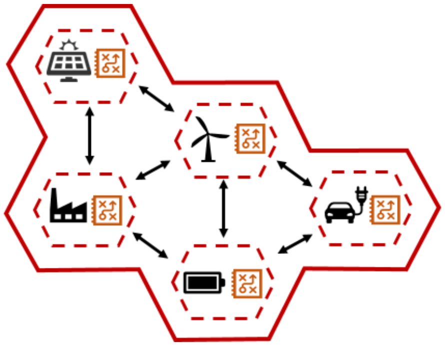

Fig. 1. VPP with renewable energy resources

We propose a cooperative, blockchain based power system scheduling scheme for VPPs. Figure 1 shows a sketch of a VPP that is coordinating five distributed energy resources, each of them associated to a blockchain node and capable of calculating solutions of optimal DER dispatch. Every DER inside the system will represent a Validator node to actively partake in the consensus. This VPP is distributed, meaning that it works collectively without a central control room.

2 Related Work

A number of publications discuss the application of blockchain for energy trade:

Münsing et al. present an architecture for peer-to-peer markets based on blockchain [13]. The DERs perform local optimizations of the schedules, however consensus is only achieved using a smart contract. This essentially results in many nodes unnecessarily performing the same calculations.

Since literature provides a variety of both approaches – Competitive and Cooperative Scheduling [3] – it is surprising that to our knowledge only competitive approaches have been tested for incorporation of blockchain [1].

Zhang et al. present a convincing categorization of Cooperative Smart Grid Scheduling approaches [20]. Since in many approaches the optimization objective is one-dimensional, it would be feasible to use these methods in conjunction with this paper's approach.

3 Virtual Power Plant Dispatch Schedules

In a VPP the power dispatch is being scheduled according to energy trades, with the goal to match supply and demand. For a given time period, a VPP has to meet the demand by scheduling energy production to the different DERs.

We consider a set of n distributed energy resources (DERs) $i = 0, ..., n$ that are orchestrated by a VPP. Their respective dispatch schedules s_i are included in the schedule vector S. Each s_i describes the generation or load of DER i, while S contains all of these, as seen in Equation 1.

$$S = \langle s_0, s_1, s_2, ..., s_n \rangle \tag{1}$$

The overall power-system includes electrical loads, whose demand has to be met. For every investigated time frame, the entirety of DERs must optimize its schedule, so that Equation 2 is met.

$$demand(t) = generation(t) \tag{2}$$

3.1 Cost function based optimization

We make use of the fact that it is difficult to find an optimal solution, but it's easy to check the validity of any solution regarding their boundary conditions and it's easy to compare the quality of two existing solutions. That way, every node is free to use any optimization formula, but in the end, the best solution can be easily verified. The cost associated with the respective dispatch schedule is written as $cost(s_i)$. The overall cost function is described with Equation 3:

$$cost(S) = \sum cost(s_i) = cost(s_0) + cost(s_1) + ... + cost(s_n) \tag{3}$$

The decentralized optimization scheme established in [13] optimizes the schedules of distributed energy resources (DER) based upon the market price and is an example of an optimization method that could be used in a node.

4 Blockchain

A blockchain is a decentralized data structure, which maintains its internal consistency over a distributed application state through a distributed consensus mechanism. Every participating node in the network has a fully replicated and synchronized view of the data [2].

By submitting transactions the state of the blockchain can be altered. Transactions are bundled into blocks, which are chained together by hashing the hashes of the transactions and the previous block. Thus, each block is chained to its predecessor and a definite order is created.

Figure 2 shows a simplified blockchain. Each block contains a *merkle tree hash* (Tx_Root), a *nonce*, a *timestamp* and the predecessors block hash. By

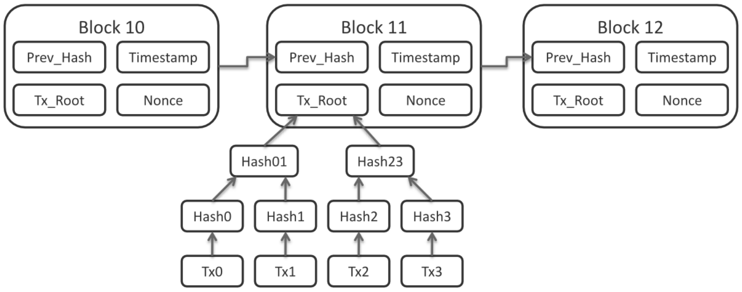

Fig. 2. Simplified blockchain datastructure [16]

including the previous hash (Prev_Hash) blocks are linked to their predecessor. The merkle tree hash is the hash over all its leaf nodes *Tx0* through *Tx3*.

Transactions are created by network participants and propagated to other nodes through a peer-to-peer network. In so-called "Cryptocurrencies" (e.g. Bitcoin [14]) they usually contain currency or trade information. Depending on the technology they can also contain function calls or application specific data.

4.1 Consensus

Through a consensus algorithm every blockchain ensures that most of its nodes share the same valid global state at all times. Depending on the consensus algorithm the majority of nodes necessary for a valid block is typically defined by 51% of nodes in chain-based algorithms or $+\frac{2}{3}$ in Byzantine Fault Tolerant (BFT)-based algorithms.

In many Cryptocurrencies, like Bitcoin and similar technologies, the consensus algorithm is based on *Proof-of-Work* (POW)[4, 14, 19]. PoW is essentially a cryptographic challenge or brute-force race to provide trust. The challenge is to hash a combination of *Nonce* and *Root-Hash* until a certain target hash is found. The target hash, or target difficulty, must contain a certain amount of leading zeros to be accepted by other nodes. The process of finding a nonce (brute-force) is called *Mining*, while nodes participating in this challenge are called *Miners*. PoW ensures the following three criteria. Firstly, verification of a valid block becomes computationally trivial, as participants only have to hash the nonce and the root hash and verify that it meets the target difficulty. Secondly, it ensures that very rarely two blocks for the same height are propagated through the network. Lastly, replacing previous blocks becomes computationally unfeasible as finding different nonces becomes harder with every new block that is appended.

In order to limit the block creations in a given time window and adjust to increasing computational power of the network the difficulty is frequently automatically adjusted. This ensures that the median time between the creation of two consecutive blocks is roughly the same. By adding more Miners to the network the increased computational power would reduce the median time, which in turn would also lead to a self-adjusted increase of the difficulty to keep the median time span intact. Therefore, removing nodes from the network will decrease the time span, which will result in a decrease of difficulty.

Contrary to PoW-Algorithms, the *Proof-of-Stake* (PoS) family of algorithms reduce the waste of energy and resources resulting from brute-forcing of *Nonces* [6, 8, 9]. Miners can stake their own coins (or assets) to decrease the target difficulty of their next block. The decrease of difficulty is directly linked to the amount of coins staked by a miner. Depending on the blockchain technology, further criteria are employed to mitigate the risk of coin hoarders controlling the network. One of these criteria can be the *Coin-Age* as used by PPCoin [9].

As not all blockchain technologies necessarily contain a currency there exists a different kind of PoS algorithm. In these BFT-based PoS algorithms there is a certain set of nodes holding leader positions, who are directly responsible for consensus. The algorithms are oftentimes based on the PBFT-Algorithm [5] or 2-Phase-Commit protocols [11]. These Leader Nodes (Validators) are taking turns in a deterministic round robin order to propose a new block. This ensures that only one block for a given height is created. Thus, BFT-based algorithms can never have two competing blocks for the same height, which prevents forking.

For our approach, we make use of a BFT-based PoS algorithm based on Tendermint [10].

5 Consensus and Cooperative Scheduling

To reach consensus, all blockchain nodes usually follow the same set of validation rules for transactions such as validation of syntax and semantics. The latter is application-specific and ranges from verifying a participants balance in cryptocurrencies to executing programs in smart contract applications. Existing approaches are based on common ground: All (good) nodes perform the same validation operations to come to the same conclusion and reach consensus.

However, the process of selecting transactions from Mempool is different for each node. To maximize profit, miners usually follow selfish selection criteria for choosing transactions for inclusion in the next block [15]. For example, transactions with higher fees or tips are favored by miners as they directly go towards the miners balance. This promotion of selfish behaviour is unfavorable in a system where cooperation is key.

Mapping this basic consensus concept to a DER network leads to an optimization problem which would be solved by all participating DER (blockchain) nodes. And they would all have to use the same solver and parameters as only a majority agreeing on the exact same solution will lead to consensus.

In a cooperative scenario, however, it could be a good idea to allow for different solutions provided by different nodes and still agree on a common solution. A possible (naive) approach could include the following steps:

1. each node calculates it's own solution
2. solutions are broadcasted as transactions
3. all transactions are valid (if they meet the boundary conditions) and included in the next block

4. in a follow-up transaction, someone decides on the best solution
5. majority agrees if this is really the best solution from their point of view

This approach has an important flaw: Many transactions will be persisted inside a block that contain inferior solutions, thus the blockchain will get bloated with useless data. Also, from a conceptual point of view, an additional role has to be established in order to decide on the best solution in a centralized manner.

6 Mempool Model

In order to consolidate the concepts of blockchain and decentralized power system scheduling, we propose a custom blockchain implementation. In the following, we will describe the implementation by highlighting the differences to a generic proof-of-stake blockchain such as Tendermint [10].

Upon reception of a transaction, a blockchain node validates it and puts it into the *Mempool*. The Mempool holds transactions that are not part of a block yet and is synchronized across nodes. We define a transaction type $T_{solution} = (S, cost(S))$, dispatched by every node that wants to contribute a schedule for a given time frame. Other nodes can validate a transaction by re-calculation the costs for the given schedule and discard it, if they don't match the schedule.

This serves two purposes: First, the early validation makes sure invalid transactions are not broadcasted through the whole network. Second, the selection process is based on the schedule costs, which is why validity has to be ensured.

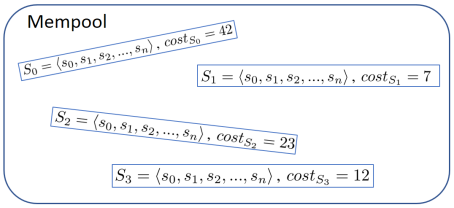

Fig. 3. Exemplary Mempool containing four transactions

Figure 3 shows an example a Mempool containing four pending transactions of the aforementioned transaction type with different costs associated. According to our proposal, only the transaction with the lowest cost can be selected for the next block. In the example, this would be schedule vector S_1 with cost 7.

Instead of creating blocks from Mempool transactions by criteria like time of reception or fees earned, nodes will *select* the single best schedule in respect to the given cost function and include it in the next block proposal. Afterwards, all other schedules still residing in the Mempool will be discarded.

As all (good) nodes operate using the same cost function and boundary conditions, nodes will accept the proposal if it is indeed the best known solution. Any new block is validated by all other nodes by comparing its transactions to the schedule vectors in their Mempool $\mathbb{M}$. If the proposed block contains a schedule vector $S_{proposal}$, while in the Mempool there is a schedule vector S_i with lower costs, the block is declined as depicted in Equation 4.

$$\begin{gathered}\text{if exists } S_i \in \mathbb{M} \\ \text{with } cost(S_i) < cost(S_{proposal}) \\ \text{then decline } block \text{ with } S_i.\end{gathered} \tag{4}$$

The described selection process provides deep integration of the business-oriented cooperative proposal/selection process and the core functions of a custom blockchain solution. Only the optimal schedule vector is accepted into the blockchain. The remainder of transactions can be discarded from the Mempool after a schedule for the given time frame is persisted inside a block.

7 Evaluation and Future Work

The proposed approach avoids cluttering the blockchain with blocks that contain to-be declined schedule vectors. In comparison to existing approaches based on smart contracts [13], our Mempool model allows for mapping cooperative scenarios to a blockchain based system where a solution does not have to be computed n-times (with n : number of nodes). Nevertheless, validation is performed both at transaction level (correctness of cost-calculation) and at block level (best available solution), thus providing valid results and integrity.

The preliminary results indicate a significant reduction in blockchain size, by picking only transactions from the Mempool which satisfy specific optimization criteria. Also the validity of the selected results is kept at its original level.

Although the approach is targeted on cooperative scenarios, additional measures are required to make it robust against implementational or operational errors or misbehaving nodes. Examples may be missing (valid) proposals for given timeslots or optimal solutions that didn't propagate to a majority of nodes.

In future experiments, we will validate the scenario through an example grid using a real-time digital simulator.

8 Conclusion

The proposed solution shows an optimization scenario for cooperative blockchain systems by applying business-driven selection and inclusion criteria for transactions. Nodes proposing a new block will have to select transactions based on minimum cost of the schedule instead of maximum block reward. In this cooperative environment all nodes inherently try to achieve the same goal.

The proposed solution is generalizable and can be applied to all processes that have multiple concurrent valid transactions but only a few transactions qualifying for optimal solutions.

References

1. Aitzhan, N.Z., Svetinovic, D.: Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. IEEE Transactions on Dependable and Secure Computing 15(5), 840–852 (2018)

2. Antonopoulos, A.M.: Mastering Bitcoin: unlocking digital cryptocurrencies. O'Reilly Media, Inc. (2014)
3. Atzeni, I., Ordóñez, L.G., Scutari, G., Palomar, D.P., Fonollosa, J.R.: Noncooperative and cooperative optimization of distributed energy generation and storage in the demand-side of the smart grid. IEEE Trans. Signal Processing 61(10), 2454–2472 (2013)
4. Buterin, V., et al.: A next-generation smart contract and decentralized application platform (2014)
5. Castro, M., Liskov, B., et al.: Practical byzantine fault tolerance. In: OSDI. vol. 99, pp. 173–186 (1999)
6. David, B.M., Gazi, P., Kiayias, A., Russell, A.: Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake protocol. IACR Cryptology ePrint Archive 2017, 573 (2017)
7. Harmouch, F.Z., Krami, N., Benhaddou, D., Hmina, N., Zayer, E., Margoum, E.H.: Survey of multiagents systems application in microgrids. In: Electrical and Information Technologies (ICEIT), 2016 International Conference on. pp. 270–275. IEEE (2016)
8. Jain, A., Arora, S., Shukla, Y., Patil, T., Sawant-Patil, S.: Proof of stake with casper the friendly finality gadget protocol for fair validation consensus in ethereum. International Journal of Scientific Research in Computer Science, Engineering and Information Technology (2018)
9. King, S., Nadal, S.: Ppcoin: Peer-to-peer crypto-currency with proof-of-stake (2012)
10. Kwon, J.: Tendermint: Consensus without mining (2014)
11. Lampson, B., Sturgis, H.E.: Crash recovery in a distributed data storage system. Tech. rep., Xerox Palo Alto Research Center (Jan 1979)
12. Li, Z., Kang, J., Yu, R., Ye, D., Deng, Q., Zhang, Y.: Consortium blockchain for secure energy trading in industrial internet of things. IEEE transactions on industrial informatics 14(8), 3690–3700 (2018)
13. Münsing, E., Mather, J., Moura, S.: Blockchains for decentralized optimization of energy resources in microgrid networks. In: 2017 IEEE Conference on Control Technology and Applications (CCTA). pp. 2164–2171. IEEE (2017)
14. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008)
15. Pontiveros, B.B.F., Norvill, R., State, R.: Monitoring the transaction selection policy of bitcoin mining pools. In: NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium. IEEE (2018)
16. Posdorfer, W., Kalinowski, J., Bornholdt, H., Lamersdorf, W.: Decentralized billing and subcontracting of application services for cloud environment providers. In: International Workshop on Engineering Service-Oriented Applications and Cloud Services – WESOACS. Springer International Publishing (2018)
17. Pudjianto, D., Ramsay, C., Strbac, G.: Virtual power plant and system integration of distributed energy resources. IET Renewable power generation 1(1), 10–16 (2007)
18. Stübs, M.: It-security in self-organizing decentralized virtual power plants: student research abstract. In: Proceedings of the 33rd Annual ACM Symposium on Applied Computing. pp. 431–432. ACM (2018)
19. Wood, G.: Ethereum: A secure decentralised generalised transaction ledger (2018)
20. Zhang, Y., Rahbari-Asr, N., Duan, J., Chow, M.Y.: Day-ahead smart grid cooperative distributed energy scheduling with renewable and storage integration. IEEE Transactions on Sustainable Energy 7(4), 1739–1748 (2016)