Demand Side Management using Blockchain for Distributed Networks

---

Aditya Pappu

January 20, 2021

Universiteit Twente

UNIVERSITY
OF TWENTE.

Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)

Master's Thesis, M-SET

Demand Side Management using Blockchain for Distributed Networks

Aditya Pappu

Chair Prof.dr. J.L. Hurink
EEMCS
Universiteit Twente

Supervisor Dr.ir. G. Hoogsteen
EEMCS
Universiteit Twente

External Dr. J. Popovic
EEMCS
Universiteit Twente

January 20, 2021

Aditya Pappu
Demand Side Management using Blockchain for Distributed Networks
Master's Thesis, M-SET, January 20, 2021
Supervising Committee: Prof.dr. J.L. Hurink and Dr.ir. G. Hoogsteen and Dr. J. Popovic

Universiteit Twente
Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)
Drienerlolaan 5
7522 NB, Enschede, The Netherlands

Abstract

With recent advances in renewable energy sources and energy storage devices, energy distribution networks are transforming from centralized one-way networks into distributed bi-directional mesh networks. This results in the challenge of implementing Demand Side Management (DSM) algorithms over distributed energy networks. DSM algorithms that were implemented in centralized energy networks with fixed authority structures must now be implemented over networks with no such centralized decision-making authority. All actors in a distributed network have the same authority and must arrive at a consensus regarding the global network-wide solution.

> *This thesis puts forward a vision for the communication layer in a distributed energy network as a complete blockchain network.*

Each prosumer in the communication layer is modeled as a full node within a blockchain. Two methods to implement DSM approaches in a distributed manner in such a blockchain network are proposed. We refer to these two methods as the *Method 1 implementation* of distributed DSM and the *Method 2 implementation* of distributed DSM. Each of these two implementation methods of distributed DSM is paired with two consensus protocols from blockchain, Proof of Stake (PoS) and Proof of Work (PoW). This results in four approaches to distributed DSM using blockchain. The two implementation methods of distributed DSM presented in this thesis may be generalized to any DSM approach. The Profile Steering (PS) approach by Gerards et al [1]. is used as the DSM algorithm in this thesis. Profile steering is an elegant decentralized DSM algorithm that does not require detailed knowledge of the network's topology and can achieve peak-shaving at each level of the grid.

The four approaches to distributed DSM using blockchain are simulated using Python 3 with a public residential dataset consisting of 25 houses from New York and their power profiles for 184 days. A comparative analysis of the four approaches on the basis of parameters such as time performance, security, scalability and energy consumption is done. We also analyze the network difficulty parameter in PoW based consensus and show how it can be configured to make PoW based consensus faster or slower than PoS based consensus. An analysis of different stake recovery models in PoS based consensus as a result of four different combinations of nodal

DSM performance and nodal ownership stake in the network is also presented. A discussion comparing the four approaches on the basis of criteria such as Byzantine Fault Tolerance (BFT) and Crash Fault Tolerance (CFT) is also presented.

The Method 2 implementation of distributed DSM is found to consume 42.17% less energy and is 64.98% slower than the Method 1 implementation of DSM. The Method 2 implementation of distributed DSM is found to be more scalable in terms of run time with increasing network size than the Method 1 implementation. Of the two consensus protocols, PoW is found to be more scalable in terms of run time with increasing network size as compared to PoS. On the other hand, PoS based consensus gives more options for DSM-based nodal incentives than PoW based consensus. We conclude by presenting the approach combining the Method 2 implementation of distributed DSM with PoS based consensus as a strong candidate for DSM in (remote) micro-grids.

Acknowledgement

I would like to express my sincere gratitude to the following people who have, with their guidance and support, made this endeavour possible.

To Dr.ir. G. Hoogsteen, EEMCS, Universiteit Twente,
Thank you for giving me this wonderful opportunity to work at the cross section of energy networks and blockchain technologies. Your constant guidance and availability and your patience with me (*and* my writing) has kept me motivated. Thank you for the many enlightening discussions we had regarding blockchain and for encouraging me to pursue my research interests in this topic.

To Prof.dr. J.L. Hurink, EEMCS, Universiteit Twente,
Thank you for your support and guidance and encouraging me to share my findings with the research group for feedback.

To my friends,
Thank you for your constant support, encouragement and for tolerating my blockchain banter over the last seven months.

To my family,
Last but never the least, thank you for your support, patience and love.

Contents

List of Figures

List of Tables

Acronyms

BFT Byzantine Fault Tolerance.

CFT Crash Fault Tolerance.

DER Distributed Energy Resources.

DG Distributed Generation.

DRES Distributed Renewable Energy Sources.

DSM Demand Side Management.

MG Micro-Grids.

P2P Peer-To-Peer.

PoS Proof of Stake.

PoW Proof of Work.

PS Profile Steering.

SG Smart-Grids.

THV Target Hash Value.

Blockchain Glossary

Blockchain

A sequence of connected blocks. Each block's hash value is derived from the previous block's hash value. Therefore each block is connected to the previous block resulting in a chain of blocks. *Explained in detail in Chapter 4*

Blocks

A block is a record or collection of the latest transactions that have occurred in the network. Transactions are recorded into a block in chronological order. *Explained in detail in Chapter 4, Subsection 4.2.2*

Distributed Ledger Technology (DLT)

A blockchain is a ledger or a record of all the transactions that have occurred in the network. A distributed ledger system or technology is a network in which all participating nodes have a complete copy of this network ledger. All nodes have access to the entire history of transactions that have taken place in the network. *Explained in detail in Chapter 4, Section 4.2*

Block Index Number

Each block in a blockchain is assigned a sequentially increasing number. The first block is assigned the number 0, the second block is assigned the number 1, the third is given 2, and so on. This number gives an easy way to refer to a particular block in the entire blockchain. *Explained in detail in Chapter 4, Subsection 4.2.2*

Hash Value

Each block has a hash value. A hash value is calculating by passing the block content (list of transactions) and previous block's hash value through a hashing function. The result is a alphanumeric string. *Explained in detail in Chapter 4, Subsection 4.2.2*

Mining or Forging

The action of calculating a new hash value for the current block derived from the block's contents and the previous block's hash value is called mining or

forging. Only after a block is mined is it officially part of the blockchain. *Explained in detail in Chapter 4, Subsection 4.2.3*

Genesis Block

The Genesis Block is the first block in the blockchain. This block has a block index number of 0. This block is created by the Network Manager and doesn't have any transactions in it. The purpose of this block is to give a starting hash value (a starting point) for the successive sequence of blocks to built upon. *Explained in detail in Chapter 4, Subsection 4.2.2*

Smart Contract:

All nodes sign a contract before entering the network. This 'smart' contract represents the terms and conditions that nodes must follow, written in the form of code. If a node breaks a certain rule, the penalty it incurs would be written in the smart contract. *Explained in detail in Chapter 4, Subsection 4.2.4*

Network Manager:

The Network Manager holds and enforces the Smart Contract. The Network Manager cannot make changes to the smart contract and its contents are transparent to all nodes at all times. The Network Manager also creates the Genesis Block. The Network Manager only moderates the network, making sure the nodes follow the very rules they agreed to in the Smart Contract. *Explained in detail in Chapter 4, Subsection 4.2.4*

Consensus Protocols

For every new block, the network must arrive at a consensus as to which node gets to mine that block. Blockchain networks use different consensus protocols to achieve this. A consensus protocol is essentially a competition or algorithm that nodes participate in and the winning node gets to mine the new block. *Explained in detail in Chapter 4, Subsection 4.2.7*

Proof of Work (PoW):

PoW was the first consensus protocol to be used in blockchain networks. In PoW, nodes are given the same mathematical problem to solve. The node that solves the problem first, wins the block mining rights. *Explained in detail in Chapter 4, Section 4.3*

Target Hash Value (THV)

In PoW, nodes are challenged to calculate a hash value for the current block that is lesser than or equal to a Target Hash Value set by the network manager.

The node that achieves this first wins the consensus round. *Explained in detail in Chapter 5, Section 5.4.1*

Proof of Stake (PoS):

PoS is another consensus protocol used in blockchain networks. In PoS, each node deposits an initial stake into the network when entering the network. The winning block in each consensus round is chosen by probability. Higher is the initial stake of a node, higher is its chance of winning block forging rights. *Explained in detail in Chapter 4, Section 4.4*

Stake Recovery

When a node wins the PoS round, it receives a PoS reward from the network manager. This is the incentive to win the PoS round and is called Stake Recovery. The Total Stake Recovery of a node is the total of all the PoS Rewards earned by it. *Explained in detail in Chapter 6, Section 6.5*

Byzantine Fault Tolerance (BFT)

Blockchain is a distributed network. Any node can join such a network and all nodes have the same privileges. In such a situation, a node may choose to sabotage the consensus round and win block mining rights to be able to mine fraudulent blocks. The capacity of blockchain consensus protocols to disincentivize any such malicious behaviour on the part of the nodes is called Byzantine Fault Tolerance. *Explained in detail in Chapter 7, Section 7.1.2*

Crash Fault Tolerance (CFT)

A blockchain network is a P2P mesh network and certain operations may require communication between nodes or between the network manager and the nodes. If a node or a group of nodes crashes or disappears off the network abruptly, this can affect the outcome of the operation (a DSM algorithm for example). A network's ability to have safeguards or provisions in place for such a situation is known as its Crash Fault Tolerance. *Explained in detail in Chapter 7, Section 7.1.1*

1 Introduction

Chapter Objective: This chapter present a brief history of energy networks and explains some important concepts in the domain of energy networks that shall formulate the basis for the work presented in future chapters.

Chapter Contents

1.1 The first DC and AC grids

Electric power first saw commercial use via small scale DC systems. These systems were used for mining and industrial reasons, and were considered an expensive luxury for domestic use. The development of the DC electric grid was led by Thomas Edison, Charles Brush, and Werner von Siemens [2]. In 1882, Edison effectively launched the electric utility industry by constructing the Perl Street electrical Power Station [3]. The Perl Street power station was the first to deliver many features such as reliable central power generation, safe and efficient distribution and a successful end use (Edison's long-lasting incandescent light bulb), which are now considered as default characteristics of an electrical grid. The Perl Street Station, achieved many firsts for the electrical grid as we understand it today, such as building a network of underground wires and cables or "conduits" and devising a way to track electricity consumption over time for every costumer, thereby being able to bill the customer for only how much they had consumed. It was a

chemical ampere-hour meter and unfortunately, was immensely inaccurate and messy and later, in the AC grids to come, it would be replaced with Oliver Shallenberger's AC ampere-hour meter [4].

DC power was low voltage and could not be transmitted over long distances, leading to very small centralized power stations and patches or 'islands' of regions between theses stations not having any light. In a way, the very first DC grids, were examples of 'islanded grids'; independently-functioning grids that provide power over a limited geographical region and are cut-off from any main or other grid. Such an islanded nature of operation was, in those days, due to a lack of the necessary technical innovation in power transport technology and not by conscious design.

In 1886, George Westinghouse set up the first single phase AC power system at Great Barrington, Massachusetts lighting up all of downtown. The design was based on stepping up 500 volts to 3000 volts during transmission eventually stepping it down to 100 volts for domestic use and was based on William Stanley's famous transformer design [5], setting off the historically famous "War of the currents" between Edison and Westinghouse. By 1893, this 'war' was settled in the favour of the more efficient AC power system, with Westinghouse underbidding Edison's General Electric to power the 1893 World's Fair Chicago. This gave birth to the initial AC electrical grid system that became the technological backbone for the present-day version [6].

1.2 Distributed Generation (DG)

During the 20th century, a one-way centralized grid with fixed points of mass electricity generation was the norm. These centralized grids were powered by huge power plants that ran on diesel and coal. Once power was generated by such carbon-intensive energy sources, it was transmitted to the those connected to the electrical grid, the members of which played the singular role of a fixed consumer using the power as required and paying a proportional amount to the energy provider.

But with the increased proliferation of Distributed Energy Resources (DER)s, a more complex electric grid network starting emerging [7]. An increase in demand on centralized utility grids leading to electricity shortages, power quality problems, rolling blackouts and electricity price spikes coupled with

increasing innovation and development of renewable energy sources, has led to a greater interest in DG [8].

The Institute of Electrical and Electronics Engineers Inc. (IEEE) [9] defines DG as:

> "*The generation of electricity by facilities sufficiently smaller than central plants, usually 10 MW or less, so as to allow interconnection at nearly any point in the power system*".

.

Distributed energy systems, also take advantage of storage solutions such as batteries and fuel cells to store the locally generated energy and ease the demand on the main grid [10]. While centralized generation suffers from environmental problems such as land use, water user and discharge and waste generation, distributed generation on the other hand, can provide clean, reliable energy power generated at or near the source, minimizing transport losses too [11]. DG usually comprises of energy sources such as solar arrays, wind turbines, combined heat and power generation, backup generation and storage systems, located close to the load [12].

As shown in Figure 1.1, the energy generated from Distributed Renewable Energy Sources (DRES) in the Netherlands increased by 10% from 15 billion kWh (2016) to 17 billion kWh (2017). The share of electricity consumption in The Netherlands covered by DRES went from 12.5% 2016 to 13.8% in 2017.

The concept of distributed generation, made possible due to increasing innovation in DRES, has led to the present-day idea of micro-grids. According to Nazari-Heris, Madadi and Mohammadi-Ivatloo [14], Micro-Grids (MG) are a

> "*combination of Distributed Energy Resources (DER) units and Energy Storage Systems (ESSs) which can be used either in a grid-tie or as a purely standalone network*".

When operated in a standalone mode, such MGs are also known as remote micro-grids.

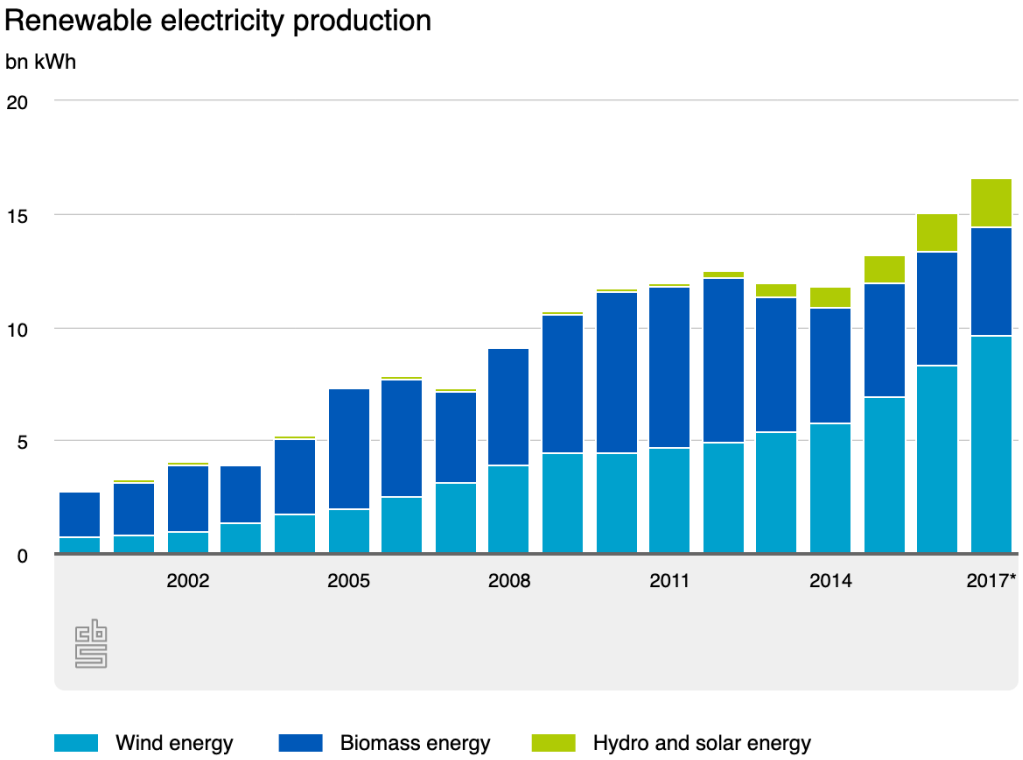


Fig. 1.1: Energy production by DRES in The Netherlands from 2000 to 2017[13]

1.3 Smart Grids (SG)

The rise of decentralized grids leads to the electricity grid becoming more complex to manage. ICT services can aid in this by enabling real-time monitoring of demand and supply and optimal grid management. According to the Netherlands Office of Science and Technology,

> "*Electricity grids that can optimize the management of the electricity network and support the coordination of the local energy market by means of advanced ICT services are called Smart-grids [15].*"

Smart grids are networks that can seamlessly integrate all aspects of an electric grid - generators, consumers and entities that play both roles to create a sustainable energy system that can transmit power in an economically feasible manner at high quality and with low losses [16].

As per a 2018 report by the US Department of Energy [17], in addition to deploying and incorporating distributed energy sources into the network, smart-grids also have the following main characteristics:

1. Dynamic optimization of grid operations and resources.
2. Cyber-security for ensuring privacy of grid data.
3. Increased use of automation, real-time monitoring and control technologies, and digital information to make the grid more efficient, economical and safe.
4. Integration of energy storage and peak-shaving technologies, including plug-in electric and hybrid vehicles.
5. Provision to consumers for receiving timely information and control options.

1.3.1 SG as P2P networks

SGs envision the electrical network as a Peer-to-Peer (P2P) network, comprising of nodes performing one or many roles such as electricity generation, transmission, consumption or even monitoring. SGs have triggered the *democratization of the energy market*. With the increased proliferation of super-fast bi-directional communication networks, traditional consumers can now add their own DRES onto the grid and sell the excess energy to the grid. This has led to the traditional uni-directional flow of energy between producers and consumers to become transformed into a multi-directional mesh-like model involving prosumers as network and grid actors. SGs allow residential households to become energy producers and sellers and thus spawn local energy markets and deliver increased transparency over one's own energy supply and consumption. Figure 1.2 visualizes the transformation in energy networks from centralized power generation and one-way transmission to distributed generation and a mesh network of prosumers.

1.4 Case for remote MGs

1.4.1 Energy Access

Energy access is a critical problem today affecting up to an estimated 1.2 billion worldwide who still live without electricity. From a mere 20% in 1995,

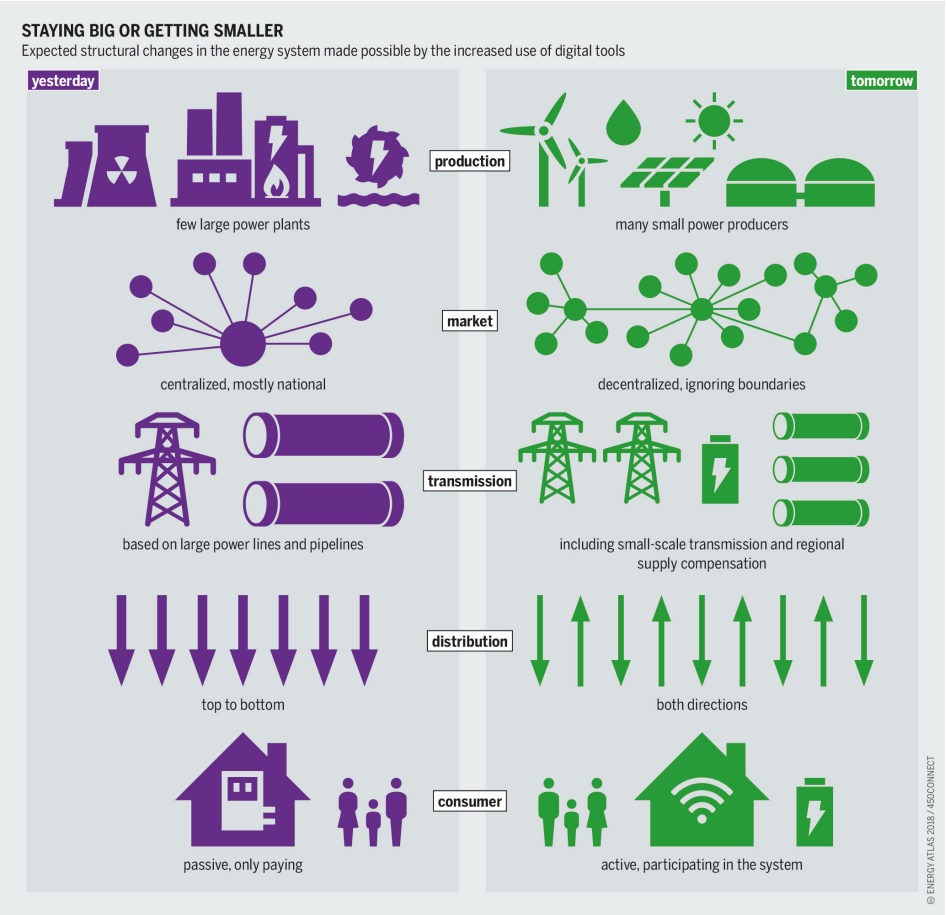


Fig. 1.2: The transition in grid architecture from central and one-way networks to decentralized mesh networks [18].

the current figure in sub-Saharan Africa has risen to 40%, which though a commendable rise, still indicates that up to 600 million people still live without access to electricity, as shown in Figure 1.3. These 600 million people represent two-thirds of the global population.

The lack of energy access leads to 'Energy Poverty', a situation wherein the daily well-being and economic prosperity of individuals living in developing countries and some developed countries is negatively affected by the lack of a stable electricity supply. A lack of electricity negatively affects the basic standard of living of the people in the community, their economy, and stunts the growth of that community.

There's also an environmental aspect to this. In most cases, the regions most deprived of energy access are generally geographically far off from urban centers or power generation centers. When governments and utility

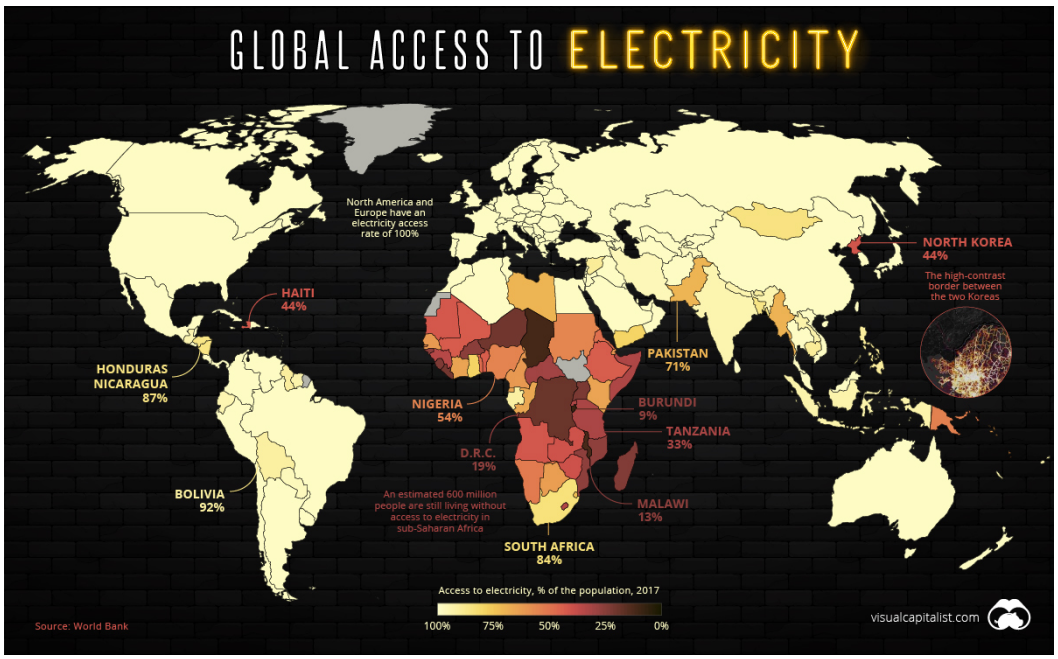


Fig. 1.3: Global map of % electricity access in 2019 [19].

companies out of good intentions, try to extend the central grid till such regions, it leads to an increased dependence on fossil fuels for both grid electricity and power generation, leading to increased levels of pollution [20]. The population of South Asia is exposed to the world's largest combustion-related concentrations of $PM_{2.5}$, the most harmful of the toxic pollutants released as a result of coal and diesel combustion. The average population weighted concentrations in Bangladesh, India and Pakistan, standing at 89, 74 and 65 micro-grams per cubic meter respectively, are many times higher than the World Health Organization's safe limit of 10 micro-grams per cubic meter. All this because of the increased consumers brought onto the main-grid (which is generally powered by fossil fuel based energy generation) and due to the energy intensive process of laying down AC cables over extremely long distances over geographically intensive terrains.

Remote MGs, given their easier initial setup and flexible payment models (as compared to extending the central grid), are often regarded as the solution to the problem of Energy Access. In the case of sub-Saharan Africa, the International Energy Agency (IEA) concluded, after detailed geo-spatial modelling, that decentralized systems such as off-grid PV-based systems and mini-grids are the most cost-effective solution for providing three quarters of the additional electricity connections required [21]. IRENA estimates

that about 350,000 mini-grids will be needed to achieve universal energy access in Africa, of which 60,000 shall be deployed in West Africa alone, with DRES being the default choice for stand alone micro-grids in rural and peri-urban regions [22]. This boost in mini-grids and micro-grids is credited to a decrease in the cost of manufacturing DRES, especially PV systems and massive technological advancement in domains such as communications, energy storage and control systems.

1.4.2 Security of Supply

In cases where the central grid supply exists, an unstable grid, can also have negative consequences over multiple domains of a community such as the economy and the entrepreneurship sector. For example, a study done by the Global Center for Development in 2019, showed that 57% of surveyed firms ranked "poor electricity" as a major hurdle to doing business [23]. As shown in Figure 1.4, majority firms in Nigeria reported up to 30 or more power outages per month of an average length of 3 hours.

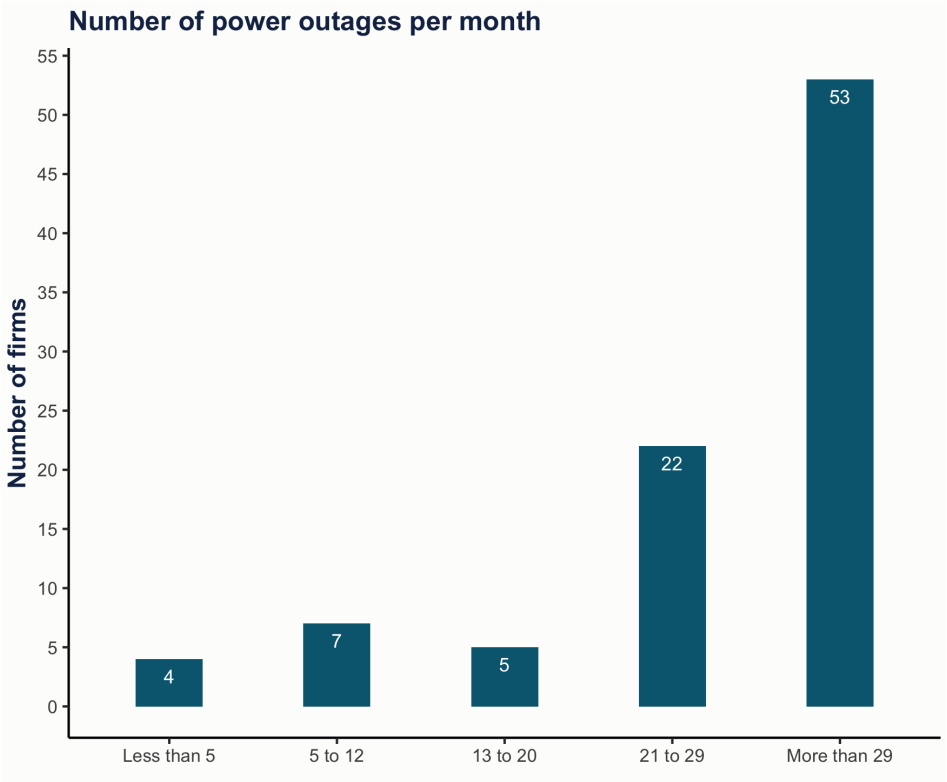


Fig. 1.4: Reported power outages by tech-firms in Nigeria in 2019. [23].

Given such frequent power outages, companies and firms lose money in due to lost sales and due to private investment in generating electricity, which in turn, are likely to be from conventional carbon-intensive sources. For example, MTN, one of Africa's largest mobile network operators, spends about 70% of its annual operation expenditure on buying and consuming up to 10 million litres of generator fuel per month. [24]. According to Shuaia et al., power outages have both direct and indirect impacts on communities [25]. Direct impacts relate to the cost of the total load lost during the outage. Take for example a power-outage caused on February 4th 2011 in the north-eastern grid in Brazil, which spread to 8 states and lead to a power loss of 8000 MW and a financial loss of $60 million [26].

The concept of having a stable guaranteed supply of electricity is called 'Security of Supply'. According to a 2012 report on SGs done by the Science and Technology Options Assessment (STOA) organization [27], security of supply is one of the critical benefits that SGs provide. An unstable flow of power results in a negative cash-flow, which researchers studied under the term 'Value Of Lost Load (VOLL)'. A set of studies done by Frontier Economics in 2008 showed that the average cost of supply interruption is around €10/kWh. In another example, the study shows that if the quality of electricity supply in Germany were to fall to Spanish levels, losses to the economy would amount to €1500 to €3200 million per year. As the report states,

> "*The conclusion from this string of research is that technologies, which help to avoid power outages, have a much greater value from the macroeconomic point of view than the purchase price of electricity. If smart grids manage to contribute considerably to stabilizing the grid in feeble or "island" networks, the investment will pay off quickly in macroeconomic terms [27].*"

MGs provide the critical advantage of a safe, reliable and efficient energy infrastructure. They can sense impending power outages and solve the outages locally before they impact either the local grid or the main grid (grid-tie mode). MGs can aid businesses and communities from a security standpoint by protecting the grid from external natural threats such as weather emergencies or man-made attacks such as cyberthreats. The conventional energy infrastructure is highly inefficient due to its ageing nature and the fact that the energy generation sources are situated far away from the communities

they serve. According to Metcalfe [28], the overall conversion efficiency from primary energy to delivered work is just 33% in the US. Two-thirds of the energy consumed as primary energy is released as waste. He argues, that there are two reasons responsible for this waste. One reason is the burning of fossil fuels to generate energy. Another reason is the energy consumed in generating and transmitting electric power to the end user. MGs often employ DRES as local sources of energy, therefore running at much higher efficiencies with much less distances for the energy transmission to cover and result in lesser load losses.

Islanded MGs can also provide the benefit of better environmental resilience by 'hardening' the grid against natural disasters. For example, the North County of Upstate New York is exposed to storms of increasing frequency and intensity. In the common scenario that a storm knocks down the main supply, it is imperative to have a contingency supply of electricity for the smooth running of essential services so that the region can be restored to normalcy as soon as possible. To achieve this, National Grid, a private energy company, is setting up a micro-grid that during normal operation connects to the centralized grid (macro-grid) but during storms, can disconnect from the main grid and operate in purely islanded mode as a separate independent network providing power for essential services. The cabling for this MG is placed underground as opposed to the main-grid's overhead cables, thus achieving protection from natural elements[29].

1.4.3 Grid Democratization

With the advent of ICT services and DRES, consumers can now play the role of micro-utilities and supply excess energy to the grid. This puts the economic power in terms of power purchase and sales in the hands of traditional consumers (now prosumers). This has led to the emergence of local energy markets that encourage home owners with DERs and Energy Storage Devices (ESDs) installed to trade excess energy onto the grid [30]. This energy market and trading can be within the remote micro-grid community with other local prosumers and/or with the main grid, in a grid-tie system. This transforms the nature of 'authority' in the energy network to a more horizontal one. With the incorporation of wireless sensor networks consumers now can get

real-time information about their consumption, production and the price of electricity.

Additionally, MGs that boast of transparency and accountability, both in the aspects of security of supply and payments, can harness emerging technologies such as decentralized DSM protocols and concepts from blockchain to achieve the same. Blockchain protocols eliminate the central monopolistic entity and make every prosumer an equal participant in the decision making system.

> *Such decentralized, distributed and democratized islanded MGs function as horizontal mesh like P2P networks, are modular to setup, rely on DERs and ESDs, the prices of which are decreasing rapidly, are resilient to environmental conditions, incorporate advanced ICT services and DSM protocols to achieve security and transparency of supply and can also, with blockchains and distributed ledgers give a high degree of power guarantee and financial accountability amongst grid prosumers.*

1.5 Central, Decentralized and Distributed Networks

The communication infrastructure employed in the energy networks of MGs and SGs is essentially a mesh network. When multiple actors or entities are involved in a single network, this requires an understanding of the network architecture. Before delving into distributed approaches to DSM, we must first understand the three different types of network architectures. Broadly speaking, all digital networks are classified into three categories based on three aspects.

- *Authority*: Who has the decision-making authority in the network?
- *Data Transparency*: Who has access to the (transaction) data generated in the network?
- *Nodal Communication*: What is the extent of inter-node communication in the network?

1.5.1 Centralized Networks

- *Authority*: All decision making authority lies with one central node. Child nodes participating in the network do not engage in any computing. They may share data with the central node but the central node does all the computing.
- *Data Transparency*: Access to data generated within the network is centralized. The central authority node stores all the data generated in a private restricted database and nodes at best have (some or full) access to their own data.
- *Nodal Communication*: There is no communication between child nodes. All communication in the centralized network is between a child node and the central authority node.

1.5.2 Decentralized Networks

- *Authority*: Network authority is divided amongst multiple parent nodes. There is no focus of authority in the hands a single central node. Each parent node holds decision making authority over a set of child nodes. Child nodes may make some decisions by themselves but major decisions are still made by their respective parent node.
- *Data Transparency*: All transaction data generated in the network by child nodes are stored within their respective parent nodes. Child nodes may have some access to just their own transaction data.
- *Nodal Communication*: There is no (or minimal) communication between nodes. However all primary communication for a child node is with its parent node.

1.5.3 Distributed Networks

- *Authority*: There is no central authority node in the network. All nodes have equal decision making privileges and all nodes perform the same computational tasks.

- *Data Transparency*: All network transaction data may be accessible to all nodes. Some nodes may choose to keep their own transaction data private, but nodes at least enjoy much higher data transparency than in a centralized network.
- *Nodal Communication*: Since there is no central authority, all communication happens between network nodes. Some distributed networks may still give nodes the option to create private channels of communication. Distributed networks may be imagined as the result of taking decentralized networks to their democratic extreme.

Figure 1.5 shows the configuration of all three network types. Throughout this thesis, we refer to actors or entities in a network as nodes.

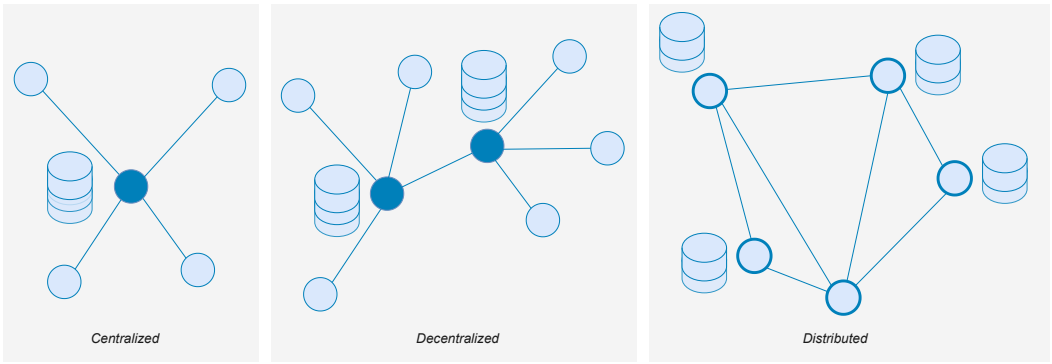


Fig. 1.5: Three types of network structures

1.6 Research Questions

The communication infrastructure of energy grids is transforming into a distributed network. Each prosumer in the energy grid represents a node in this distributed network and is capable of generating, buying, storing or selling energy to the grid. This makes it necessary to take DSM approaches built for the one-way communication layer of past electric networks with rigid authority roles and investigate how they can be transformed to be applied to a distributed communication layer. The most popular example of distributed networks that has found considerable commercial use is blockchain. While blockchain has typically been used in the domain of finance and cryptocurency, it is finding wider adoption in other domains of industry as

well. Blockchain technologies can aid in implementing DSM approaches in distributed networks with no differential distribution of authority.

1. **Main Question:** How can demand side management be implemented in distributed networks?
2. **Research direction:** How can concepts from Distributed Ledger Technologies (DLT) such as blockchain aid in this objective?
3. **Implementation:** Can a DSM algorithm such as profile steering [1] be implemented using DLT concepts in a distributed network?
4. **Evaluation:** What would be the result of a comparative analysis between the resultant approaches towards implementing DSM in a distributed network using blockchain?

1.7 Thesis Flow

Chapter 2, presents of review of some approaches to implementing DSM in distributed energy networks and important inferences derived from these approaches.

Chapter 3, explains the working of a decentralized DSM algorithm called profile steering. This DSM algorithm is implemented in a distributed manner in future chapters.

Chapter 4, introduces the concept of blockchain and provides an introductory explanation of some critical concepts used in blockchain networks.

Chapter 5, uses the blockchain concepts introduced in the previous chapter to propose two methods to implement distributed DSM in an energy network paired with two consensus protocols from blockchain. This results in four approaches to implementing distributed DSM using blockchain in energy networks.

Chapter 6, presents simulation results of all four proposed approaches to perform a comparative analysis amongst the approaches on the basis of parameters such as run time, scalability and energy consumption.

Chapter 7, compares all four approaches on the basis of parameters such as network security and nodal incentives. Finally, a concluding reflection of the main research question and some points for future research are provided.

Related Work

2

Chapter Objective: The objective of this thesis is to develop approaches for implementing a DSM algorithm in a distributed manner in an energy network and to achieve consensus amongst nodes of that network. Therefore, in this chapter, a literature review in the fields of decentralized demand side management approaches and decentralized consensus in energy networks is covered in this chapter.

Chapter Contents

2.1 Decentralized DSM approaches

As more distributed energy sources and storage systems get added to the grid, energy management becomes an important aspect to consider. Energy management approaches may be either centralized or decentralized. In centralized approaches, all decision-making is done by a central node and the other nodes follow its decisions. In decentralized approaches, the decision making computation is divided amongst multiple child nodes and aggregated by a coordinator node. Given below are five decentralized DSM approaches.

Distributed optimization based on consumer types and differential tariffs: Longe et al. [31] propose a Microgrid Energy Management Distributed Optimisation Algorithm (MEM-DOA) model wherein, network nodes are placed into four categories based on their energy characteristics. Type-A nodes are passive and buy all their energy from the grid, Type-B nodes have an Energy Storage System (ESS), Type-C nodes only have a DER and Type-D nodes have both an ESS and a DER. Each of these nodes is allowed to

optimize its own load profile, a process which is in turn influenced with different financial incentives and tariffs offered to each node based on its node type. Essentially, the MEM-DOA model envisions a MG as an energy market comprising of these 4 types of nodes and uses financial incentives to incentivize energy transactions between these nodes with the aim to achieve a smooth aggregate load profile. The model achieved a 68% reduction in aggregate peak demand. However it relies on a strict classification of network nodes into the four categories.

Battery Energy Storage controller approach: Jha and Kumar [32] propose to use batteries for DSM in an MG. They employ a dedicated Battery Energy Storage (BES) system controller that is designed to tackle voltage instability and the sudden demand of load. The BES controller algorithm is designed to release power from the batteries as and when regulated by a Voltage Control Unit and a Current Control Unit. The control scheme of the BES system is based on the voltage generated by the DER in the MG and the state of charge of the battery. The model is able to achieve about 18% saving on active power and 16% saving on reactive power in MATLAB simulations of a stand-alone MG.

Multilayer Ant Colony Optimization: Marzband et al. [33] propose an energy management system based on the Multilayer Ant Colony Optimization (MACO) algorithm. They place an emphasis on using the efficiency of the Ant Colony Optimization (ACO) algorithm to solve performance optimization problems, achieve improvement in DER scheduling and cost reduction of system performance in an MG. ACO is a common optimization algorithm that is modeled after the behaviour of real ants searching for food and building the shortest path from their nest to the food source. Marzband et al. propose that MACO achieves better results in fewer iterations at lower energy cost with respect to the traditional Particle Swarm Optimization (PSO) algorithm and Modified Conventional Energy Management System (MCEMS) both.

Game Theoretic approach and Blockchain trading: A decentralized DSM approach based on the principles of game theory is proposed by Noor et al [34]. Each house is treated as an independent player in a non-cooperative game. The aggregate load profile is divided into N local profiles for each of the N houses in the network. Each house then attempts to optimize its local profile with respect to its ideal local profile. Houses compete to optimize their local profile and earn proportional financial awards in return. For the

financial transactions, the paper proposes using blockchain technology given its capacity for security and transparency. The suggested model is tested in a decentralized network with all houses sending their locally optimized load profiles to a central server for aggregation and calculation of payback values. The model achieves a 46% and 29% reduction in the Peak-to-Average Ratio (PAR) and a 22% and 19% reduction in the electric bill for test models with and without energy storage elements respectively. The houses conduct the financial transactions by trading PowerCoins over a ZipLedger blockchain network. These PowerCoins are custom digital tokens that are assigned a monetary value and ZipLedger is a Blockchain as a Service (BaaS) on which the smart contracts for trading are run.

Profile Steering: An elegant solution that uses flat power profiles as steering signals instead of price steering was proposed by Gerards et al. [1]. With price steering, prosumers are most likely to optimize their energy profile in a way that maximizes financial payback. This means that the highest consumption is planned for the time slots with lowest tariffs or lowest cost of electricity. This can lead to more voltage peaks, in terms of number and intensity. In the profile steering approach, an energy network is envisioned as a network of nodes assigned to different levels in a top-down hierarchy. The nodes at Level 0 are the child nodes for the coordinator node at Level 1 (Level 1 is above Level 0). The coordinator node sets a desired aggregate load profile for the day and requests each child node to send a local profile. The coordinator node chooses the child profile that has the maximum improvement. Improvement in this case is determined by calculating the Euclidean distance between the aggregate load profile and the desired profile before and after the child nodes submit their new optimized profiles. The child node profile with maximum improvement is chosen and the other candidate profiles are discarded.

Multiple such iterations are executed. The coordinator node tries to minimize the Euclidean distance and steer the aggregate load profile of the child nodes closer to the desired profile. Gerards et al. tested this approach via simulations on 121 houses. Each house owned an EV whose power consumption could be controlled and shifted in time. A comparative analysis between profile steering, uniform pricing and no control was performed. The profile steering approach resulted in an aggregate load curve that stayed between 220V and 235V and the distribution losses were reduced by 57% when compared to a simulation with no control (no DSM approach) and 48% when compared to a uniform price steering approach.

Profile steering is an elegant and hierarchical approach to DSM that aims at peak-shaving at each level of the grid. At the same time, profile steering does not require a detailed knowledge of the grid topology. This thesis uses profile steering as the DSM algorithm.

This DSM approach is explained in detail in Chapter 3.

2.2 Blockchain in Energy Networks

2.2.1 Blockchain in DSM

Decentralized ADMM using a virtual aggregator: Münsing, Mather and Moura [35] use the Alternating Direction Method of Multipliers (ADMM) algorithm to solve the Optimal Power Flow problem (OPF) to schedule a set of batteries, deferrable loads and shapable loads in an energy network. ADMM is an approach that allows a complex global problem to be broken down into multiple local sub-problems who's local solutions are then coordinated by a central aggregator. Münsing at al., formulate the global problem and then divide the global problem using ADMM among the child nodes in the network. Each node finds the best fit for it's local problem (hence a local solution). The central aggregation is then done by a smart contract (the virtual aggregator) hosted on a blockchain network. Each house or building's smart meter acts as a computational node on a blockchain network (and as a child node in the ADMM algorithm). The smart contract acts as the coordinator node. The model also proposes using the immutable nature of a blockchain for the energy billing part, which allows for more transparency, more security and stronger accountability in terms of penalties to nodes for shifting from their promised load profile.

This model was tested on a simulated SCE-55 bus network with solar arrays placed at 60% and deferrable loads placed at 70% of the buses in the network. A private Ethereum Homestead Network was used to host the smart contracts. Münsing et al. hypothesize that the communication overhead and the verification required in blockchain may limit the use of such a model for day-ahead scheduling problems. The same model was tested by Alskaif and Leeuwe [36] using data from 23 households in the East Harbour Prosumers Community in Amsterdam and a 23 bus test network. Their test reported

that it took 300 iterations for the decentralized ADMM model to converge and this translated into a 0.8% higher cost than the centralized solution.

2.2.2 Blockchain in Energy Trading

Most applications of blockchain in energy networks lie in the domain of facilitating local energy markets by leveraging the secure aspects of blockchain networks such as distributed ledgers, transparency and public key cryptography. Blockchain is generally used for financial transactions such as power purchase and sale. Given below are some applications of blockchain for energy trading.

Power Ledger

Powerledger [37] is a peer-to-peer energy trading blockchain company (and platform) in Australia. The platform provides a market trading and clearing mechanism to owners of DERs to sell their surplus energy to other prosumers or to the grid. All financial transactions are recorded in a distributed ledger for security and transparency.

LO3 Energy

LO3 Energy [38] in Portland, U.S.A, is another company that offers a similar digital platform called 'Pando' to all prosumers in a local grid to enable each prosumer to purchase or sell energy using a secure blockchain based transaction network. A transaction manager (smart contract) is used for allowing custom marketplace rules and all transactions are recorded on a Digital Transaction Ledger (distributed ledger).

NRGCoin

NRGCoin [39], which began at the Vrije Universiteit Brussel in Belgium is based on the Ethereum blockchain network and encourages prosumers to trade their energy on the NRG Energy Network by awarding them NRGCoins in return. NRGCoins are a form of currency (like Bitcoin). One NRGCoin is always equal to 1kWh of energy. This constant ratio is encouraging for prosumers since they get the same financial payback for the same amount of energy sold regardless of price fluctuations on conventional fiat currencies. A smart contract is used to fix rules such as subsidies. This smart contract cannot be altered by a utility or by a prosumer. The smart contract eliminates the risk of policy change on DRES subsidies. For example, a subsidy for solar

panels written into the smart contract is immutable. This gives prosumers a strong guarantee for consistent financial rewards for injecting energy into the network. Now acquired by Enervalis [40] in Belgium, it is currently being embedded into their SmartPower Suite which will enable a full suite of energy trading and monitoring options for prosumers in an MG. At the moment Enervalis is working with Eemnes Energy on building the first large scale peer-to-peer energy trading network in The Netherlands based on blockchain. Upto 4000 households in the municipality of Eemnes can engage in energy trading using NRGCoins [41].

Open Charging Network (OCN)

The OCN [42] curated by the Share and Charge Foundation in Germany, is an example of a P2P energy network for EV charge point operators, eMobility Service Providers and EVs to participate in. It is based on the open de-facto standard for digital interoperability in EV charging, the Open Charging Point Interface (OCPI). OCPI is a protocol that describes communication relating to exchange of charge point information such as transaction events, charge detail records and smart charging commands and authorization of EV drivers. The OCN is a decentralized implementation of the OCPI protocol. Any e-mobility service provider or EV charging station provider can join the OCN as a node. EV owners can then use their services with all the financial transactions secured with blockchain.

2.3 Literature Review Table

Table 2.1 presents a summary of the literature studied and the main points derived from each.

Tab. 2.1: Summary of main inferences drawn from the literature study.

Beginning of Table

Author	Topic	Proposed model	Tests (if any)	Results	Points of interest
Longe et al. (2017) [31]	"Distributed Optimisation Algorithm for Demand Side Management in a Grid-Connected Smart Microgrid"	Divide prosumers into four different categories based on their characteristics. Use different financial incentives for each role to control energy behaviour of prosumers of each category.	Simulations with residential data and Time-of-Use pricing tariffs from four hundred consumers in South Africa.	Aggregate peak demand reduced by 68%. Aggregate PAR demand reduced by 46%.	Relies on dividing nodes into set roles. Also nodes must act in accordance to role-specific tariffs for successful DSM.
Jha and Kumar (2019) [32]	"Demand Side Management for Stand-Alone Microgrid Using Coordinated Control of Battery Energy Storage System and Hybrid Renewable Energy Sources"	Proposes a control mechanism design for Battery Energy Storage (BES) system using signals from voltage and current units based on a PI controller. BES system is used to compensate for peak demand.	Feasibility test of the BES module done in Matlab.	Achieves 18% decrease in active power and 16% saving in reactive power.	Relies on having a central BES reservoir dependent on control signals about the voltage and current in the entire grid.

Continuation of Table 2.1

Author	Topic	Proposed model	Tests (if any)	Results	Points of interest
Marzband et al. (2016) [33]	"Real time experimental implementation of optimum energy management system in standalone Microgrid by using multi-layer ant colony optimization"	Proposes a multi-layered version of the ACO algorithm. In each layer, ACO is run amongst the nodes in an energy network.	Simulations performed for a stand-alone wind turbine / photovoltaic / microturbine and energy storage system using simulators. Algorithm written in C and executed on a 2.53 GHz Core(TM) Duo P8700 personal computer with 4 GB RAM.	The MACO solution keeps more energy in the battery than conventional optimization solutions. The MACO solutions also reduces stress on the batteries with less frequent charging and discharging cycles.	The paper implements MACO in a decentralized manner. A coordinator chooses the most optimal 'route' or solution.
Noor et al. (2018) [34]	"Energy Demand Side Management within micro-grid networks enhanced by blockchain"	Nodes in an energy network participate in a non-cooperative game. Each house optimizes it's profile with respect to local requirement and earns financial awards in return.	Simulations done with residential data from 15 households. Test also took three types of households into account - families, working singles and college students.	Tests show a 46% and 29% reduction in the Peak-to-Average Ratio (PAR) and a 22% and 19% reduction in the electric bill.	A decentralized DSM algorithm that's very similar to Profile Steering.
Gerards et al. (2015) [1]	"Demand side management using profile steering"	Each house optimizes it's local profile with respect to a desired profile set at the neighbourhood level by a coordinator. Coordinator chooses local profile with best improvement. Multiple iterations are run to arrive at an optimal neighbourhood profile.	Simulations done on a three phase Dutch LV grid model with 121 houses. A steerable EV is attached to each house.	Results in 57% less losses with respect to no DSM control and 48% less losses with respect to uniform pricing.	Leads to better peak-shaving and better power quality with respect to other DSM algorithms. Most importantly algorithm doesn't require low level knowledge of the topology of the grid making it a promising candidate for DSM in energy networks.

Continuation of Table 2.1

Author	Topic	Proposed model	Tests (if any)	Results	Points of interest
Münsing, Mather and Moura (2017) [35]	"Blockchains for decentralized optimization of energy resources in microgrid networks"	Implement a decentralized version of ADMM amongst nodes in an energy network. Global solution is broken into local solutions for nodes to solve.	Digital simulations done on a SCE-55 bus network with randomly generated load profiles. 60% of the buses had solar arrays and 70% of buses had deferrable loads. Batteries placed on each bus with power capacity of 50% of peak controllable load and a 4 hour energy storage capacity.	ADMM algorithm converged in 204 iterations with 1.2s for each iteration. ADMM solution cost was 0.4% larger than the centralized solution.	The model implements ADMM in a decentralized manner by assigning the role of coordinator to the Smart Contract in a blockchain network. Authors caution against the increased delay and data overhead due to blockchain integration.
Alskaif and Leeuwe (2019) [36]	"Decentralized Optimal Power Flow in Distribution Networks Using Blockchain"	Model is the same as Münsing et al. Unique contribution was the test done using Amsterdam residential data.	Simulations done with residential data from 23 households from the East Harbour Prosumers Community in Amsterdam. Distribution: 8 prosumers with PV, 3 prosumers with PV and EV, 8 consumers and 3 consumers with EV. Each EV driven for 36km/day with driving efficiency of 5km/Wh (assumption). Model run on a private Ethereum blockchain network.	ADMM algorithm run for 300 iterations and ADMM model costs found to be 0.8% higher than centralized solution.	-

Blockchain in Energy

Powerledger [37] and LO3 Energy [38] both are P2P energy trading platforms that store transaction information in a blockchain in distributed ledgers.

While NRGCoin [39] also uses distributed ledgers to store transactions, it's unique factor lies in the NRGCoins mined and exchanged within the energy network and the fixed valuation of the NRGCoins.

OCN [42] is for e-mobility what Powerledger and LO3 Energy are for energy trading. OCN also uses blockchain to store the transaction information. Its key asset is that it is open-source and can be used by developers to test and deploy their e-mobility DApps (Decentralized Applications).

End of Table

In the next chapter we explain profile steering, a decentralized DSM algorithm. In Chapter 4 we introduce important concepts from blockchain and in Chapter 5, we propose two methods for implementing DSM over a distributed energy network and pair each with two consensus protocols from blockchain.

3 Profile Steering

Chapter Objective: In this chapter, we explain a specific decentralized DSM approach called Profile Steering (PS), which was first proposed by Gerards et al. [1]. Thereafter, profile steering simulations for a public residential data set are presented and the achieved results are discussed.

Chapter Contents

3.1 Introduction

The design of the electrical grid is changing from the conventional centralized grid paradigm to the decentralized paradigm due to the advent of DERs in the grid. With an increase in DERs, conventional Energy Management (EM) approaches initially used in centralized unidirectional grids do not work. New EM approaches are required which are scalable taking into account the dynamic nature of the electric grid. Additionally, centralized EM approaches require the data (load profiles, scheduling data) to be collected and worked upon in one single central server leading to increased bookkeeping, privacy issues and extreme computational requirements for that single central node.

New EM approaches must take an electric network and re-imagine it as a distributed mesh network. In a *centralized* network, all child nodes submit local data to the coordinator node. The coordinator node performs all of the necessary computation and achieves the global solution or the *aggregate*. Child nodes do not engage in any computation.

In a *decentralized* network, the coordinator node breaks down the global problem into multiple local problems. Each child node, attempts to solve its own local problem and submit the solution to the coordinator node. The coordinator aggregates the local solutions and achieves the global solution.

Taking it a step further, in a *distributed* network, there is no separate coordinator node. Or to put it another way, each node plays the 'roles' of child and coordinator both. Each node is a *full* node capable of all the computation. Each node (as child), identifies its local problem from the global and attempts to find the local solution to that. Then (as coordinator) it acquires the local solutions of other nodes and finds the aggregate of all local solutions, the global solution.

There are many decentralized EM approaches for DSM under research and implementation today. One example of this is *demand response* [43] which refers to the capacity of end-consumers in the energy supply chain to change their daily consumption profile based on market signals. In contrast there are no distributed approaches to DSM yet in practice.

In this thesis, we investigate whether a decentralized DSM approach can be transformed into a distributed method by using blockchain. We start by explaining profile steering in this chapter. Profile Steering is a decentralized approach to DSM by Gerards et al. [1]. In Chapter 5 we present two methods to implement *distributed* profile steering using consensus protocols from blockchain.

3.2 Profile Steering

3.2.1 Network Structure

Figure 3.1 shows the network structure for PS. In PS, an energy network is visualized as a tree-network of child nodes and their respective coordinator nodes. A node at Level n is the coordinator node for all nodes connected to it at Level $n-1$ and a child node for the coordinator node at Level $n+1$.

Taking a Low Voltage (LV) grid as example, each device in a house is a PS node (Level of Control 1), each house in a neighbourhood is a PS node (Level

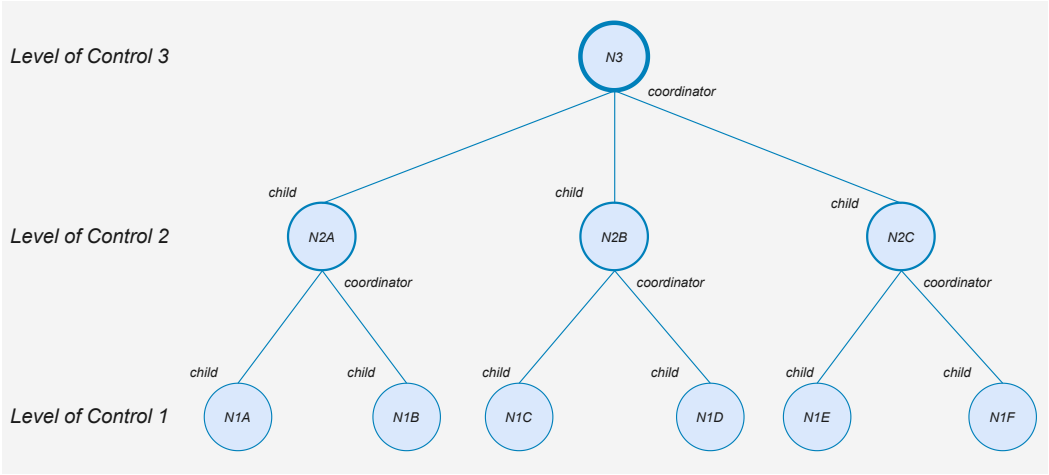


Fig. 3.1: Network structure for PS.

of Control 2) and the local transformer is a PS controller (Level of Control 3). Figure 3.2 visualises this.

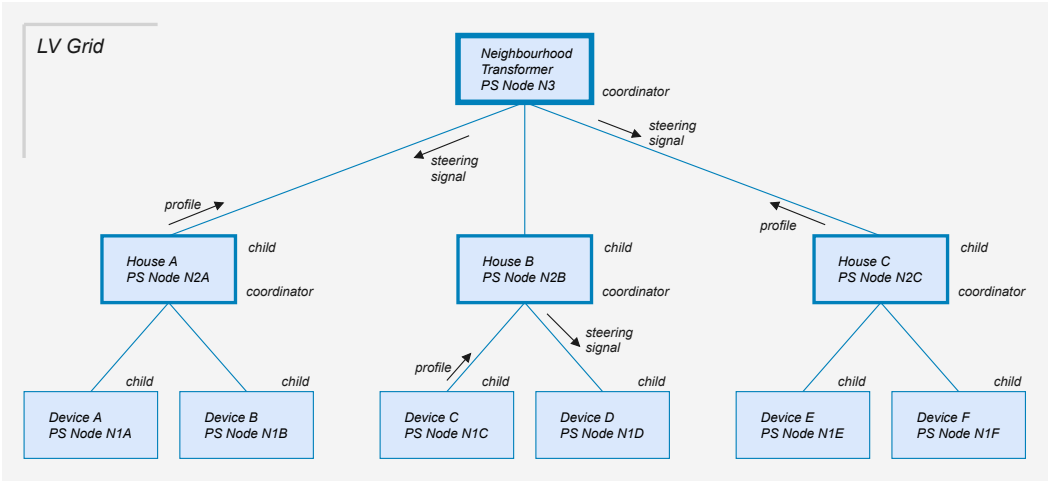


Fig. 3.2: Network structure for PS visualized for an LV grid as an example.

Take the case of Node N2A. Node N2A acquires a part of the Level 3 global problem that applies to it. We call this as the desired profile for Node N2A. Then Node N2A (coordinator), asks both its child nodes N1A and N1B to come up with their preferred profiles or schedule for the day. These are called candidate profiles. Coordinator node N2A acquires the candidates profiles and updates each in the N2A aggregate profile. Node N2A selects the candidate profile that brings the N2A aggregate profile closest to the N2A desired profile. The other candidate profile is discarded. This completes one PS iterations. Then the coordinator N2A can initiate more PS iterations and in each PS iteration, child nodes N1A and N1B will submit candidate profiles,

and N2A tests and selects the candidate profile that brings the N2A aggregate closer to the N2A desired.

> *In profile steering, the coordinator node N2A uses the N2A desired profile as a signal to appropriately select and set N1A and N1B child node profiles in a way that steers the actual N2A profile closer to the N2A desired profile.*

In each PS iteration the actual profile at a certain level gets closer to the desired profile at that level. A coordinator node engages in PS iterations until a required degree of closeness between the two profile or a minimum number of PS iterations is achieved. Similarly, coordinator node N3 engages in PS iterations with child nodes N2A, N2B and N2C to steer the aggregate N3 profile closer to the desired N3 profile. In the LV grid example, this is equivalent to a transformer (N3) selecting and discarding child house profiles (N2A, N2B, N2C) to steer the aggregate neighbourhood profile closer to the desired neighbourhood profile.

3.2.2 Steering Signals

In a decentralized steering approach, a global controller has the objective of optimizing the aggregate load profile depending on the steering signal. It is based on this steering signal, that the global controller will decide to choose or drop local candidate profiles. Therefore the steering signal chosen plays an important role in determining the aggregate profile. Below we briefly discuss steering based on price signals and power signals. We refer to [44] for an in-depth discussion of steering signals.

Price Steering

1. **Uniform Pricing**: In this scheme, the same price signal is sent to all households. Households respond to the price signal by shifting their loads to the time slot with lesser tariffs or cheaper Time-of-Use (ToU) prices. Since all households are incentivized to engage in this behaviour based on the exact same price signal, this only leads to the peaks being shifted in time but not cancelled.

2. **Dynamic Pricing**: One solution to the peak-shifting problem of uniform pricing is dynamic pricing. In dynamic pricing, different houses are presented with different price signals. However this incentivizes each house to shift their load to the time slot during which they receive the lowest ToU price. While this may cancel any peaks at the neighbourhood level, this behaviour leads to houses having local peaks at different times during the day. This may lead to a power imbalance at the house level resulting in possible voltage problems and overloading of cables. Thus dynamic pricing only partially alleviates the problem.

Power Steering

It is preferable to use a steering signal that indicates the desired objective clearly. If the objective is minimizing voltage peaks and thereby power outages, then it is better to use a flat power profile as the desired profile. Coordinators in a hierarchical network steer the profiles of their child nodes to achieve an aggregate profile as close as possible to this desired flat power profile. Coordinators thus aim to minimize the distance between the aggregate profile and the desired flat power profile.

3.2.3 Algorithm

Table 3.1 presents the required notations and their meaning. We use these notations to present the PS algorithm in Algorithm 1.

Tab. 3.1: Notations used in PS Algorithm 1

Notation	Meaning
$\vec{x}$	Aggregate profile at Level of Control N
$m \in \{1, ...M\}$	Child nodes at Level of Control $N-1$
$\vec{p}$	Desired profile at Level of Control N
$\vec{x_m}$	Current profile of child node m
$\vec{d}$	Difference vector
$\vec{p_m}$	Desired vector for child node m
$\vec{\hat{x}_m}$	Candidate profile of child node m
$\|\|\vec{\hat{x}_m} - \vec{p_m}\|\|_2$	2-norm Euclidean distance between candidate profile and desired child node profile
e_m	Improvement in child node's profile
ϵ	Constant for error margin (to limit PS iterations)

Algorithm 1: Hierarchical PS Algorithm (Coordinator at Level N) [1]

Request each child node $m \in \{1, ..., M\}$ at level $N-1$ to minimize $||\vec{x_m}||_2$
$\vec{x} := \sum_{m=1}^{M} \vec{x_m}$ {Aggregate level N profile }
repeat
  $\vec{d} := \vec{x} - \vec{p}$ {Difference Vector}
  for $m \in \{1, ..., M\}$ **do**
    $\vec{p_m} = \vec{x_m} - \vec{d}$
    For child node m, find a planning that minimizes $||\vec{\hat{x_m}} - \vec{p_m}||_2$
    $e_m = ||\vec{x_m} - \vec{p_m}||_2 - ||\vec{\hat{x_m}} - \vec{p_m}||_2$ {Improvement in profile of child node m}
  end
  Find the child node m with the highest contribution e_m
  $\vec{x} := \vec{x} - \vec{x_m} + \vec{\hat{x_m}}$ {Update the aggregate level N profile}
  $\vec{x_m} = \vec{\hat{x_m}}$ {Update the profile of the child node m}
until $e_m < \epsilon$ {Repeat as long as there is sufficient progress}

The local *objective* is to minimize the Euclidean distance between $\vec{x_m}$ and $\vec{p_m}$. We use the Euclidean 2-norm for the objective since it has much better performance in terms of power quality and losses, because losses are proportional to the squared power. We refer to [1] for a detailed discussion about the same

3.2.4 PS Flowchart with LV grid example

Figure 3.3 gives a visual idea of PS Iterations executed by the neighbourhood coordinator node on child house nodes to arrive at the final neighbourhood profile. We take the example of a neighbourhood consisting of 25 houses. These 25 houses are child nodes to the neighbourhood transformer. The neighbourhood transformer performs PS iterations with the house nodes and attempts to steer the neighbourhood profile towards the desired profile.

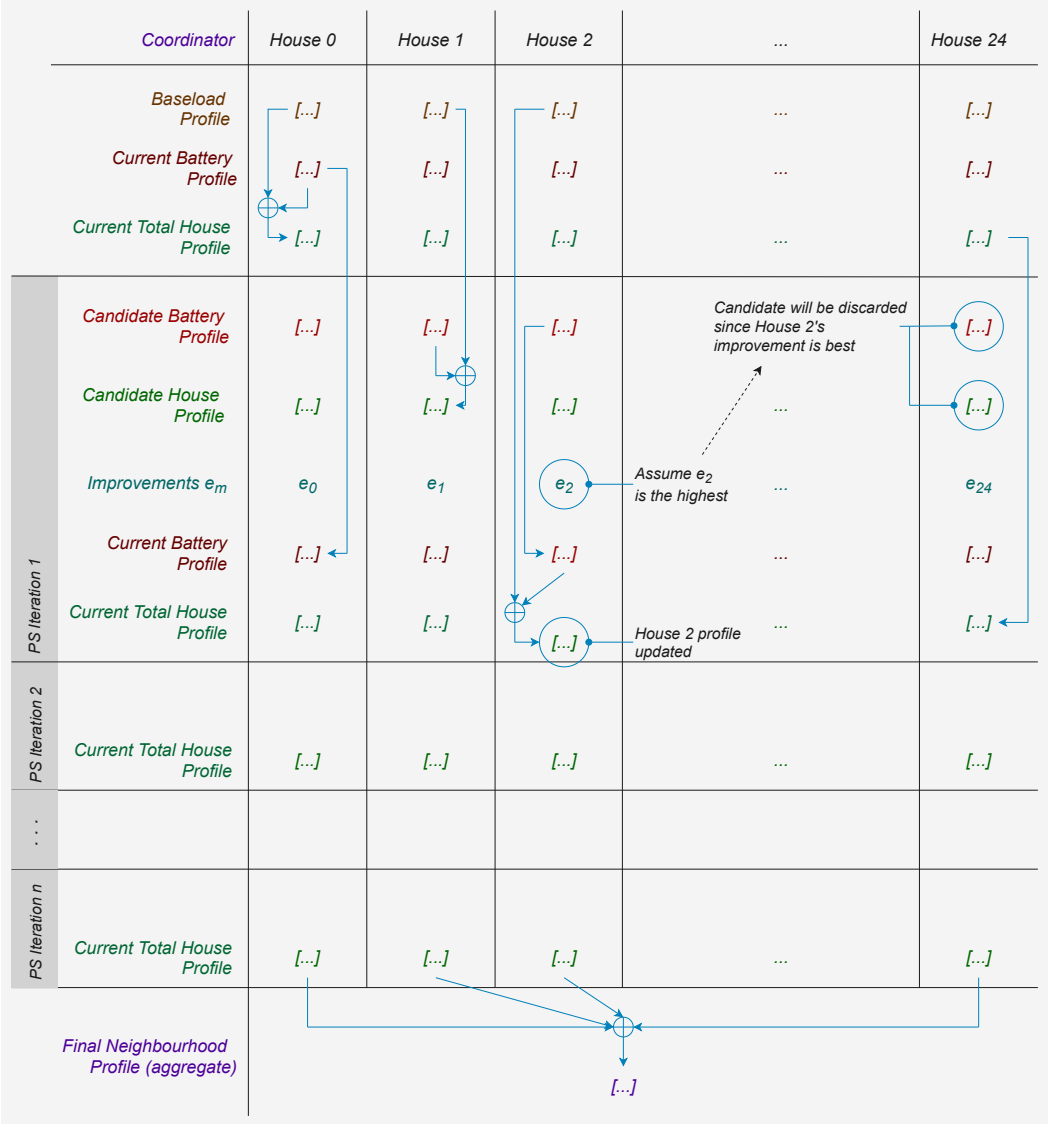


Fig. 3.3: PS Iterations performed by coordinator on house nodes to optimize neighbourhood profile.

3.3 Simulations

3.3.1 Dataset

We used residential data from Pecan Street [45]. The data set consists of the load profiles of multiple appliances from 25 houses in New York over 184 days (6 months from May 1st to October 31st). Samples are recorded once every 15 minutes per day (so 96 samples per day). The dataset also contains the power generation from PV panels for houses that do have a PV system

installed. We aggregate the load profiles of all the appliances and the PV generation into one baseload profile for each house per day. So each of the 25 houses starts out with a baseload profile containing 96 samples for each day. In one case where a data point was missing in the initial dataset (there were 95 total samples for that house on that day), an extra data point was appended to that set to give that house 96 samples for that day. This extra data point was a copy of the preceding data point. As per the dataset, none of the houses have any storage devices.

3.3.2 Simulations

Load duration curves for three cases are simulated (Table 3.2).

Tab. 3.2: Description of the three simulation cases.

Case	Description	Battery Capacity per house (Wh)
Case 1	No DSM appraoch used. Just the baseload profiles.	0
Case 2	Profile Steering used. Each house has a B-Box Pro 5.0 battery	5120
Case 3	Profile Steering used. Each house has a Tesla Powerwall 2 battery	13500

3.3.3 Load Duration Curves

Figure 3.4 shows the load duration curves achieved in all three scenarios.

3.3.4 Discussion

Figure 3.4 shows that the load curve is closest to the desired profile in Case 3, then in Case 2 and then in Case 1. The graph shows how the PS algorithm uses the flexibility offered by the house battery to set house profiles in a way that the aggregate neighbourhood profile can be steered closer to the desired profile. The flexibility offered by the house depends on the battery capacity of the house. Case 3, wherein houses have a higher battery capacity than in Case 2, performs better than Case 2. These simulation results are used

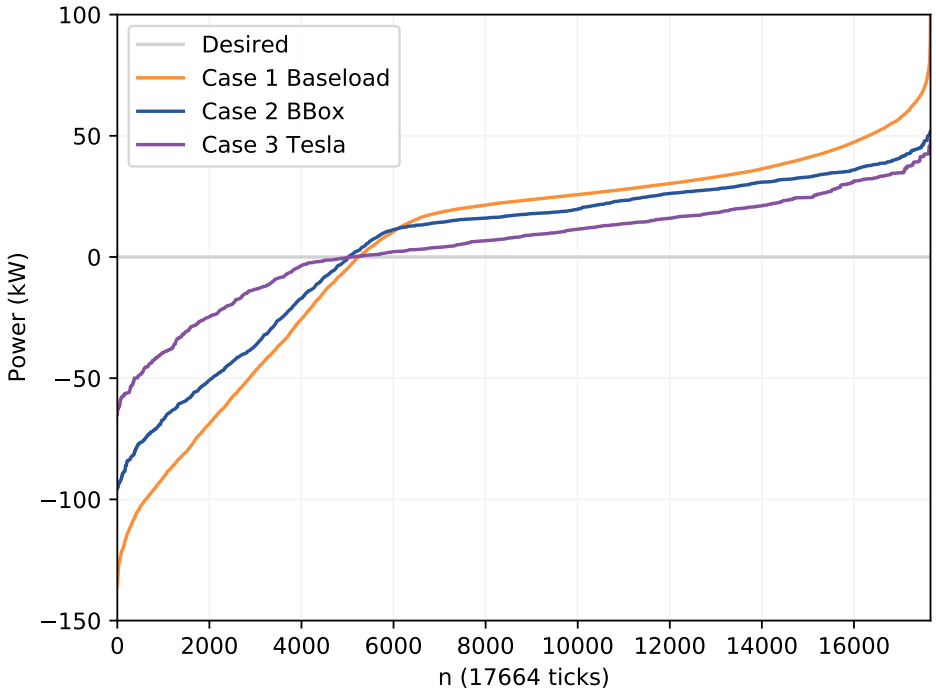


Fig. 3.4: Load Duration Curves for all three cases.

as a baseline to evaluate the distributed implementations of profile steering proposed in Chapter 5.

Chapter Summary

1. Profile Steering is a decentralized approach to DSM.
2. Child nodes submit candidate profiles to coordinator node. Coordinator node selects and sets child node profiles from candidates in a way that steers the aggregate profile closer to the desired profile.
3. Quality of output of PS depends on battery capacity. More is the battery capacity of a child node, more is the total flexibility of the network.

Understanding Blockchain

4

Chapter Objective: The concept of distributed ledger systems, and its most famous implementation, Blockchain, is explained in this chapter. Furthermore, the working of a blockchain network from node roles to consensus protocols is presented. Concepts given in this chapter form the basis of the distributed profile steering models presented in the next chapter.

Chapter Contents

4.1 Introduction

A blockchain is distributed network consisting of multiple nodes wherein each node holds with itself a copy of the so-called ledger (distributed ledger). This ledger is a database containing all the transactions that have happened in that network. The records in the ledger are called blocks. Each new block is linked to the previous one cryptographically. Thus the ledger is essentially a growing 'chain of blocks', hence *blockchain*.

Nodes in a blockchain network engage in transactions with each other. Whenever a transaction is done it is broadcasted to all nodes in the network. Nodes listen for transactions and keep adding them to draft block. Once a draft block has enough transactions, that block must be formally added to the blockchain. A transaction is confirmed only when the block it has

been recorded into is been added to the blockchain. All nodes have the transactions that must be recorded in the new block (all nodes have the same draft block), but only one node can get the right to add the block to the blockchain. That is, the network must arrive at a consensus as to which node gets to add the block to the blockchain. This process of creating and adding a block to the blockchain is called mining and blockchain networks use different consensus protocols to decide the winning or mining node.

A blockchain, due to the connected nature of the blocks, is highly resistant to alteration. To alter a block, all the blocks from the current block back to the block that needs to be altered would have to be deleted. This automatically reduces the incentive for a malicious node to change or delete a particular block. Thus, all blockchain networks can be seen as distributed P2P networks consisting of mutually suspicious nodes with individual self-interests, that autonomously manage a distributed ledger and update the ledger with transactions validated by mass collaboration. Due to their distributed and transparent nature, blockchain networks boast of a very high degree of security from malicious attacks on the blockchain and its contents from network nodes.

The idea of blockchain was first conceptualized by a person (or group of persons) by the name Satoshi Nakamato in 2008 [46]. Satoshi's identity remains unknown till this date. While some previous theoretical work on the topic of distributed ledger systems had been done prior to Satoshi's paper, his paper gave a practical implementation of the distributed ledger systems. Satoshi used a Hashcash-like method to make it possible to digitally sign the blocks without requiring approval from any central or dedicated signatory *(detailed explanation in Subsection 4.2.5)*. Satoshi also introduced a difficulty parameter into the blockchain to stabilize the rate at which new blocks were being generated *(detailed explanation in Chapter 6, Subsection 6.2.2)*.

Bitcoin [47], the world's first cryptocurrency network was started by Satoshi based on the model outlined in his paper. While Distributed Ledger Systems is one concept and blockchain makes use of this concept, there are many cryptographic concepts that blockchain relies on. To put it another way, blockchain is the assimilation and collective working of all these cryptographic concepts developed independently over the past many decades of research. In the next section, some of these concepts are explained. Emphasis is placed on

explaining concepts used in the distributed DSM models proposed in this thesis.

4.2 Basic Blockchain concepts

4.2.1 Network Structure and Node Roles

A blockchain network is a distributed network. There is no central or coordinator node. There are three types of nodes in a blockchain network [48]:

1. **Full Nodes**: Full nodes store a copy of the entire blockchain within themselves. Their primary responsibility is to validate each block and its transactions. They can reject a block if its component transactions are invalid (fraudulent transaction or financially impossible transactions such as insufficient balance or double spending). Validation is not consensus. Validation happens before consensus. By storing copies of the entire blockchain, full nodes serve to distribute the entire network further making the blockchain more resilient. Full nodes store a copy of the entire blockchain and are data heavy nodes. For example, there are 11237 full nodes in the Bitcoin network [49] and each node stores the entire Bitcoin ledger which has grown to a size of 300 GB [50].

2. **Light Nodes**: Also known as thin nodes, a light node verifies the integrity of the blockchain copy stored inside a full node. Light nodes are thus paired with a full node. A light node validates the blockchain copy stored within a full node and informs its respective full node of any issues in the blockchain copy such as an incorrect hash. Light nodes do not store the entire blockchain copy. They aren't data heavy. They only store the block header and use Merkle Trees to validate the blockchain.

   Subsection 4.2.5 and Subsection 4.2.2 explain Merkle Trees and Hash Values in blockchain respectively, in detail.

3. **Miner Nodes**: Miner nodes participate in the consensus process of the blockchain. Once a block is validated by a full node, miner nodes compete on the basis of a consensus protocol (Subsection 4.2.7) to win the right to add the block to the blockchain. Giving a block a hash value

and adding it to the blockchain is called mining. Miner nodes are also paired with a full node.

The concepts of mining a block and consensus protocols are explained in detail in Subsection 4.2.3 and Subsection 4.2.7 respectively.

Figure 4.1 shows the network structure of a blockchain network.

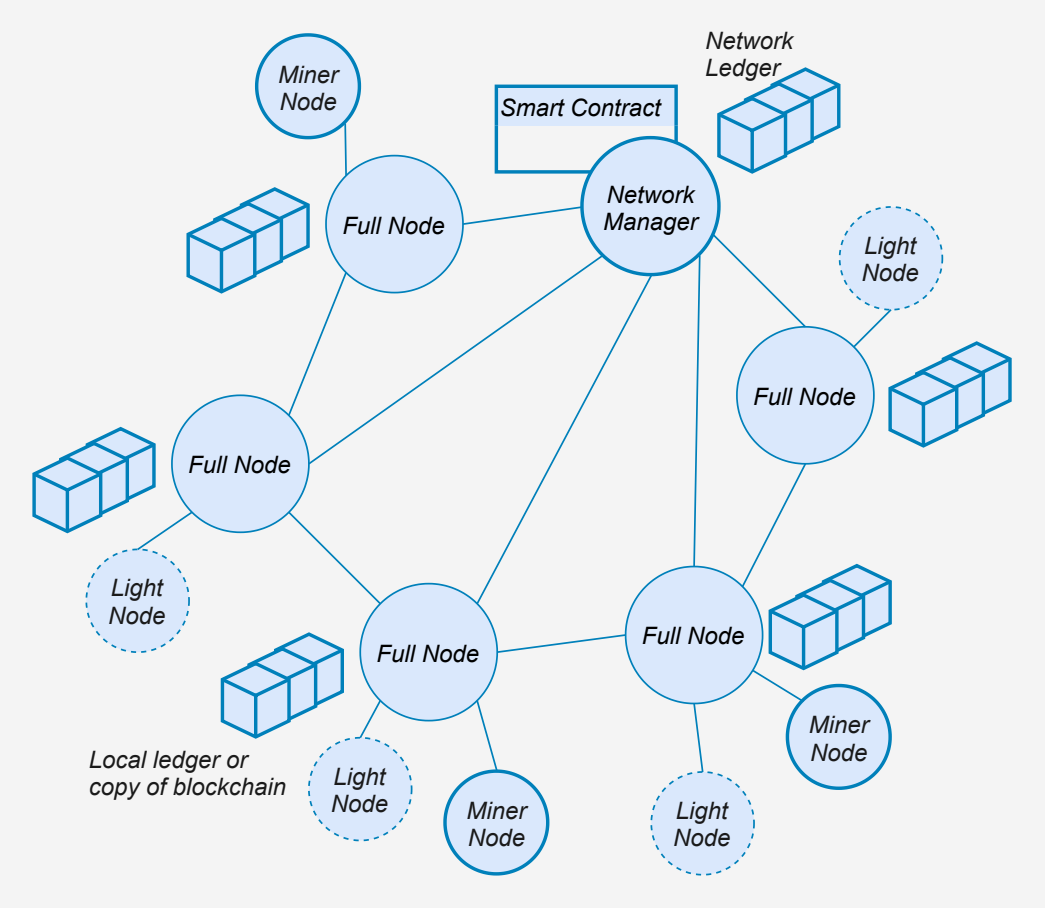


Fig. 4.1: Network structure of a blockchain network.

Since a copy of the ledger or blockchain is stored within each full node, blockchain is an example of a distributed ledger system. Unlike centralized or decentralized networks, access to the network's transaction history is not limited to one or more central nodes.

4.2.2 Blocks and Hashes

Each block holds a record of a certain number of transactions that occurred in the blockchain network. These transactions make up the block content section of the block. Apart from the block content, a block contains the following elements:

- **Block Index Number**: Each block in the blockchain has a sequentially incrementing number for easy identification. This is called the block index number. This number has no cryptographic importance and is only there to allow easy reference to a block in the blockchain.
- **Previous Block Hash**: A hash is a 256 bit value. Each new block records the hash of the previous block in its block header.
- **Current Block Hash**: Apart from the previous block hash, each new block has its own hash. This hash is a 256 bit value created by hashing the current block content and the previous block hash together. Thus the current block hash is unique throughout the blockchain and is derived from the previous block's hash value. Bitcoin uses the SHA256 hashing algorithm.

In networks that use Proof Of Work as their consensus protocol, each block also has the THV in the block header. This is explained in Section 4.3. The block content is the set of transactions added to the block. In Bitcoin this would be a set of latest cryptocurrency transactions that transpired in the network.

Figure 4.2 visualizes blocks, hashes and the resultant blockchain. A block is only formally added to the blockchain once it has its own hash value. Since this hash value depends on the previous hash value, this establish the linear chain-like connected nature of the blocks, hence blockchain.

Genesis Block

The Genesis block is the very first block in the blockchain. Miner nodes require each block to record the previous block hash value to be able to create the current block hash value. The first block in a blockchain, for lack of a previous block, cannot be mined. Therefore, the Network Manager (Subsection 4.2.4) creates the first block in a blockchain. This is called the

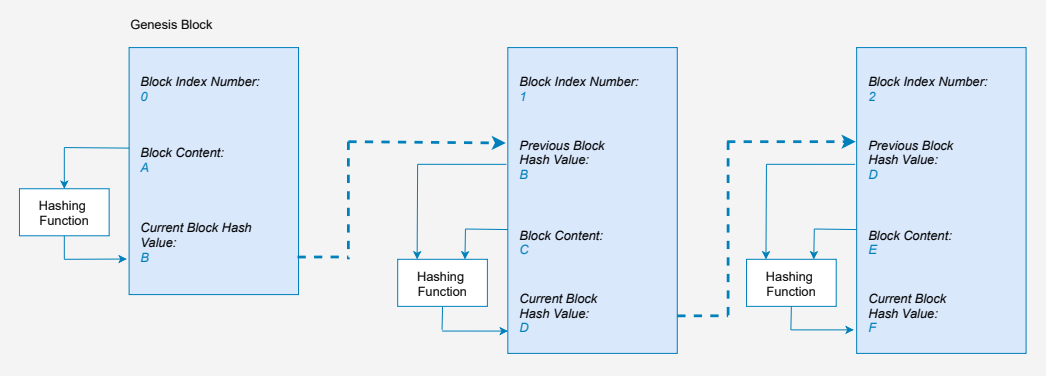


Fig. 4.2: Structure of a blockchain.

Genesis Block. The block may not have any content or may have some descriptive information about the network such as the starting timestamp of the network. The Genesis Block has no transactions and is a placeholder block created to start the chain. All successive blocks are mined or forged by the network's nodes.

4.2.3 Mining and Forging

A draft block consists of certain transactions that have occurred between nodes in a blockchain network. These transactions are first validated by full nodes. After validation the miner nodes step in to *mine* the block. A block is only considered as part of the blockchain once it has its own hash value. Miner nodes take the draft block and compete for the right to mine the block. The nature of this competition depends on the consensus protocol (Subsection 4.2.7) followed in that blockchain network. The miner node that wins the consensus round get the right to calculate and add a hash value to the current block. Once a block has its own hash value, it is formally part of the blockchain. This is known as *mining a block*. In some consensus protocols, this is also known as *forging a block*. Depending on the protocol, miner nodes may receive transaction fees or other financial awards as payback.

4.2.4 Network Manager and Smart Contracts (DApps)

Prior to entering any blockchain network, a node must sign what is known as a Smart Contract. A smart contract is a codified version of the rules that a

node is required to follow while it is a part of the network. It can be thought of as the terms and conditions a node must agree to before becoming part of the blockchain network. Apart from general rules relating to malicious or fraudulent behaviour, the smart contract contains the rules of the consensus round. By signing the smart contract, miner nodes agree to compete in the consensus round with respect to these rules. The smart contract also contains details about the financial payback for winning miner nodes.

The Network Manager holds the smart contract and is tasked with *enforcing* it. The network manager cannot alter the contract. The contents of the contract are transparent to all nodes at all times. Enforcing a contract means running the smart contract program when required, for example, when finding a winning miner node in a consensus round.

> *Since a single network manager runs the smart contract and miner nodes depend on this smart contract, smart contracts are also known as Decentralized Applications or DApps implemented over a distributed network.*

4.2.5 Security Concepts

A distributed network by its very nature gives rise to many doubts regarding the security of the network and the integrity of the blockchain itself. Explained below are some concepts used to make blockchain secure. Security in blockchain is an extremely vast and complex field and the below explanations only serve the purpose of an introduction to some of these concepts.

Digital Signatures for Authentication

Authentication is an important requirement in a distributed P2P network. When a miner node mines a block and broadcasts the new block to the other nodes, the other nodes on receiving the block must be able to confirm its source so as to not fall prey to fraudulent blocks from malicious nodes. Authentication in blockchain is achieved by using Public Key Cryptography (PKC), also known as Asymmetric Key Cryptography.

Figure 4.3 shows how nodes employ PKC to digitally sign a block before sending it and how receiver nodes use this for authentication. Each blockchain

node is given a Private Key and a Public Key. A Private Key and a Public Key form a key pair. Public Keys of all nodes are accessible throughout the network. However each node must keep its Private Key confidential.

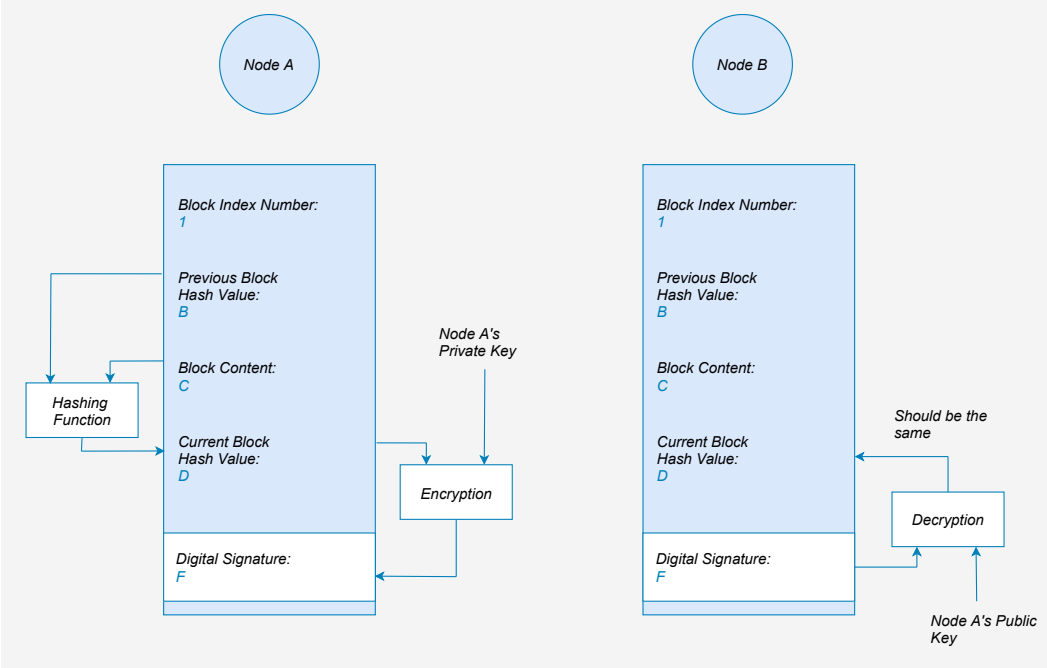


Fig. 4.3: Public Key Cryptography for authentication in blockchain.

To digitally sign a block, the sender node creates a hash value for the current block by sending the block content through a hash function. Then it encrypts this hash value with its Private Key. The result of this encryption is the digital signature added to the block. When a received node gets the block, it decrypts the digital signature with the apparent sender node's Public Key. The output would be a hash value. The receiver node then sends the block content through the same hash function and checks if the resultant hash is the same as the hash value derived after decryption. A match implies successful authentication.

Merkle Trees for Blockchain Integrity

Light Nodes attached to Full Nodes use Merkle Trees to verify individual blocks or parts of the blockchain. Figure 4.4 shows how a Merkle Root and the resultant Merkle Tree is created [51].

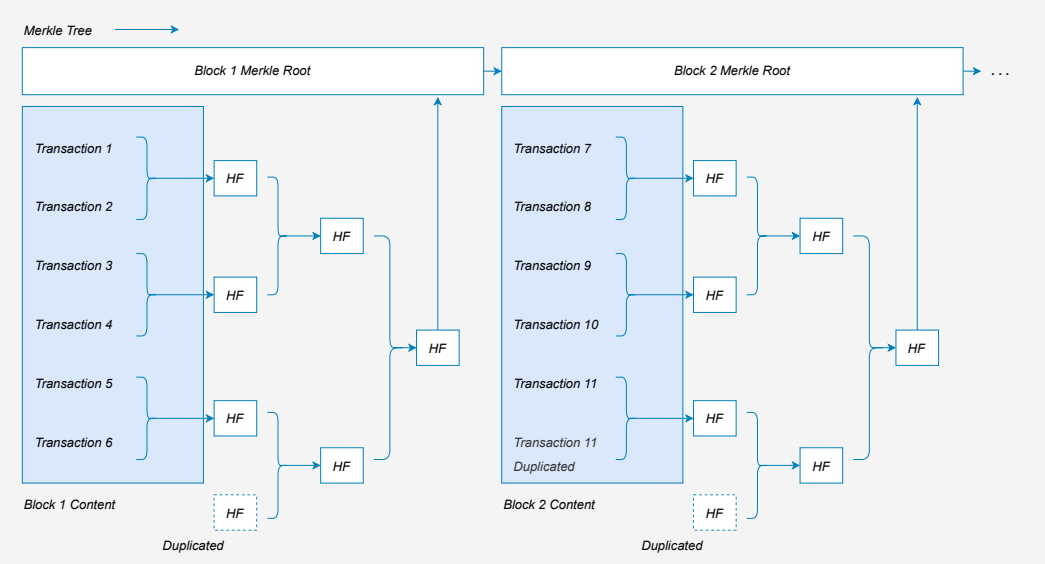


Fig. 4.4: Merkle Roots and Merkle Trees in Blockchain. HF = Hashing Function.

Each block has a unique Merkle Root. To create a Merkle Root, transactions in a block are grouped in pairs. If there is an odd number of transactions, then the last transaction is duplicated. Each pair of transactions are passed through a hashing function. The resultant hash values are again grouped in pairs and pass through a hashing function. This process is repeated until only one hash value remains, which is the Merkle Root of the block.

Each block in a blockchain has a Merkle Root and this sequence of Merkle Roots is known as a Merkle Tree. When a Light Node wants to verify its local blockchain copy, it downloads the current Merkle Tree from the Network Manager's ledger and compares it with the Merkle Tree of its own local ledger. If each Merkle Root in the local ledger matches with its corresponding Merkle Root from the network ledger, then this verifies the contents of the local ledger. Merkle Trees give Light Nodes a quicker way to verify the local blockchain copy without having to download the entire network blockchain's contents and actually having to compare the individual blocks line by line. Since two different pieces of data cannot give the same hash value, equality of each Merkle Root and the Merkle Tree implies successful validation of the local blockchain.

4.2.6 Types of Blockchain Networks

There are four types of blockchain networks.

Public

Features: They are non-restrictive and permission-less. Anyone with internet access can join this network. All nodes have the right to access the ledger, verify transactions and mine blocks. All ledger data is transparent and visible to every node.

Examples: Bitcoin, Ethereum and Litecoin.

Private

Features: They are restrictive in terms of access. Permission from a central authority is required to join the network. The central authority decides the levels of permission and access each node has. They are usually deployed in more private or controlled settings wherein a clear cut idea of each node's identity is mandatory. They offer maximum protection against malicious nodes entering the network, with the downside that they are much less decentralized in terms of governance than public networks.

Examples: Hyperledger Sawtooth, Corda and Multichain.

Consortium (Federated)

Features: They are part public and part private hence semi-decentralized in terms of authority. The division of roles, rights and powers is done via consensus. Such blockchains are governed by a group and not a single entity. Such networks are used when a balance between public and private blockchain features is required.

Examples: Quorum, Corda and Hyperledger.

Hybrid

Features: They are a combination of private and public blockchains. They consist of both public permissionless systems and private permissioned ones. Nodes can decide which part of the ledger data they would like to keep private and which part should be public. Nodes can be part of multiple public and private blockchains within the parent hybrid network.

Examples: Dragonchain

4.2.7 Consensus Protocols

Once a certain number of transactions are completed, they must be added to a block and the block must be made part of the blockchain. As explained before, the task of taking some block content, adding a hash value to that block and making it part of the blockchain is called mining or forging. Since only one miner node can gain the right to mine a block, networks must arrive at a consensus as to which miner node gains mining rights. Depending on the blockchain network, different consensus protocols are used for the consensus stage. Consensus protocols are one of the most important aspects of blockchain since they are directly responsible for its distributed nature.

The two most important consensus protocols used in blockchain are Proof of Work (PoW) and Proof of Stake (PoS) explained in Section 4.3 and Section 4.4 respectively. They were the first two consensus protocols to be implemented in blockchain, with many commercial blockchain applications initially adopting PoW. Other consensus protocols used or being tested in commercial blockchain networks are described in Section 4.5.

4.3 Proof of Work (PoW)

PoW was the first consensus protocol applied in Bitcoin crypto-currencies networks based on Satoshi's proposed model. All miner nodes are given a mathematical problem to solve. Solving this mathematical problem is a computationally intensive operation. On the contrary, verifying the solution of the mathematical problem is a computationally easy operation. Miner

nodes take the block content and the previous block hash value and attempt to solve the mathematical problem in the shortest amount of time. If successful in finding one valid solution, they submit their solution to the Network Manager for verification. The miner node that submits a valid solution in the shortest amount of time is declared the winner and gains the right to calculate the current block hash value and add it to that block. PoW is therefore a time-critical competition between miner nodes to find a valid solution to a pre-determined mathematical problem. In Bitcoin, winning nodes are rewarded transaction fees for every block they mine.

4.3.1 Algorithm

Algorithm 2 shows the steps in the PoW consensus protocol.

Algorithm 2: Proof of Work Consensus

```
1  Get block content and previous block hash value
   /* For consensus                                          */
2  repeat
3  |  Attempt to find a valid solution to pre-determined mathematical problem
4  |  if Solution is valid then
5  |  |  Send solution to Network Manager for verification
6  |  end
7  until Winner node is declared by Network Manager
8  if This node is Winner then
9  |  Calculate current block hash value
10 |  Add current block hash value to block
11 else
12 |  Receive new block from other winning node
13 end
```

4.3.2 PoW Mathematical Problem

The PoW mathematical problem is a problem whose solution is difficult to calculate but easy to verify. One common example of the PoW problem is finding a valid hash value. In this problem, all miner nodes are given a Target Hash Value (THV). Miner nodes take a random variable factor called a *nonce* and append it to the block content and the previous block hash value. Then

they find its combined hash value. If this hash value is less than or equal to the THV then this hash is considered a valid hash and the corresponding nonce factor, a valid nonce. If not, then it's an invalid solution and the miner node finds another random nonce factor and tries again [52]. The finding valid hash value mathematical problem is used in the PoW models presented in this thesis. The practical implementation of PoW and the hash value problem are further explained in detail in Chapter 5. Another example of a PoW mathematical problem, is the Integer Factorization problem. Miner nodes must take a pre-specified number and present it as a multiple of two other numbers.

4.3.3 Merits and Demerits

PoW, by its computationally heavy nature, imposes certain automatic restrictions on actions performed in and against the network. For a malicious node to attack and take over the network's consensus algorithm and generate fraudulent blocks, it would have to own atleast 51% of the network's computational capacity. The high cost involved in this ownership disincentivizes attackers. This form of attack is called a *51% attack* or *majority attack* and the application of PoW makes it immensely costly for any node to attempt this. However it must be noted that while such attacks are difficult, they aren't impossible and nodes can form coalitions amongst themselves to assemble more computational power together. Such (super-)nodes then stand a higher chance of winning mining rights in every iteration and share the rewards amongst the members of the coalition. This is detrimental to the distributed nature of authority that blockchain networks aspire for as their core operating principle.

On the other hand, a primary demerit of the PoW algorithm is its huge expenditure and the useless nature of its computations. Mining blocks requires specialized computer hardware known as ASICs (Application Specific Integrated Circuits). This leads to the creation of special mining pools that consists of multiple rows of ASICs all running to solve the PoW mathematical problem. The large financial investment required to assemble, run and maintain such mining pools becomes a prohibitive factor for new miners to join the blockchain network. Again this risks centralizing the network in the hands of rich miners.

In addition, each mining operation requires the nodes to solve the mathematical problem by competing to find a valid hash function. Since there can be only one winning node, this leads to many useless energy-intensive computations by the miner nodes. PoW is notorious for its immensely high energy consumption and its unsustainable nature. Taking the example of the Bitcoin network, the current electricity consumption of the Bitcoin network is 77.78 TWh which is equivalent to the amount of electricity required to power a country like Chile. Bitcoin has a carbon footprint of 34.95 Mt CO_2 comparable to that of New Zealand [53]. Figure 4.5 shows bitcoin electricity consumption relative to that of several countries and Figure 4.6 shows an energy comparison between Bitcoin and VISA transactions.

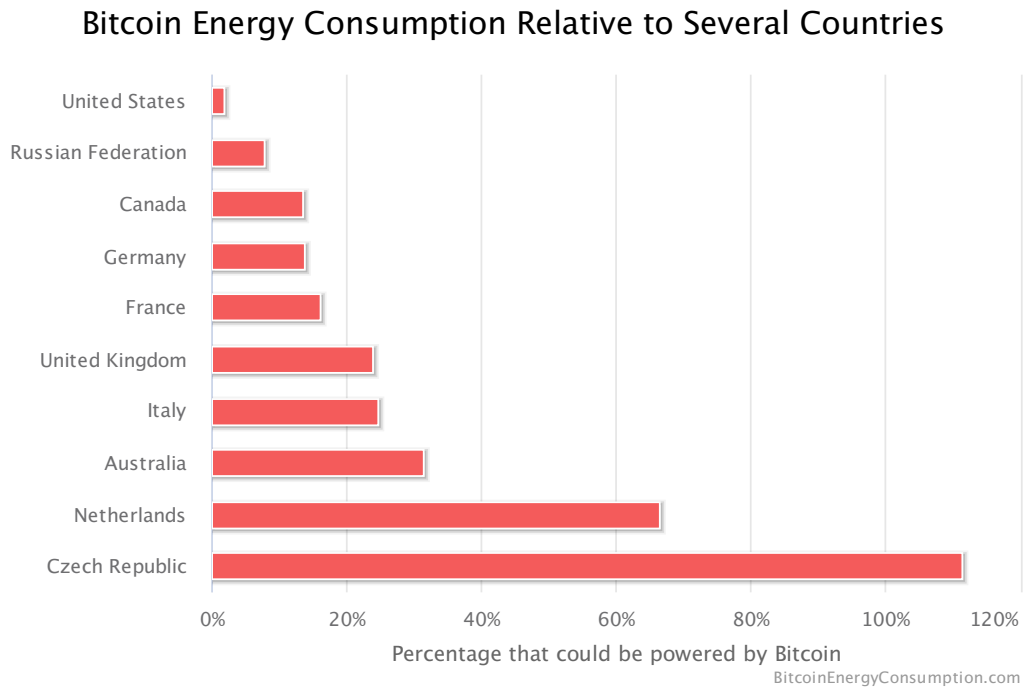


Fig. 4.5: Bitcoin energy consumption relative to that of several countries [53].

Some blockchain platforms that use PoW for consensus are Bitcoin, Litecoin and Ethereum.

4.4 Proof of Stake (PoS)

In a blockchain network using PoS as its consensus protocol, nodes enter the network by pledging some initial stake in the network. In Bitcoin networks,

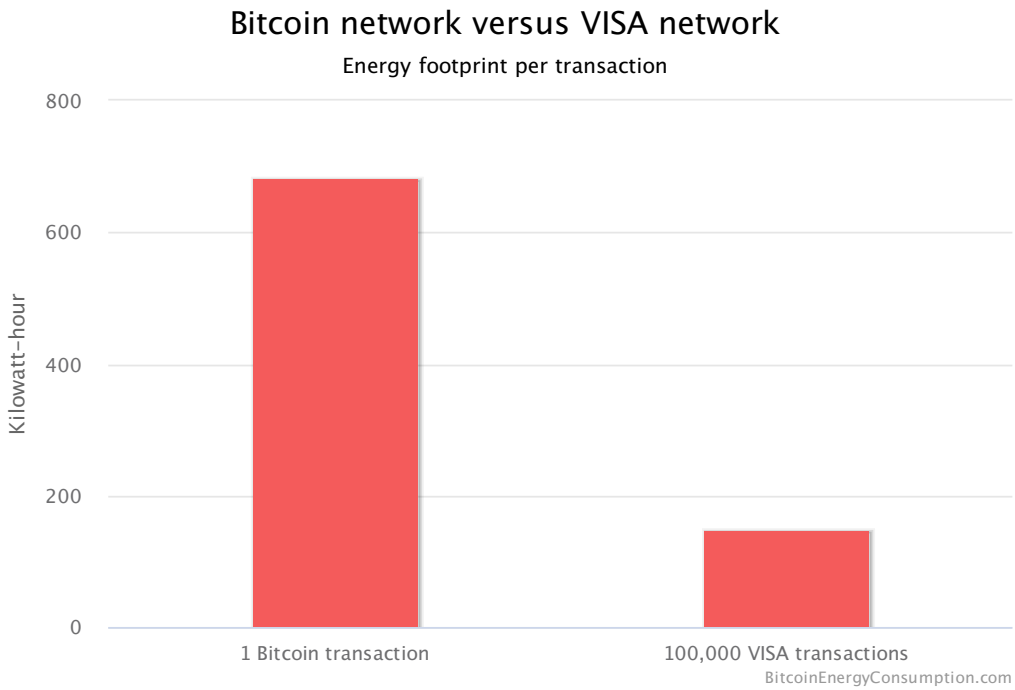


Fig. 4.6: Bitcoin vs VISA transactions in terms of energy footprint per transaction [53].

this would be a financial amount that new nodes deposit into the network. During the consensus round, the chances of a node winning block forging rights is proportional to its initial stake deposited in the network [54]. Unlike in PoW, in PoS nodes do not engage in a computationally intensive consensus competition. In PoS, miners are limited to mining a percentage of blocks reflective of their initial stake or ownership stake in the network. The winner in PoS is selected by a pseudo-random process based on a combination of factors such as the value of the initial stake and (in some networks) the staking age. The staking age is the amount of time for which a node has deposited stake in the network. Nodes receive financial rewards in return for forging blocks.

4.4.1 Algorithm

Algorithm 3 shows the steps in the PoS consensus protocol.

Algorithm 3: Proof of Stake Consensus

```
   /* At the start                                                    */
 1 Deposit initial stake while entering network
   /* For consensus                                                   */
 2 Get block content and previous block hash value
 3 (Network Manager calculates winning node by using initial stake values of
     nodes)
 4 (Network Manager declares winner)
 5 if This node is Winner then
 6 |   Calculate current block hash value
 7 |   Add current block hash value to block
 8 else
 9 |   Receive new block from other winning node
10 end
```

4.4.2 Merits and demerits

The primary merit of PoS over PoW is that the selection of the winning node depends on the amount of initial stake deposited in the network and this selection is easy in computational terms, unlike the energy-intensive process in PoW. PoS is much more sustainable from an energy consumption point-of-view than PoW. Indeed this is the primary reason why PoS was developed as a solution to PoW's energy problem. From a technical perspective at least, joining a PoS based blockchain network is much easier since no knowledge or investment in specialized ASIC mining pools is required.

Another important merit of PoS is its built-in security feature. In order to effectively control the network and forge fraudulent blocks a node would have to own majority financial stake in that network. Firstly, depending on the value of cryptocurrency at the time, achieving this majority ownership would be difficult and expensive. Secondly, if a node does achieve majority stake ownership in a network, then attacking the same network would also devalue its own stake in the same network and be financially detrimental for the malicious node itself.

However one demerit of a PoS network is *stake runoff* or the 'the rich get richer' problem [55]. When a node wins the right to forge a block it receives a financial reward in return. This node may then choose to reinvest that reward back into the same network thus increasing its ownership stake in the network. It's also possible that the same node already invested a huge amount

of initial stake in the PoS network. Thus this node would end up increasing its chances of winning block forging rights further. This form or re-investment of the transaction reward back into the network is likely to lead to a high chance of the same node winning on consecutive occasions. This problem is called stake runoff. It leads to a similar consequence of centralization of mining power as does rich mining pools in PoW consensus.

Some blockchain platforms that use PoS are Nxt, Dash and Tezos. Peercoin uses a mixed system with both consensus protocols. Currently Ethereum is in the process of switching from PoW to PoS consensus.

4.5 Other consensus protocols

While PoW and PoS are the most widely used distributed consensus protocols in blockchain networks, we briefly touch upon some other blockchain protocols in to complete our overview on the topic.

Delegated Proof of Stake (DPoS) [56]:
It's similar to PoS but all nodes aren't allowed to forge blocks. Nodes vote for *delegate* nodes who in turn forge blocks. Each node's voting power is proportional to the stake deposited in the network. Delegates that fail to contribute to consensus lose some part of their *reputation*. Reputation decides possibility of re-election. It is possible for nodes to join the network in its early stage, and form voting coalitions by voting for each other, thus centralizing the network (demerit).

Proof of Weight [57]:
A protocol used in blockchain networks meant for more literal applications than just crypto-currencies. For example, a file storage project. Chances of winning consensus round is proportional to the *weight* of each node. Weight of a node is dependent on both number of coins staked in the network and some other physical parameter (such as number of files stored for the network in a file storage project). There is an incentive to both, hold coins for the network and meaningfully contribute to the network as well (merit).

Proof of Burn (PoB) [58]:
PoB is called a PoW system without the energy waste. Miners 'burn' coins to buy virtual mining power in the network. Chances of mining blocks is

proportional to amount of virtual mining power. To burn coins miners send the coins to an un-spendable address in a process that doesn't consume many resources. This ensures that networks remains agile and active. Miners receive financial rewards for mining blocks. It is expected that overtime these financial rewards compensate for money spent in buying virtual mining power. However, PoB still needs to be tested on a large scale.

Proof of Participation (PoP) [59]:
First introduced by Blockchain Zoo in 2019, this protocol was intended to remove the link between decision power and resources to better enable decentralization. Anyone can apply to be a node. All nodes have an equal chance of being randomly chosen to mine the next block. Once mined, the block itself randomly chooses nodes to distribute the mining reward to. Chances of any node to be chosen depends on its level of useful participation in the network. Nodes that do not create high quality blocks or do not participate in the consensus protocol are removed from the network.

Proof of Elapsed Time (PoET) [60]:
Was developed by Intel Corporation for permissioned blockchain networks. A random leader election code is run by each node that generates a random sleep time. All nodes sleep for that random amount of time. The node with the shortest sleep time wakes up first and gets to mine the new block. The primary challenge in this algorithm, is to make sure that nodes do indeed select a purely random sleep time. The second challenge is to validate whether the winner has indeed completed the allotted sleep time.

Proof of Assignment (PoA) [61]:
Was designed for Internet of Things (IoT) devices to enable micro-mining in them. The basic difference from PoW is that the ledger storage is outsourced to other trusted nodes on the network. Miner nodes do not store the ledger. Miner nodes engage in a form of micro-mining to find the hash value. Like in PoW, the first to find the correct hash value wins, mines the block and sends it to the storage nodes for safekeeping.

Directed Acyclic Graphs (DAG) [62]:
DAGs are a data structure and different from the conventional blockchain data structure. Unlike in a blockchain wherein new blocks are connected to old blocks, in a DAG, new transactions are built on previous transactions. A transaction is validated only if it is built upon with another transaction. The use of DAGs in cryptocurrency to achieve consensus is a very new concept

and still needs to be tested in terms of capacity for decentralization and scalability.

4.6 Applications of Blockchain

Given below are some examples of applications of blockchain in different industries.

In Healthcare:
Factom [63], a Texas-based company has created a secure blockchain platform for hospitals and healthcare administrators to store their digital records. Even physical documents such as bills and prescriptions can be equipped with special Factom chips, giving them a digital identity and then be stored on the Factom blockchain platform.

In Gaming:
HashCash Consultants [64] uses blockchain in gaming for safe storage of scores and in-game transactions. The immutable nature of blockchain helps in curbing rampant fraud in the gaming industry. Additionally, gamers now have the option to execute transactions using cryptocurrencies.

In Voting:
Voting is a domain for which many proposals to use blockchain networks to prevent voter fraud have been suggested. Hjálmarsson et al. [65] propose a blockchain based e-voting system. They propose a permissioned network with fixed roles for each node and smart contracts to tally all votes in a certain location. The platform prevents double-voting(-spending) by assigning each voter a unique digital wallet prior to the election and a unique transaction ID after they cast their vote as proof.

Proof of Work and Proof of Stake are the two most commonly used consensus protocols in blockchain implementations. In the next chapter, the blockchain concepts explained in this chapter and the PoW and PoS consensus protocols are used to implement profile steering over a distributed energy network.

Chapter Summary

1. Blockchain networks are distributed networks. Each node has a copy of the blockchain ledger. Each block contains some transactions, the previous block hash value and the current block hash value.
2. The current block hash value is calculated as the hash of the block content and the previous block hash value. SHA256 is the most secure hashing algorithm in commercial use.
3. The Network Manager holds and runs the Smart Contract. The Smart Contract is the codified version of legal rules that all nodes agree to follow. The Smart Contract contains rules regarding consensus protocols, mining rewards and penalties for malicious behaviour, among other things.
4. In PoW, nodes compete to solve a mathematical problem. The first node to find a valid solution gets block mining rights. Primary demerit is high energy consumption.
5. In PoS, the chances of a node winning block forging rights is proportional to its initial stake in the network. Primary demerit is stake runoff.

5 Distributed DSM using PoW and PoS

Chapter Objective: In Chapter 3, a decentralized approach to DSM called Profile Steering (PS) was explained and in Chapter 4 foundational concepts regarding blockchain and consensus protocols were explained. We use the concepts put forward in these two chapters to propose two implementation methods for distributed PS each paired with Proof of Work (PoW) or Proof of Stake (PoS) for consensus.

Chapter Contents

5.1 Introduction

In this chapter, two methods of implementing distributed DSM are put forward. We refer to these two methods as the *Method 1* implementation of distributed DSM and the *Method 2* implementation of distributed DSM. Each implementation method is then paired with two consensus protocols (PoW and PoS), one at a time. This results in four approaches to achieving distributed DSM using blockchain in energy networks as shows in Table 5.1.

A distributed energy network starts with a certain number of nodes which, in each new planning period (eg. day), must implement the DSM algorithm

Tab. 5.1: The four approaches to distributed DSM.

Approach	DSM implementation method	Consensus Protocol
Approach 1	Method 1 implementation	Proof of Work
Approach 2	Method 2 implementation	Proof of Work
Approach 3	Method 1 implementation	Proof of Stake
Approach 4	Method 2 implementation	Proof of Stake

in a distributed manner and arrive at the local solution for each node and a global solution for the network. These local solutions and the global solution become the transactions recorded in that planning period's block. Nodes then use either the PoW or the PoS consensus protocol to determine the node that gets the right to mine this block and add it to the blockchain.

While the proposed approaches can be applied to any energy network requiring distributed DSM, below we present an example use case to better aid understanding.

Use Case: Assume there are 25 houses in a neighbourhood all connected to the same transformer. The overall neighbourhood profile for the current planning period (in our case, current day) is the global solution and the household profiles or schedules for the current day are the local solutions. The baseload profile and the parameters of the energy storage of each house are the local node data. Given a desired profile, this blockchain network of 25 full (+miner) nodes must form a consensus regarding the local and neighbourhood profiles for that day.

Each distributed DSM approach consists of 4 stages. The blockchain network is initialized during Stage 0. This stage is run only once at the start of the blockchain network or when a new node joins the network. A DSM algorithm such as profile steering is implemented in Stage 1. We present two methods of executing the profile steering algorithm over a distributed network in Stage 1. At the end of Stage 1, each node has the final local solutions or power profiles of all the nodes and the global solution or aggregate profile.

In Stage 2, nodes use a consensus protocol to determine the winning or mining node. In Stage 3, only the mining node gets to add the local and global solutions to the blockchain as part of a new block. Other nodes receive the new block once it has been mined. The new block records the local and

global profiles that all the network nodes have agreed to execute for the current planning period.

The execution of Stages 1 to 3 completes one model run. Each model run sets the local profiles and the global profiles for one planning period. Therefore, each model run represents one entire planning period. Figure 5.1 shows the four stages in distributed DSM using blockchain.

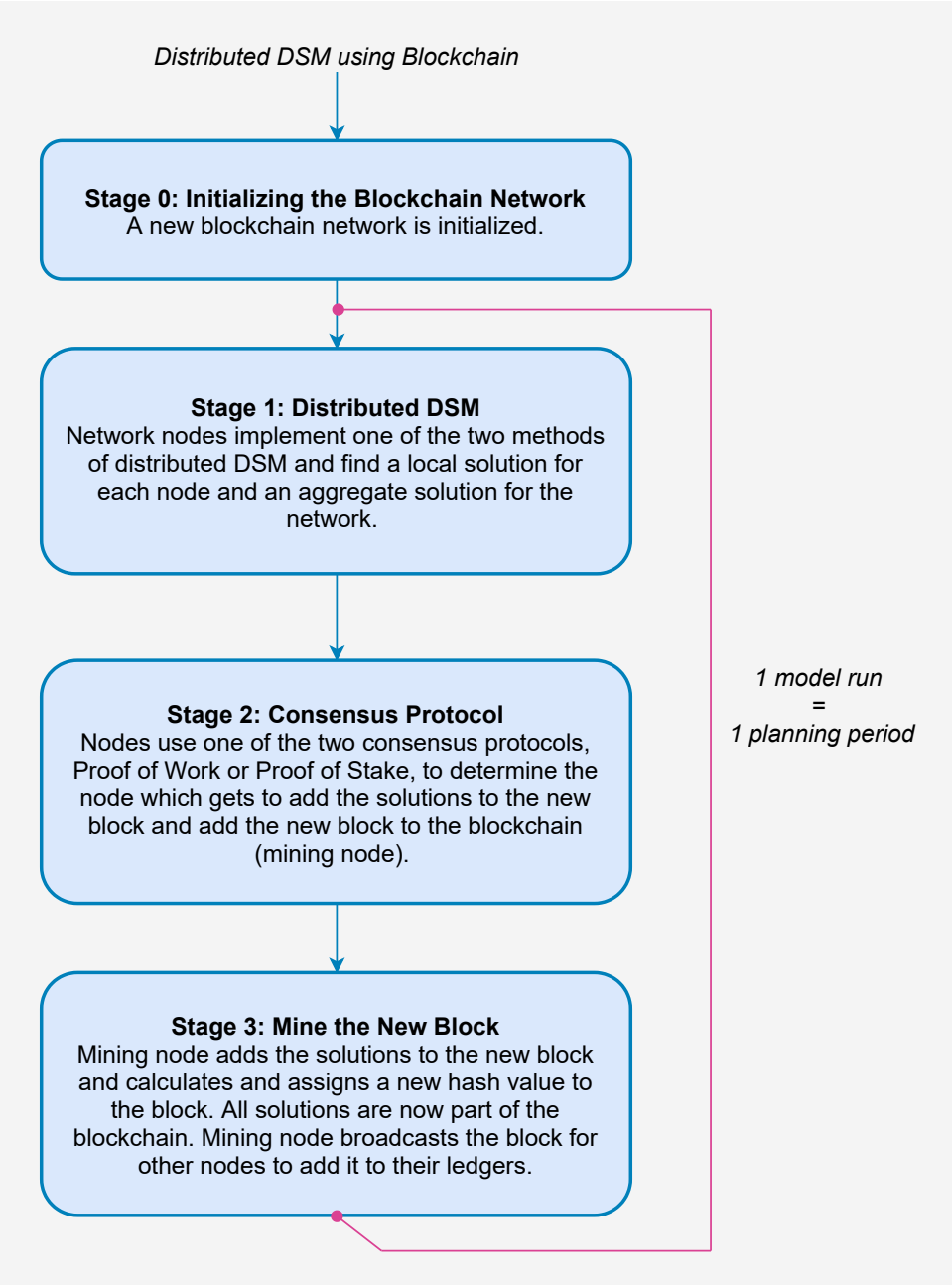


Fig. 5.1: The four stages in distributed DSM using blockchain.

5.2 Stage 0: Initializing the Blockhain Network

Stage Objective: To register new nodes onto the blockchain network and sign the smart contract, create the network and local ledgers and create the genesis block.

5.2.1 Network Structure

We envision the communication layer of a micro-grid or smart-grid as a blockchain network. Each prosumer in the grid is equivalent to a full node in blockchain. Each prosumer has its own local ledger. All prosumers or nodes are connected to a network manager node. The network manager maintains the network ledger and holds (and enforces) the Smart Contract. Figure 5.2 shows the blockchain network structure for the distributed DSM models.

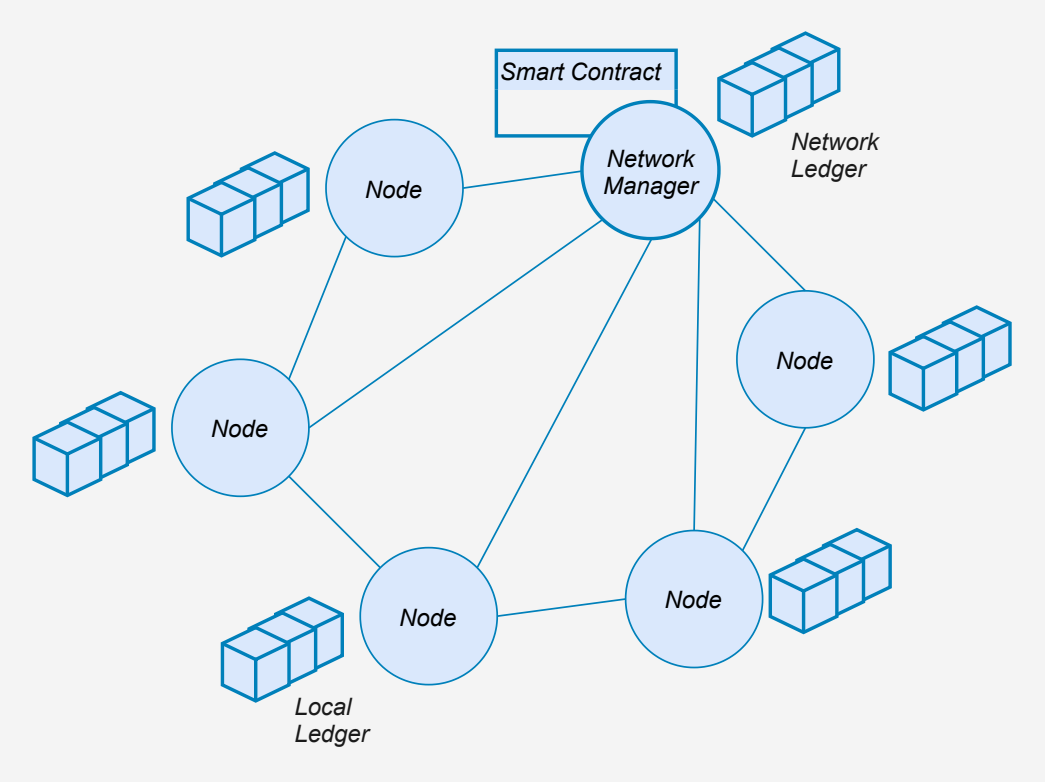


Fig. 5.2: Structure of the distributed DSM blockchain network.

Smart Contract:
The Network Manager holds the Smart Contract and enforces it. Prior to

the network becoming active, nodes decide upon the contents of the smart contract. The Smart Contract contains rules to be followed during the consensus rounds. New nodes must sign the contract upon entering the network. Participation in the network is subject to these rules and disobeying these rules can lead to penalties also recorded in the Smart Contract. The Smart Contract may also contain rules to set the desired profile.

Refer to Chapter 4, Subsection 4.2.4 for a detailed explanation of the concept of Smart Contracts in blockchain.

Genesis Block and Network Ledger:
The Network Manager creates the Genesis Block. This block doesn't contain any transaction information and only holds the timestamp of the beginning of the network. This block is created to give a starting hash value for miner nodes for future blocks. The Network Manager adds the Genesis Block to its copy of the blockchain. The blockchain copy stored in the Network Manager is called the Network Ledger.

Local Ledger:
The blockchain copy stored in each node is called the Local Ledger. After registering on the network, nodes download the latest blockchain copy from the Network Manager (the current contents of the Network Ledger).

Refer to Chapter 4, Subsection 4.2.2 and Section 4.2 for a detailed explanation of the Genesis Block and Distributed Ledgers respectively.

Register Initial Stake (PoS only):
In the PoS model, nodes register their initial stake with the Network Manager.

Refer to Chapter 4, Section 4.4 for an explanation of Initial Stake in PoS. We explain this further in Subsection 5.4.2 in this chapter.

5.3 Stage 1: Distributed DSM

Stage Objective: To execute a DSM algorithm over the distributed network to determine local profiles and the global profile for the current planning period.

Profile Steering (PS) is used as the DSM algorithm in the proposed approaches. We present two methods by which profile steering can be implemented in a distributed manner. In a distributed network, all nodes perform the same actions independently. In the following explanation, *Node A* represents any node in the network. The term *Node A* is used only to make the explanation more objective. The actions performed by Node A are simultaneously performed by all other network nodes too.

5.3.1 Method 1 implementation of distributed DSM

The steps in Method 1 are as follows:

1. Node A shares its baseload profile, starting battery profile and battery parameters for the current planning period with all other network nodes. The battery parameters are the power rating, initial State of Charge (SoC) and the required final State of Charge of Node A's battery. Node A also acquires the same initial or starting data of all the nodes in the network. Nodes use the Network Manager as a data collection point, uploading and downloading their initial data to and from a shared repository in the Network Manager.

   Node A exits this step with the baseload profile, starting battery profile and battery parameters of all the network nodes for the current planning period.

2. Node A calculates the starting local solutions or profiles of all the nodes. The local profile of a node is the sum of its baseload profile and battery profile.

   Node A exits this step with the starting local profiles of all the nodes.

3. Node A performs PS iterations with the data of each node.
   In each PS iteration, Node A finds the candidate profile for each network node. This is possible since Node A has the baseload profile and battery information of all the nodes. So Node A can optimize the battery profile of each node and find its candidate profile (baseload profile + candidate battery profile). Node A then calculates the improvement achieved by

each node and selects the node with the highest improvement. Node A updates the local profile of that node with its candidate profile and discards the candidate profiles of the other nodes. This is one PS iteration.

> *Node A optimizes the battery profiles of all the network nodes and exits this step with the final local profiles of all the nodes. If every node executes the profile steering algorithm with the same battery optimization algorithm, then every node will exit this step with the same final local solution set.*

4. Node A calculates the final aggregate solution. The aggregate solution or aggregate profile is the sum of the final local solutions.

5.3.2 Method 2 implementation of distributed DSM

The steps in Method 2 are as follows:

1. Node A calculates it starting local profile. The starting local profile is the sum of the baseload and starting battery profiles for the current planning period. Node A shares its starting local profile with the other nodes and acquires their starting local profiles. Again, the network manager is used as a data collection point.

> *Node A exits this step with the starting local solutions or profiles of all the network nodes.*

2. Node A calculates the starting aggregate profile. The starting aggregate profile is the sum of the starting local profiles.

> *Node A exits this step with the starting aggregate profile of the network.*

3. Nodes execute PS iterations. In Each PS iteration, Node A finds its candidate battery profile and candidate local profile. Node A calculates the improvement in its local profile and shares its improvement with the other nodes. Node A acquires the improvements achieved by the other nodes in the current PS iteration.

Now that Node A has the improvements of all the nodes, it checks which node achieved the highest improvement in this iteration.
If Node A achieved the highest improvement, then it replaces its current local profile with its candidate local profile and calculates the new aggregate profile. Node A then calculates and shares the delta profile. The delta profile is the difference between the previous aggregate profile and the new aggregate profile.
If Node A didn't achieve the highest improvement, then it waits for the node that did, to broadcast the delta profile and updates the aggregate profile with that delta profile.

> *Node A exits each PS iteration with an updated aggregate profile. After all the PS iterations are completed, Node A has the final aggregate profile and only its own final local profile.*

4. Node A shares its own final local profile with all the network nodes and acquires their final local profiles.

> *Node A exits this step with the final local profiles of all the nodes and the final aggregate profile of the network for the current planning period.*

Figure 5.3 visualizes both implementation methods of profile steering.

5.3.3 Differences between the two methods

Data Privacy

The critical difference between the two methods lies in the nature of the data being shared between nodes. In Method 1, nodes share their local information (their baseload profiles, battery profiles, battery parameters) with other nodes, thus risking the data privacy of the nodes. Sharing of battery and baseload information is required since each node has to find the candidate profile for every other node.

In Method 2, nodes do not share any local information since each node only finds its own candidate profile and only shares the improvement achieved. Thereafter, nodes only update the aggregate profile based on the shared delta

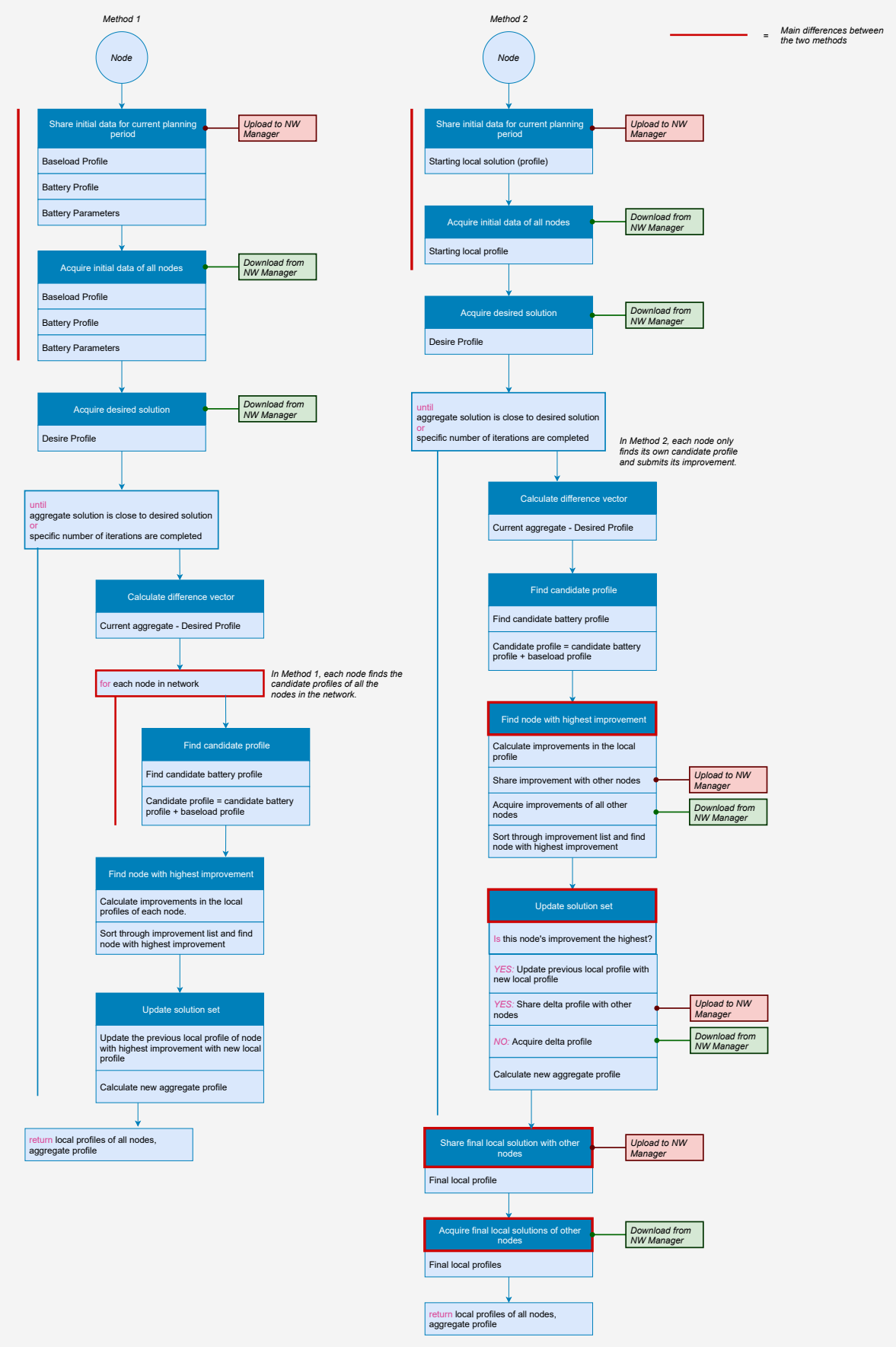


Fig. 5.3: Method 1 and Method 2 implementations of distributed DSM (profile steering).

profile. Only the starting and final local profiles are shared. In terms of data privacy, this makes Method 2 better than Method 1.

Wait Time

On the other hand, in Method 1, data sharing only happens once at the start of the planning period. Each node shares its baseload and battery information for the current planning period. Thereafter, no data sharing occurs between the nodes.

In Method 2, data sharing happens once at the start of the PS iterations, twice per PS iteration and once at the end of all the PS iterations. The first instance of data sharing is for the nodes to share their starting local solutions with each other. During each PS iteration, nodes share their improvements and the node with the highest improvement shares the delta profile. After all the PS iterations are completed, nodes share their final local profiles.

If each instance of data sharing is accompanied by a wait period, the total waiting time then becomes much higher in Method 2 in comparison to Method 1. In Method 1, the wait time is zero during the PS iterations since all iterations are locally performed by each node and no data sharing takes place between nodes. This gives an advantage to Method 1 over Method 2 in terms of speed of execution.

The aspect of Data Privacy is further discussed in Chapter 7, Subsection 7.1.1. In Chapter 6, Section 6.2, we analyse the run time of the two methods.

Each node exits Stage 2 with the list of local profiles or schedules of all the nodes and the aggregate profile of the network for the current planning period. The local profile of each node is the transaction (equivalent to financial transactions in cryptocurrency).

5.4 Stage 2: Consensus Protocol

Stage Objective: To determine winning node that shall mine the new block with the Stage 1 local and global profiles.

For transactions to be confirmed, they must be added to a block and that block must be assigned a hash value derived from the previous block's hash value. Given the singular nature of the blockchain, nodes must engage in a

consensus competition to decide the winning node that gets to execute this operation. Given below are detailed explanations of the implementation of the two consensus protocols in the proposed models.

5.4.1 PoW Consensus

Nodes follow the below steps in PoW consensus.

1. **Create Draft Block**: Nodes first create the draft block. A draft block is created by putting the block index number, the block content (local profiles) and previous block hash value together.
2. **Pick Nonce Factor**: Each node picks a random nonce 32-bit nonce factor from 0 to 4294967295 ($2^{32} = 4294967296$).
3. **Append Nonce Factor**: Nodes append the nonce factor to the block content and the previous block hash value.
4. **Calculate Hash Value**: Each node calculates the hash value of this combined data. The hashing algorithm currently in use is the SHA256 algorithm. The SHA256 algorithm inputs the combined data and gives a 256 bit hex string as output. This is the hash value.
5. **Is Calculated Hash <= Target Hash Value (THV) ?**: Nodes check if their calculated hash value is less than or equal to the PoW THV set by the Network Manager. For example, if the NW Manager sets the THV to 2^{250} (also stated as *6 zeroes to the left*), then a node is successful only if its calculated hash value is less than or equal to 2^{250}.
6. **(If valid) submit for verification**: If the calculated hash value is indeed less than or equal to the target, then that node has found a valid hash value and a corresponding valid nonce factor. The node then submits the draft block, hash value and nonce factor to the NW manager for verification.
7. **(If invalid) try again**: If the hash value is greater than the target, then it is invalid. Nodes check if a PoW winner has already been declared by the NW Manager. If not, they try finding a valid hash again (Steps 2 to 5 again). If yes, then nodes exit the PoW consensus stage.

8. **Time Critical Process**: Most importantly PoW is a time critical process. The challenge is not only to find a valid hash-nonce pair but also to do it first. The NW Manager will verify submissions as they come in. The first submission verified by the NW Manager is declared as the winner and the consensus round is stopped.

Figure 5.4 shows the find-nonce-hash-value loop nodes execute in this consensus protocol.

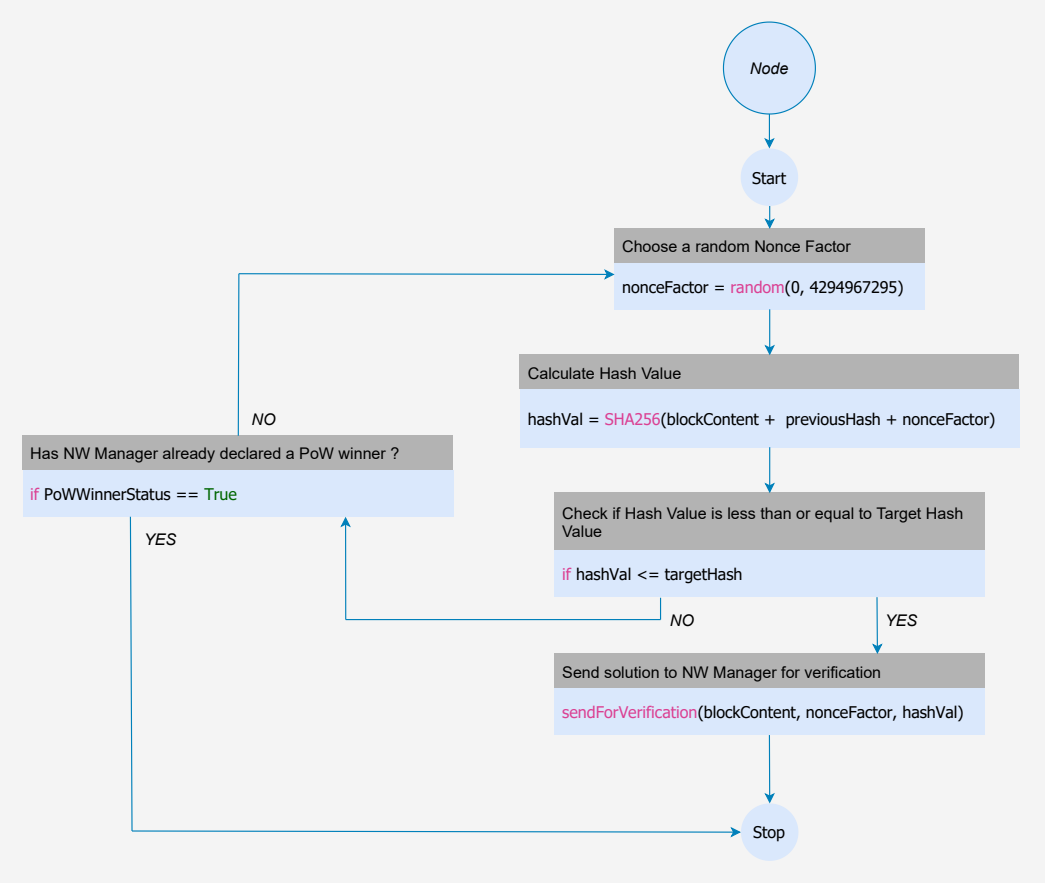


Fig. 5.4: Steps in attempting to find a valid nonce-hash value pair.

The node that submits a valid nonce-hash pair in the shortest time to the NW Manager is the PoW winner. This node gets to mine the new block in Stage 3.

5.4.2 PoS Consensus

Nodes follow the below steps in PoS consensus.

1. **Register initial stake**: Nodes execute this step in Stage 0 when entering the network. There are many parameters that may be considered as initial stake by a PoS-based distributed DSM network. For example, such a network may just ask for a direct financial deposit and treat that as initial stake. Or a PoS network may ask nodes to pay a certain pre-paid energy bill and take that as its initial stake. A PoS network may ask nodes to register their energy storage capacity and consider that as the initial investment or ownership stake of the nodes. In the models implemented in this thesis, we use the last option.

   Nodal battery capacity is treated as nodal initial stake.

2. **Submit Total PS Improvement**: Each node calculates the total improvement in its local profile that it has been able to achieve over all the PS iterations for the current day. It submits this total improvement to the Network Manager.

3. **Find PoS Winner**: The NW Manager collects the total PS improvements of all nodes and declares the node that achieves the highest PS improvement as the winning node.

4. **Give PoS Reward**: The NW Manager calculates the PoS reward for the winning node from its submitted PS improvement. The PoS reward for a node is proportional to its total PS improvement. This reward can represent a financial payback the winning node receives in proportion to the quality of its participation in the profile steering algorithm for the day. We also call this *Stake Recovery* (part of Initial Stake *recovered* from the network).

Figure 5.5 visualizes the steps in PoS based consensus for distributed DSM.

Conventional PoS vs PoS for DSM

In conventional PoS as implemented in cryptocurrency, the winning node is simply the node that has the highest initial financial deposit in the network. A nodes chances of winning block forging rights is proportional to and dependent only on its initial stake. The PoS reward is a fixed sum of cryptocurrency.

In the PoS for DSM implementation proposed in this thesis, combinations of both the initial stake and the daily nodal DSM performance are considered

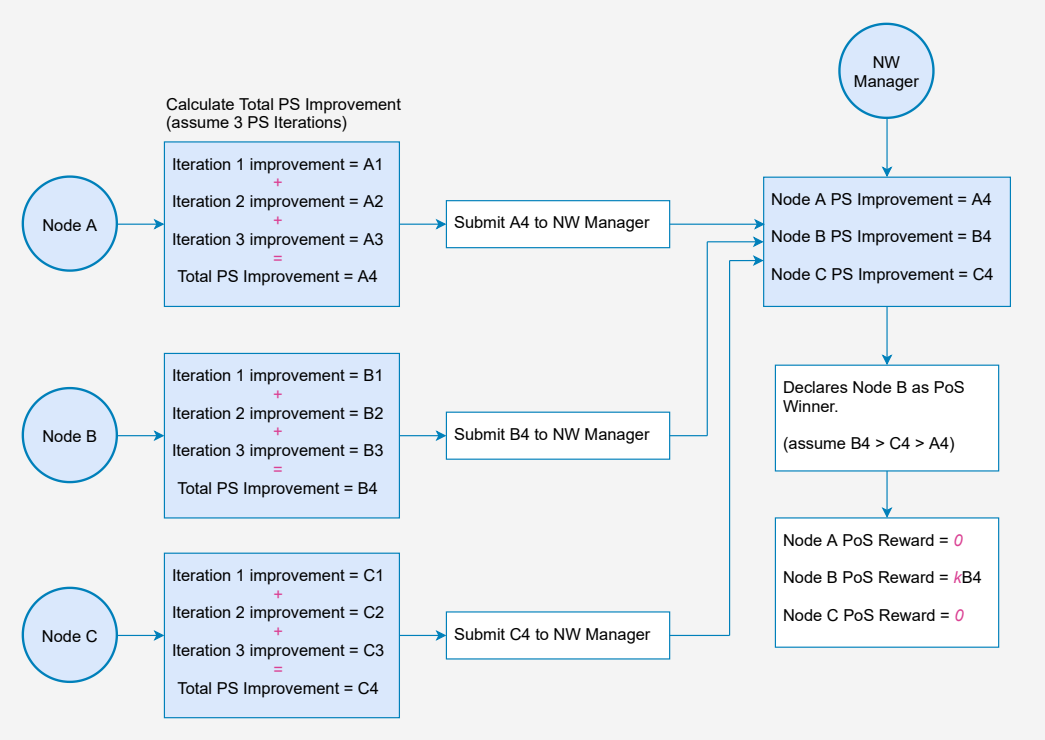


Fig. 5.5: Steps followed in PoS based consensus.

while awarding block forging rights and calculating the value of the PoS Reward. In the above explanation, only daily nodal DSM performance (PS Improvement) was used for simplicity in explanation.

In Chapter 6, Section 6.5, we study the Stake Recovery parameter under different combinations of initial stake and DSM performance.

5.5 Stage 3: Mining (or Forging) the Block

Stage Objective: To calculate the current hash value and update the blockchain.

The winner of the PoW or PoS round is the node that gets to mine the new block. Mining the new block means creating a new hash value for the current block from the block content and the previous block hash value. Once a new hash value is created, the new block has been formally added to the blockchain. The miner node broadcasts this new block to all the nodes in the network so that each node can add the new block to their own copy of the blockchain.

In PoW, this operation is called mining. The miner node is the node that submits the valid nonce-hash value pair in the shortest time. This valid hash value is also taken as the new block's hash value.

In PoS, this operation is called forging. The forging node uses the SHA256 algorithm to create a new hash value from the block content and the previous block hash value.

Figure 5.6 visualizes this final stage in distributed DSM using blockchain.

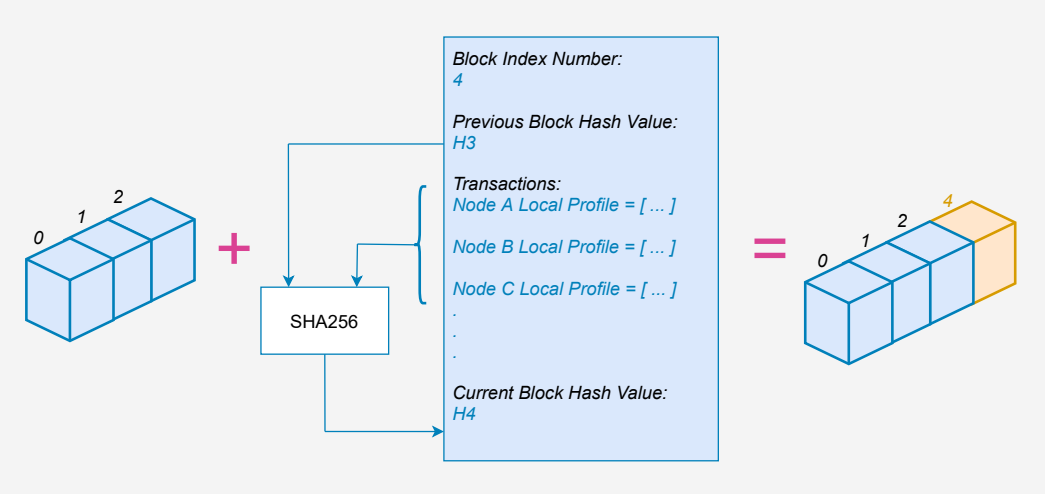


Fig. 5.6: Mining or Forging a new block.

5.6 Discussion

The primary difference between a decentralized and a distributed network is the distribution (or lack thereof) of DSM roles. In decentralized DSM, some nodes are coordinator nodes and some nodes are child nodes. Each node has certain operations to perform based on its allotted role. In distributed DSM, there is no such role division. All nodes must perform all steps in a certain algorithm. Each node is a 'complete' node. This makes distributed DSM more Crash Fault Tolerant (CFT) than decentralized DSM.

Decentralized DSM networks are hierarchical networks. Distributed networks because of their lack of specific role allotment do not have any hierarchy. This makes distributed DSM networks more Byzantine Fault Tolerance (BFT).

An important aspect of consensus protocols is the incentive they give nodes to participate in the network. Some consensus protocols also incentivize good quality network participation. Either way, the implementation of consensus protocols is that they can give an avenue to further encourage good quality performance from nodes in during the DSM stage.

CFT, BFT and participation incentives are further discussed in Chapter 7.

Figure 5.7 shows the stages applied in all four approaches to distributed DSM using blockchain. Two implementation methods of distributed profile steering are combined with two consensus protocols from blockchain.

Chapter Summary

1. There are 4 stages in distributed DSM using blockchain. Stage 0 is run only once at the beginning of the blockchain network and Stages 1 to 3 constitute one model run.
2. In Stage 0, nodes copy the ledger from the network and sign the smart contract.
3. In Stage 1, nodes execute the DSM algorithm. Profile steering is used here and two methods of implementing it in a distributed manner are put forward.
4. In Stage 2, a consensus competition determines the winning node that gets to add the local profiles as transactions to the new block.
5. In Stage 3, the winning node adds a hash value to the current block.

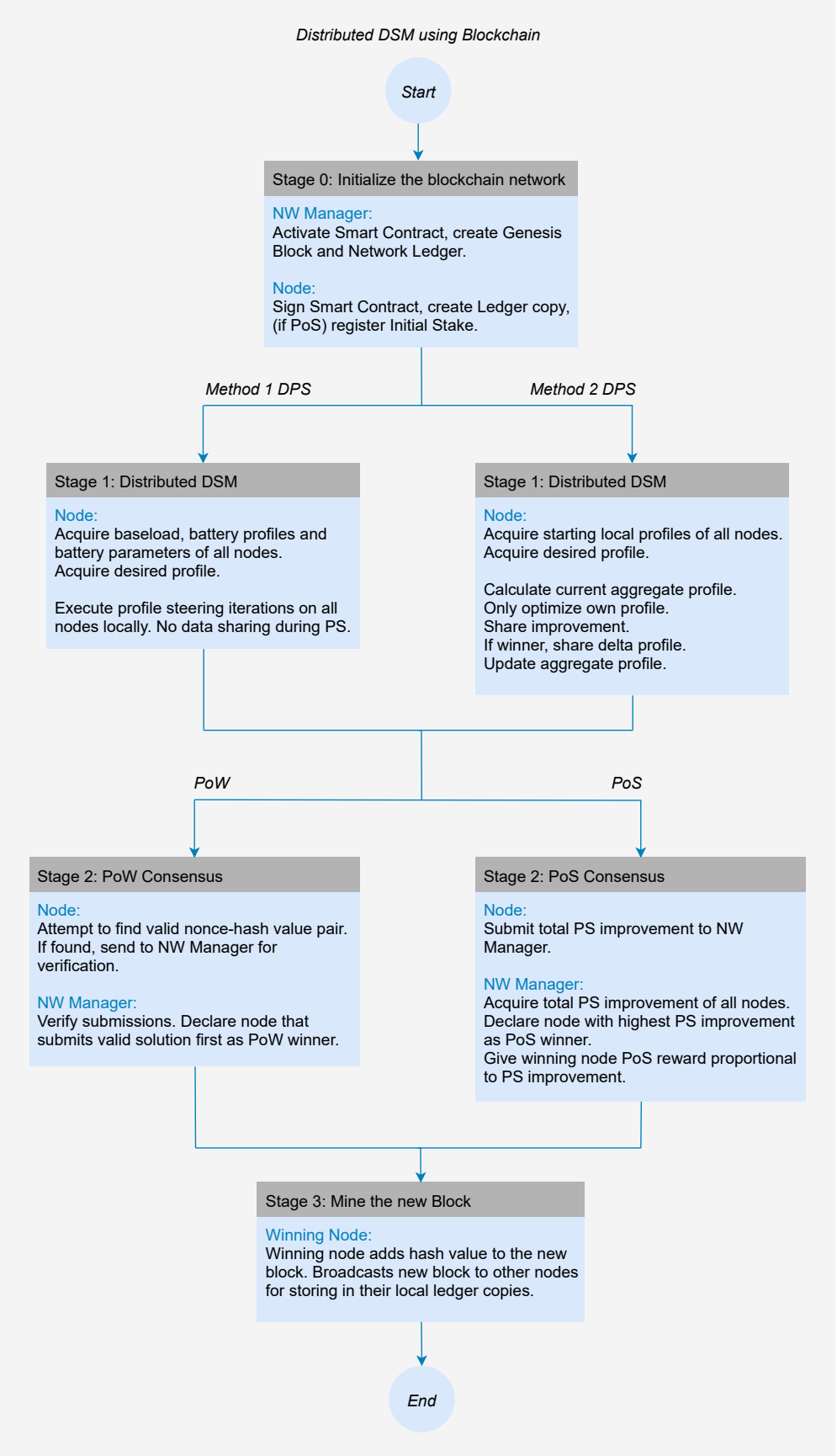


Fig. 5.7: Stages in all four distributed DSM approaches

6 Analysis

Chapter Objective: In this chapter we present various simulations performed on the four approaches to distributed DSM using blockchain. These simulations aid in performing a comparative analysis of the two consensus protocols and the two implementation methods of the profile steering algorithm.

Chapter Contents

- DPS: Load Duration Curves (6.1)
- Time Performance (6.2)
- Scalability of the models (6.3)
- PoW: Winner Distribution (6.4)
- PoS: Stake Recovery (6.5)
- Energy Consumption (6.6)

Table 6.1 summarizes the 4 combinations or approaches of the two consensus protocols and the two distributed profile steering implementations.

Tab. 6.1: Summarizing approaches to distributed DSM with blockchain.

Distributed Profile Steering (DPS)	Consensus Protocol	Description
Method 1 implementation	PoW	In Stage 2, nodes acquire baseload and battery data of all nodes and optimize local profile of all nodes. In Stage 3, nodes use PoW to achieve consensus.
Method 2 implementation	PoW	In Stage 2, nodes only optimize local profile and share improvements and the delta profile. In Stage 3, nodes use PoW to achieve consensus.
Method 1 implementation	PoS	In Stage 2, nodes acquire baseload and battery data of all nodes and optimize local profile of all nodes. In Stage 3, nodes use PoS to achieve consensus.
Method 2 implementation	PoS	In Stage 2, nodes only optimize local profile and share improvements and the delta profile. In Stage 3, nodes use PoS to achieve consensus.

6.1 DPS: Load Duration Curves

About Load Duration Curves Simulations (6.1)

1. Models for both methods were run for 184 iterations (184 days) with data from 25 nodes (houses). The dataset has 96 samples per day (1 sample every 15 minutes). This results in a total of 184 x 96 = 17664 points.
2. These 17664 baseload and final aggregate(neighbourhood) profile points were then arranged in an ascending order and plotted. Since both load duration curves are exactly the same and overlap completely, markers are used to distinguish the two curves.
3. Battery capacity of 13500 Wh (Tesla Powerwall 2) is used for all nodes.

Figure 6.1 shows the load duration curves for Method 1 DPS implementation and Method 2 DPS implementation. The curves show that both methods implement profile steering in a distributed manner correctly.

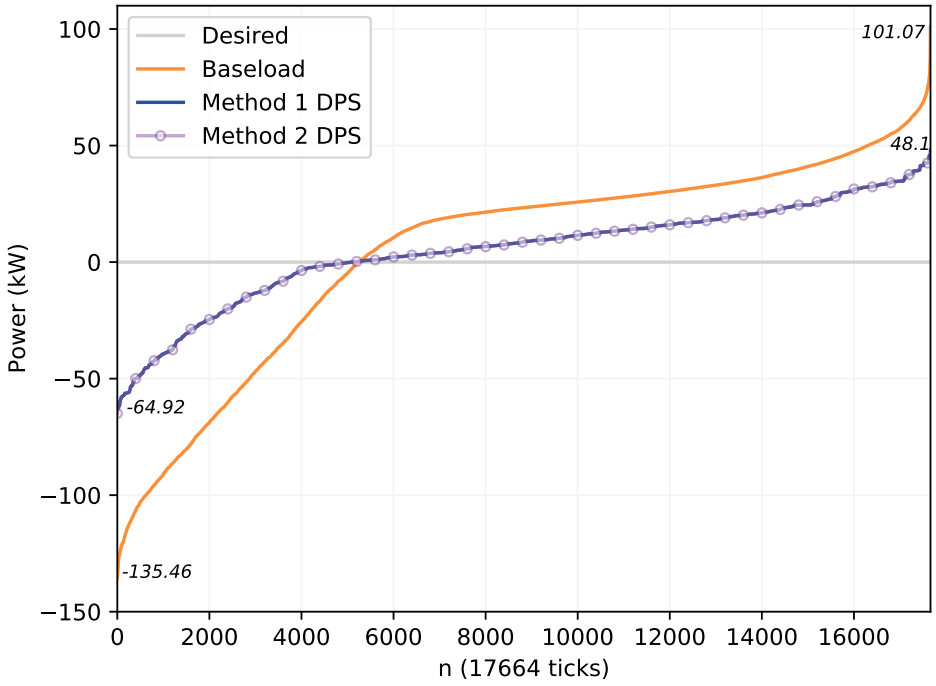


Fig. 6.1: Load duration curves for both implementation methods of distributed profile steering.

6.2 Time Performance

Table 6.2 shows the average time per iteration for each step to execute. The fastest timings in each step are marked for easy reference.

About Time Performance Simulations (6.2)

1. These readings are also recorded from the same simulation run executed for the load duration curves. The simulation network consisted of 25 nodes with the daily baseload profile of each node consisting of 96 samples per day.
2. Simulation was run for 184 iterations or 184 days. The timings presented in Table 6.2 are the average of 184 values (iterations).
3. THV used in PoW consensus was 2^{250}. This target value was chosen since with this value, all 184 PoW iterations were getting completed in a reasonable amount of time (184 iterations in approximately 3 hours).
4. The PS algorithm was also run for 184 days with 1 coordinator and 25 nodes (houses) to record timings for the sake of comparison with the timings of the distributed PS implementations.

Tab. 6.2: Time performance of the four approaches. All values are average of 184 iterations.

	Distributed Profile Steering			Consensus			
Approach	Setup time (s) A	PS Iterations time (s) B	Total PS time (s) C = A+B	PoW/ PoS time (s) D	Declaring Winner time (s) E	Total Consensus time (s) F = D+E	Run Time per Iteration (s)
PS	1.5045	0.8802	2.3847	-	-	-	2.4691
Method 1 DPS, PoW	10.7802	2.6203	13.4005	0.3276	0.3624	0.6900	14.2687
Method 2 DPS, PoW	3.3818	36.3786	39.7604	0.1759	0.8049	0.9808	40.7557
Method 1 DPS, PoS	11.2545	2.4764	13.7309	0.0272	4.5114	4.5386	18.9536
Method 2 DPS, PoS	3.3453	36.5367	39.8820	0.0152	3.9289	3.9441	43.8490

6.2.1 Inferences

1) Setup time:

a) *Meaning:* The initial setup time is the time required for a node to acquire the starting data for a model run.

b) *Observation:*

$$Setup\ Time(s):\ Method\ 1\ >\ Method\ 2$$

c) *Reason:* In DPS Method 1 implementation, each node has to find local solutions for all nodes. It has to optimize the local profile of all nodes. So each node has to acquire baseload and battery profiles and starting battery parameters of all nodes. In DPS Method 2 implementation, each node only has to find its own local solution and update the global solution in each PS iteration. So each node only has to download the starting local profiles of all nodes. This reduces the setup time in the Method 2 implementation of DPS.

2) PS Iterations:

a) *Meaning:* This is the time taken for a node to execute all the PS iterations given the starting data from the Initial Setup step.

b) *Observation:*

$$PS\ Iterations\ Time(s):\ Method\ 2\ >\ Method\ 1$$

c) *Reason:* Because of their local nature of PS iterations, DPS Method 1 implementation is found to be faster than DPS Method 2 implementation. There is no data sharing between the nodes during the PS Iterations step in DPS Method 1 implementation. Each node finds the local solutions for all nodes and doesn't upload or download any data since it already has all the starting data of all nodes. In DPS Method 2 implementation, each node finds its own local solution and then uploads the improvement value. Then it acquires the improvement values of all other nodes and uploads (or downloads) the delta profile. These data communication processes

occur in each PS iteration thereby increasing the total time for PS iterations.

d) *Discussion:* The PS Iterations time in DPS Method 2 implementation illustrates the network latency problem of distributed networks. Take for example the communication process of downloading improvement values. With 25 nodes used in the simulation model, each node has to wait for 25 improvement values to be uploaded by each node in the shared repository. This leads to a large wait time per PS iteration. One solution implemented in commercial networks to reduce wait times is strict timeout periods. If a node is unable to submit its values within the timeout period, then its previous iteration value or some default value may be considered. Another method is to setup a local and therefore dedicated network for DSM.
Yet another method is to use file systems that are better geared to handle concurrent write operations. This means going for file systems that have better input buffer designs so that a node's first write operation is successful.
Finally, nodes can be designed to store information locally and the network manager must then access and collate this information. While this may reduce the chances of lost write operations to a shared repository, it leads to the question of nodal data privacy. Having strict timeouts seems to be the most common solution implemented in commercial network applications.

3) POW/POS time:

a) *Meaning:* This is the time taken for nodes to engage in the consensus protocol and make the necessary submissions (nonce factor-hash value in PoW and total PS improvement in PoS) to the Network Manager.

b) *Observation:*

$$PoW/PoS\ Time(s):\ PoW\ Consensus\ >\ PoS\ Consensus$$

c) *Reason:* In PoW consensus, nodes must engage in computationally heavy work to find a valid nonce factor-hash value pair and submit

that to the Network Manager. In PoS consensus, each node must only sum up and report its total PS improvement.

d) *Discussion:* The PoW consensus time is recorded at a THV of 2^{250}. This value was chosen because, at this target value, a 184-day simulation would get completed in a reasonable amount of time (approximately 3 hours). The PoW time depends on the THV which in turn determines Network Difficulty. PoW consensus time is proportional to the Network Difficulty set by the Network Manager. Changing the THV changes the PoW consensus time.
The concept of Network Difficulty is discussed in the next Subsection (6.2.2).

4) Declaring Winner time:

a) *Meaning:* This is the time taken by the Network Manager to find and declare the consensus round winner based on the submissions.

b) *Observation:*

$$Declaring\,Winner\,Time(s):\ PoS\,Consensus\ >\ PoW\,Consensus$$

c) *Reason:* In PoW, nodes compete to find a valid nonce factor and submit it to the Network Manager. This is a time-critical process. The Network Manager verifies nonce factors and hash values *as they are submitted*. The moment it finds the first valid submission, it declares the node that made that submission as the winner and awards it the block mining rights.
On the other hand, in PoS, the Network Manager decides the winner as the node that achieves the highest total PS improvement. To do this, the Network Manager has to *wait* for all the nodes in the network to calculate and submit their total improvement values. This increases the waiting time in PoS on the Network Manager's end. In PoW, there is no waiting time on the Network Manager's end.

d) *Discussion:* Large waiting times in PoS can be solved by having strict timeouts periods within which nodes must make submissions. Else they lose the opportunity to be considered in the consensus

round and may lose the chance to win forging rights and receive the PoS reward.

5) **Total Consensus time:**

a) *Meaning:* Total time taken for the consensus round. Sum of consensus time and winner declaration time.

b) *Observation:*

$$Total\ Consensus\ Time(s):\ PoS\ Consensus\ >\ PoW\ Consensus$$

c) *Reason:* This result depends on the PoW time which in turn depends on the THV used. The PoW time may be increased or decreased and this inversely affect network resiliency or security. The PoW time depends on Network Difficulty (see 6.2.2).

6) **Total Run time:**

a) *Meaning:* Overall run time per iteration of each combination.

b) *Observation:* Slowest to Fastest:

i. $Method\ 2\ DPS,\ PoS$ (43.8490s = 17.75*PS_t)

ii. $Method\ 2\ DPS,\ PoW$ (40.7557s = 16.50*PS_t)

iii. $Method\ 1\ DPS,\ PoS$ (18.9536s = 7.67*PS_t)

iv. $Method\ 1\ DPS,\ PoW$ (14.2687s = 5.77*PS_t)

v. PS (2.4691s = PS_t)

c) *Reason:* The higher timings of the DPS Method 2 implementation as compared to the DPS Method 1 implementation affect the total run times. Additionally the THV of 2^{250} makes the PoW time lesser than the PoS time. The THV may be set lower (more PoW time) to improve network security and Byzantine Fault Tolerance (BFT) of the network (see 6.2.2).

When compared to the PS implementation, the fastest distributed DSM implementation (Method 1 DPS, PoW) is 5.77 times slower and the slowest distributed DSM implementation (Method 2 DPS, PoS) is 17.75 times slower.

6.2.2 Network Difficulty in POW models

Network Difficulty decides how much time the entire network takes to find a valid nonce factor and a valid hash value. The network's difficulty can be changed by altering the THV. In Bitcoin, the Network Manager runs an algorithm that sets the THV such that a new block is mined and added to the blockchain approximately every 10 minutes [66]. As new nodes get added to the network, the Network Manager updates the THV to ensure that the 10-minutes rate of block creation is maintained.

In PoW, nodes compete to find a hash value lower than the THV set by the Network Manager. If THV is high (easy target), then the time required to find a valid hash is low and it is said that it's 'easy' to mine a block in that network. If the THV is low (difficult target), then the time required to find a valid hash value is high and it is said that it's 'difficult' to mine a block in that network.

About Network Difficulty Simulations (6.2.2)

1. This simulation was started with a parameter sweep at $n = 100$ resulting in a THV of 2^{100}. The value of n was increased until a value of n at which the average mining time was around 10 minutes (but greater) was found. This was done to achieve at least one point above the 10 minutes ($600s$) mark. This point was found to be at $(n = 239,\ mining\ time = 780.7677s)$. That is 13.0127 minutes at THV = 2^{239}.

2. Then the THV was increased from n = 239 till n = 256 in steps of 1 and the average mining time was plotted. In each case, the PoW simulation was run for 10 days and the mining time was averaged over these 10 iterations. The PoS time was taken from Table 6.2, Section 6.2.

Figure 6.2 shows the time required to find a valid hash value for the PoW models at different hash values. As the THV increases, it gets easier for nodes to find a hash value lesser than the target. Hence the PoW competition ends sooner and the winning node mines the new block sooner. Since PoW uses the SHA256 hashing algorithm, the THV has to be some target less than or

equal to 2^{256}, with 2^{256} being the highest and the easiest target that can be set.

The THV directly affects the PoW time and therefore the Run Time per iteration (Table 6.2, Subsection 6.2.1). In all simulations, we use a THV of 2^{250}. This translates to a network of 25 nodes requiring 0.3276 seconds on average to find a valid hash value.

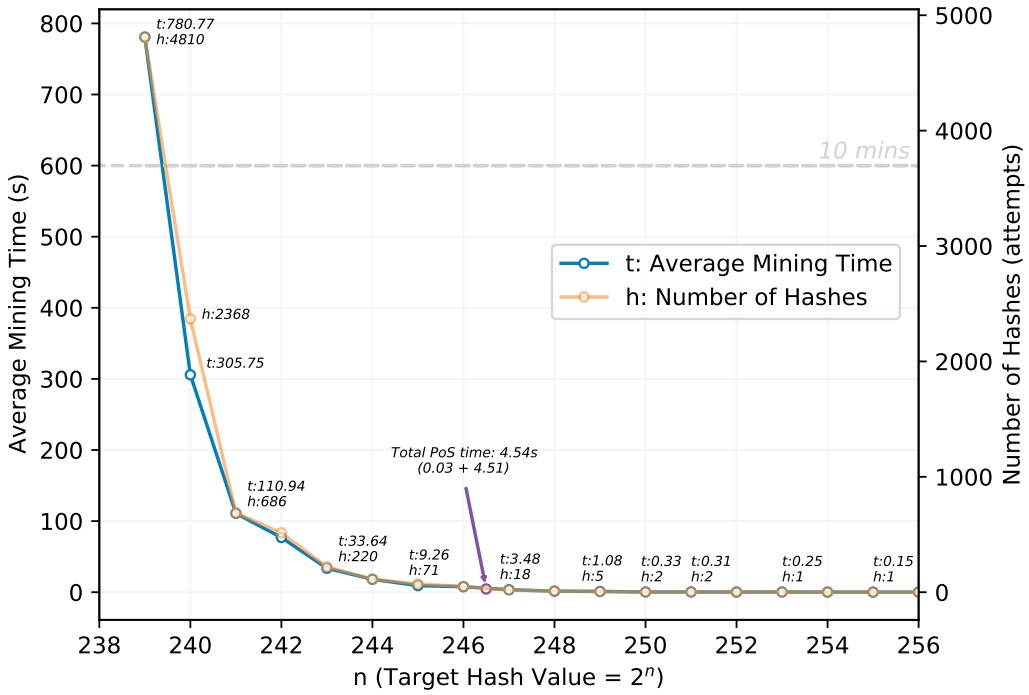


Fig. 6.2: Network Difficulty in PoW. All mining time values are average of 10 iterations. Number of hashes represents the number of hashing attempts made by the winning node to arrive at a valid hash value. Hashes represent the required computational work.

Network Difficulty and BFT

A network's resilience to attacks from malicious nodes is called the Byzantine Fault Tolerance (BFT) level of that network. In blockchain networks, a challenging THV is used for two reasons. One reason is to keep a considerable time interval between consecutive blocks being added to the blockchain. The aim is to give enough time for network nodes to update their local ledger after the previous block has been mined. The second reason is to act as a protection against malicious nodes with high computational capacity attempting to mine fraudulent blocks. The more challenging is the THV, the more energy nodes must expend to mine a valid hash value. A malicious node may end us losing more energy (and hence more financial value) than it could gain

from attempting to mine a fraudulent block in a transparent network. A challenging hash value reduces the Return-On-Investment possibly gained from mining a fraudulent block. This disincentivizes fraudulent mining. A challenging THV gives increased security and BFT at the cost of increased model run time and energy use.

PoW vs PoS consensus time

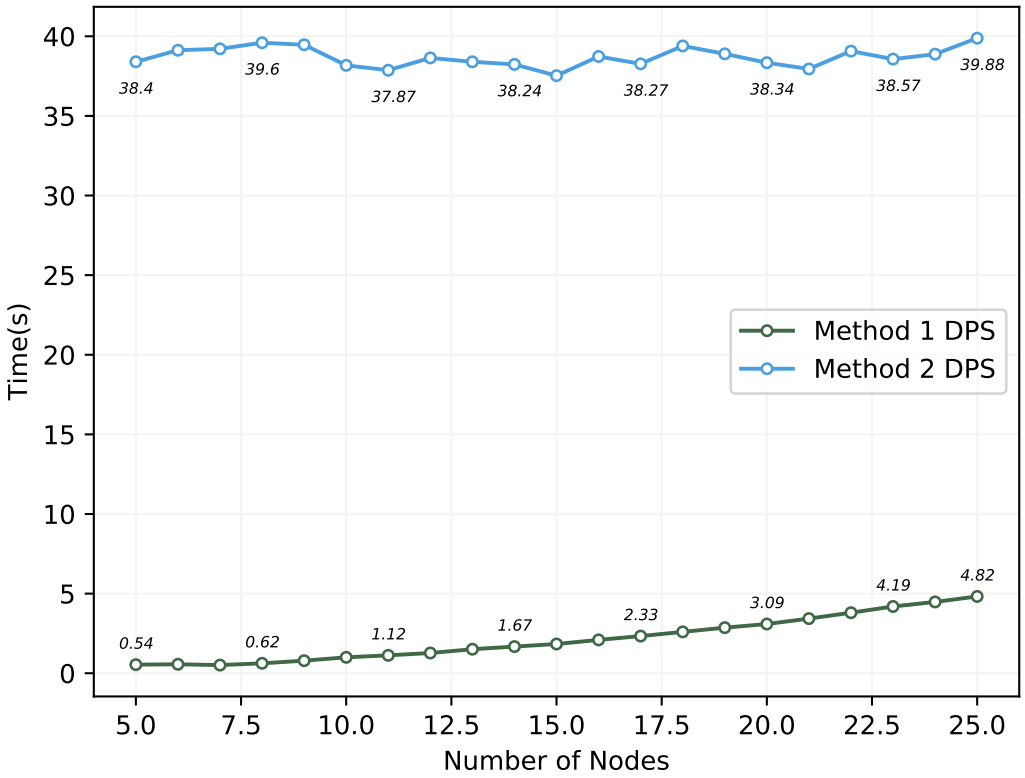
As shown in Figure 6.2, the PoS time for 25 nodes is 4.54 seconds. This also includes the Winner Declaration Time. In PoS the task that consumes the most amount of time is the time taken by the Network Manager to wait for all nodes to submit their improvements and then find the node with the highest improvement. In PoW, the more time consuming operation is finding a valid hash value. Therefore the major part of the consensus stage time in PoW is on the end of the nodes and the major part of the consensus stage time in PoS is on the end of the Network Manager. While PoW time can be reduced by setting a higher THV (decreasing Network Security), PoS time can be reduced by having strict timeouts for nodes to submit their improvement.

6.3 Scalability

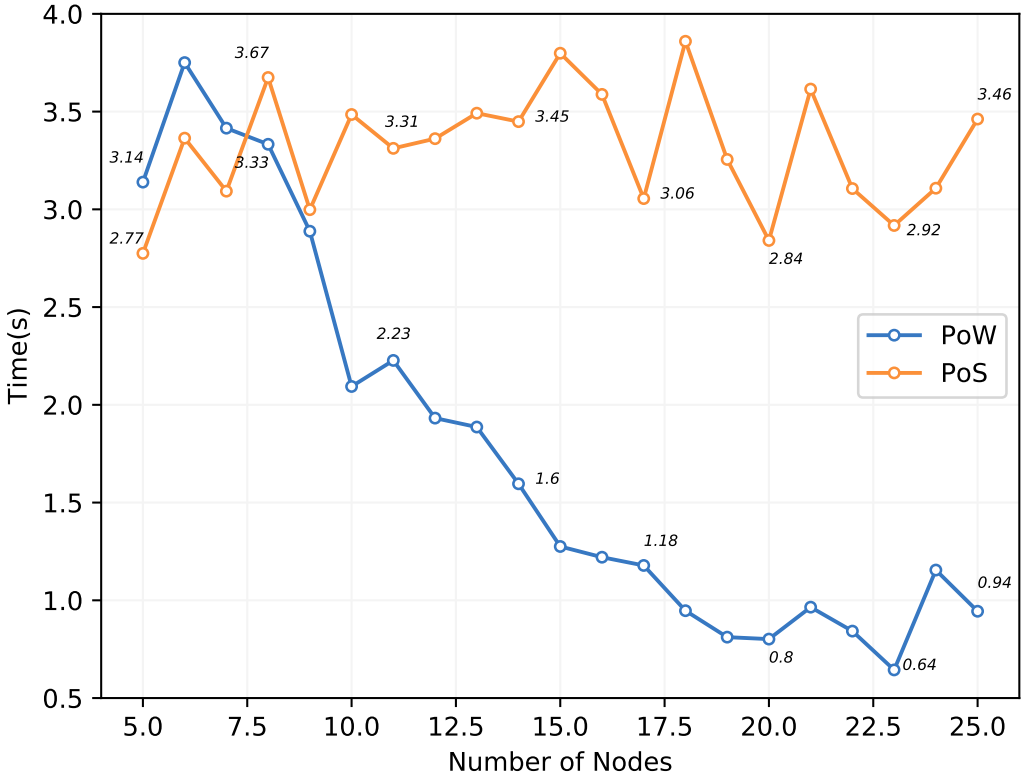
About Scalability Simulations (6.3)

1. Different network sizes from nodes $= 5$ till nodes $= 25$ in steps of 1 were simulated for both implementation methods of distributed profile steering and both consensus protocols.
2. All readings are an average of 31 iterations. In PoW a THV of 2^{250} was used. In PoS, forging rights and PoS reward was awarded on the basis of performance (further discussed in Section 6.5).

Figure 6.3 shows a scalability comparison for both implementation methods of distributed profile steering and both consensus protocols. Figure 6.3a shows the profile steering run time in both implementations of distributed PS for different network sizes. Figure 6.3b shows the consensus round run time for PoW and PoS for different network sizes.



(a)



(b)

Fig. 6.3: Scalability analysis (a) Scalability in DPS methods and (b) Scalability in consensus protocols.

6.3.1 Discussion

In Figure 6.3a, the time taken to complete distributed profile steering can be seen rising steadily in the Method 1 implementation of DPS while in the Method 2 implementation of DPS, the profile steering time stays within the same range. This is because in the Method 1 implementation of DPS, as the network size increases, each node has more starting information to acquire from all other network nodes. In terms of profile steering run time, the Method 2 implementation of DPS is better for larger network sizes than Method 1 given its relatively stable run time.

In Figure 6.3b, the PoW run time decreases with increasing network sizes while the PoS run time, in comparison stays within the same range. As the number of nodes increases, the number of nodes competing to find a valid hash value also increases. This means, in each PoW round, there are more nodes working to solve the mathematical problem. Therefore the PoW solution is found faster in larger networks than in smaller ones.

6.4 Winner Distribution in PoW

6.4.1 Simulations

Winner Distribution refers to the distribution of mining victories among nodes in a PoW network. Winning a PoW round depends on the computational power of the node. One way to simulate different computational capacities for nodes is to set delays within the code. This delay comes into play for that node in each of its attempts to find a valid hash value. The delay value slows down the node during the PoW round and acts as a representation for 'decrease in computational power'. More delay value implies less nodal computational power.

We consider two cases.
Case 1: There are no delays assigned to any of the nodes. Thus all nodes are assumed to have maximum computational power.

Case 2: Nodes are assigned to a speed category and given a delay value in seconds. Case 2 is done to simulate a more realistic network with nodes

having different computational speeds. Table 6.3 shows the different speed categories, the nodes assigned to each category and their respective delay values in Case 2.

Tab. 6.3: Artificial delays assigned to nodes in PoW Winner Distribution: Case 2.

Speed Category	Assigned to nodes	Delay Value(s)
Fastest	0, 1, 2, 3, 4	0.15
Fast	5, 6, 7, 8, 9	0.30
Medium	10, 11, 12, 13, 14	0.45
Slow	15, 16, 17, 18, 19	0.60
Slowest	20, 21, 22, 23, 24	0.75

About Winner Distribution Simulations (6.4)

Both Case 1 and Case 2 simulations were done for 184 days with 25 nodes and a THV of 2^{250}.

Figure 6.4 shows the resultant winner distribution in both cases.

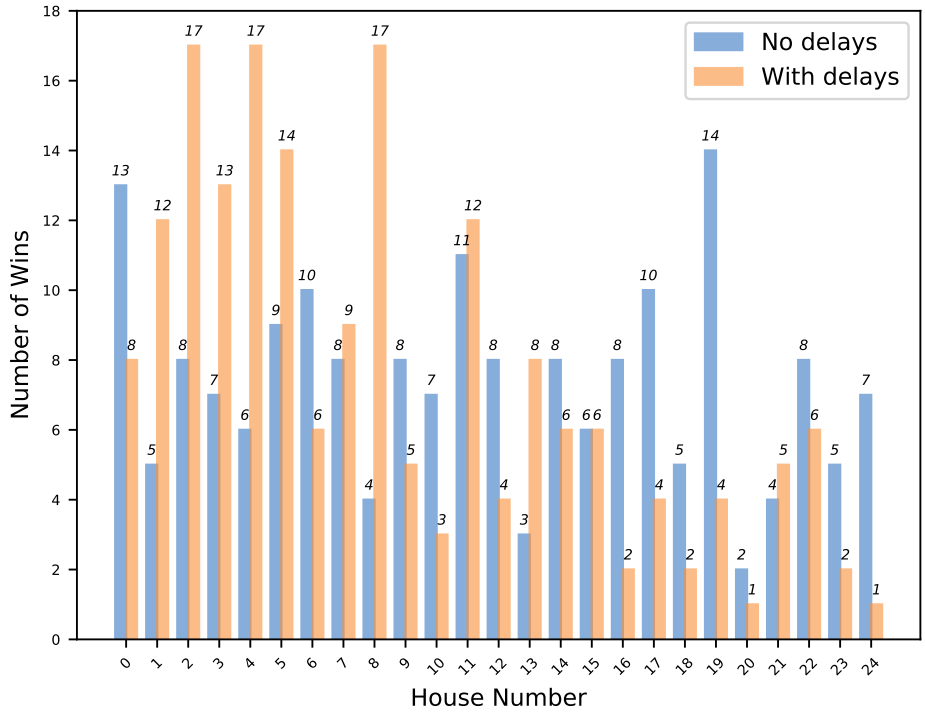


Fig. 6.4: PoW winner distribution in Case 1: No delays VS Case 2: With delays.

6.4.2 Discussion

Figure 6.4 shows that in Case 1, all nodes, given their equal computational speeds have an equal chance of winning the PoW consensus round. No concentration of wins is seen for any house. Any wins gained are a result of the house being successful in its recurrent random operation of finding a nonce.

In Case 2, wins are concentrated amongst the Fastest, Fast and some of the Medium nodes while other Medium nodes, Slow nodes and Slowest nodes achieve lesser wins. This conforms with the intuitive hypothesis that slower (computationally weaker) nodes have a lesser chance of winning than faster (computationally stronger) nodes in PoW. Since nodes randomize the search for finding a valid nonce factor, lesser computational capacity doesn't completely nullify that node's chances of winning, but does decrease its winning probability.

Table 6.4 shows the total wins achieved by each speed category. There is no specific concentration of wins in any category in Case 1. In Case 2, highest total wins is assimilated by nodes of Speed Category: Fastest and lowest total wins is assimilated by nodes of Speed Category: Slowest.

Tab. 6.4: Total wins per speed category in Case 1 and Case 2.

Nodes	Case 1 Speed Category	Case 2 Speed Category	Case 1 Total Wins	Case 2 Total Wins
0,1,2,3,4	No delay	Fastest	39	67
5,6,7,8,9	No delay	Fast	39	51
10,11,12,13,14	No delay	Medium	37	33
15,16,17,18,19	No delay	Slow	43	18
20,21,22,23,24	No delay	Slowest	26	15

6.5 Stake Recovery in PoS

In a PoS based consensus round, the winning node, also known as forging node can be chosen on the basis of *performance* or on the basis of *initial*

stake. *Performance* based choosing means choosing the node that submits the highest PS improvement or the highest DSM performance as the forging node. *Initial Stake* based choosing means the node that makes the highest initial investment in the network has the highest chance of being chosen as the forging node.

Once a node is chosen as the forging node, it creates the new block and receives a PoS reward in return. This PoS reward is a way for the node to recover its initial investment in the network. We call this Stake Recovery.

$$Stake\ Recovered\ =\ \sum PoS\ Reward\ -\ Initial\ Stake$$

The PoS Reward is set by the general formula:

$$PoS\ Reward\ =\ A(initialStake)\ + B(DSMperformance) + C$$

A, B and C are constants.

For profile steering, we consider the nodal battery capacity as its initial stake and daily nodal PS improvement as its DSM performance. This transforms the general PoS reward formula into a PS-specific one as follows:

$$PoS\ Reward\ =\ A(batteryCapacity)\ + B(psImprovement) + C$$

The rules that determine how the forging node is chosen and the value of the PoS reward given to the forging node (values of A, B and C) are agreed upon by all nodes in the network and are part of the Smart Contract.

We simulate Stake Recovery in four different cases of PoS rules. Table 6.5 states the rules in the four cases. The values of A, B and C are set randomly with the only intention being to result in substantial PoS reward values in each case for comparison.

We test these 4 PoS rule cases in two simulation scenarios.

1. *Scenario 1:* All nodes have the same battery capacity and therefore the same initial stake. Battery specifications used in this scenario are given in Table 6.6.

2. *Scenario 2:* Nodes are assigned to an initial stake category ranging from Highest to Lowest. Battery specifications used in this scenario are given in Table 6.7.

About Stake Recovery Simulations (6.5)

Simulations in all 4 cases for both scenarios were done for 184 days with 25 houses.

Tab. 6.5: The rules for choosing forging node and setting PoS reward in all four cases.

Case	Forging Node Rule	PoS Reward Rule
Case 1	*Performance* based Node with the highest total PS improvement shall be the block forging node.	*Reward is proportional to Performance* A=0, B=k, C=0
Case 2	*Performance* based Node with the highest total PS improvement shall be the block forging node.	*Reward is proportional to Performance and Initial Stake* A=k_1, B=k_2, C=0
Case 3	*Initial Stake* based Node with the highest initial stake has the highest chance of being chosen as the forging node.	*Reward is proportional to Initial Stake* A=k, B=0, C=0
Case 4	*Initial Stake* based Node with the highest initial stake has the highest chance of being chosen as the forging node.	*Reward is constant* A=0, B=0, C=k

Tab. 6.6: Initial Stake assigned in Scenario 1.

Battery Product	Battery Capacity (Wh) =Initial Stake	Min Power (W)	Max Power (W)	Target SoC (Wh)	Initial SoC (Wh)	Nodes Assigned
Tesla Powerwall 2	13500	-5000	5000	6750	6750	All nodes

Tab. 6.7: Initial Stake categories in Scenario 2. Sonnen Eco is the *highest* initial stake category and B-Box Pro 5.0 is the *lowest* initial stake category. Initial stake is equal to battery capacity.

Battery Product	Battery Capacity (Wh) =Initial Stake	Min Power (W)	Max Power (W)	Target SoC (Wh)	Initial SoC (Wh)	Nodes Assigned
Sonnen Eco	20000	-8000	3300	10000	10000	0, 1, 2, 3, 4
Pika Harbour 5	14300	-5600	3300	7150	7150	5, 6, 7, 8, 9
Tesla Powerwall 2	13500	-5000	5000	6750	6750	10, 11, 12, 13 ,14
LG Chem RESU10H	9300	-5000	5000	4650	4650	15, 16, 17, 18, 19
B-Box Pro 5.0	5120	-4500	3300	2560	2560	20, 21, 22, 23, 24

6.5.1 Case 1: *Performance* based, only B=k

The node that submits the highest total PS improvement is chosen as the forging node. Formula used for PoS reward is:

$$PoS\ Reward\ =\ 0.01 * (psImprovement)\ |\ B = 0.01$$

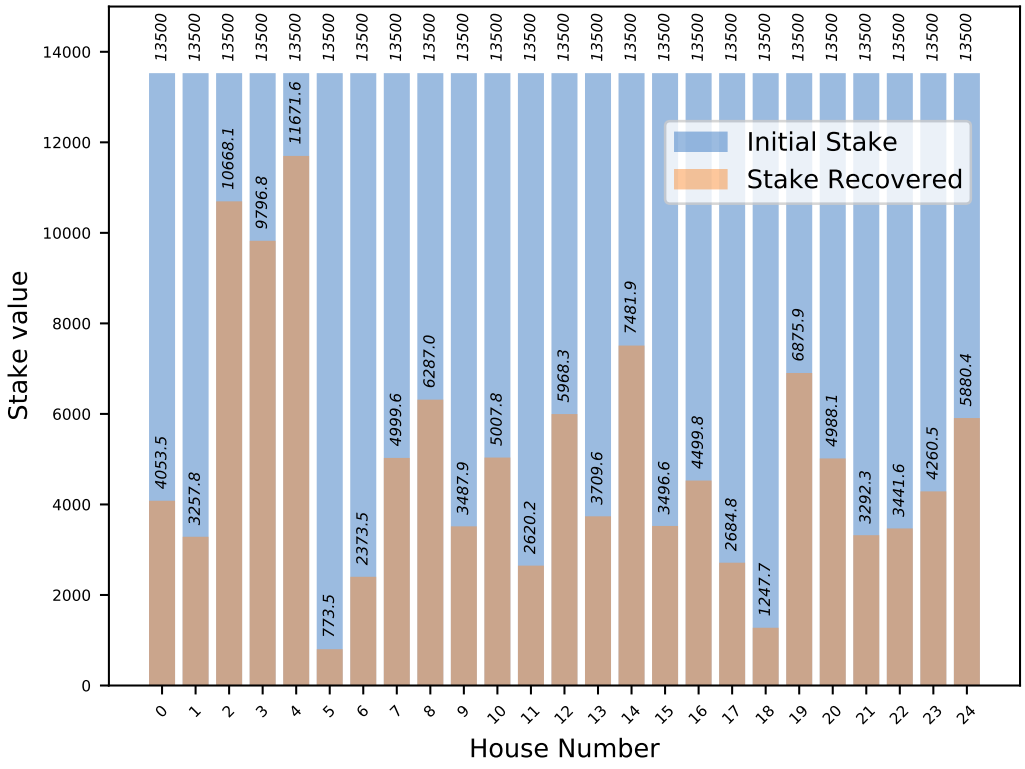
Figure 6.5 shows the stake recovery of all 25 nodes in Scenario 1 and Scenario 2.

6.5.2 Case 2: *Performance* based, A=k_1, B=k_2

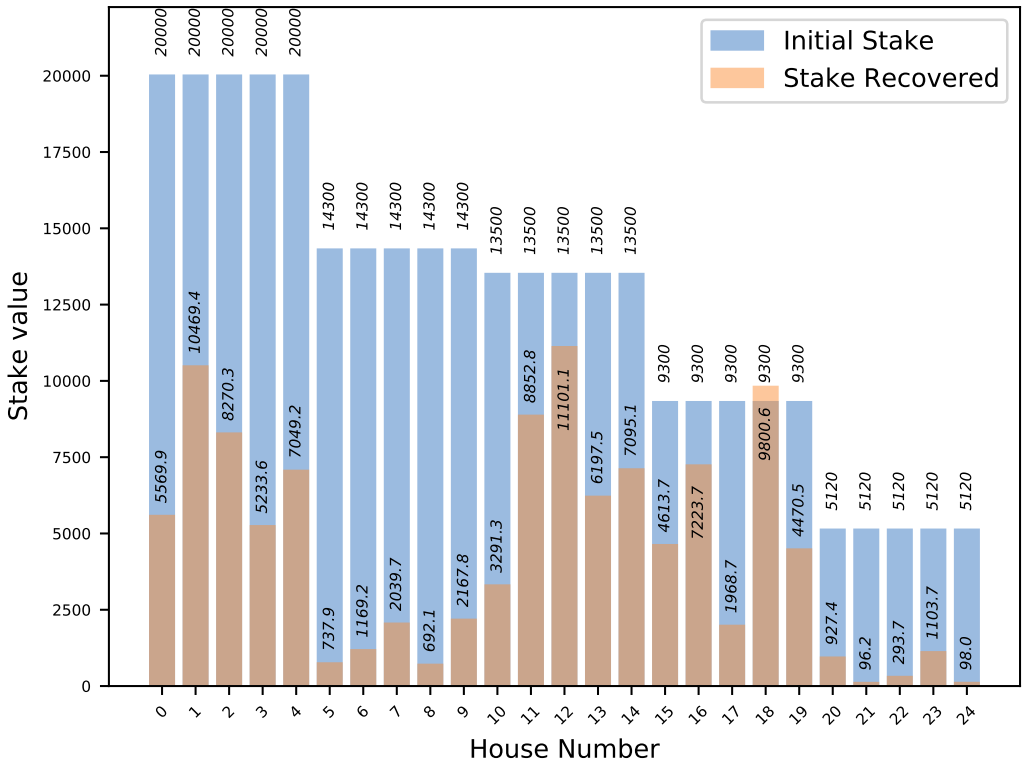
The node that submits the highest total PS improvement is chosen as the forging node. Formula used for PoS reward is:

$$PoS\ Reward\ =\ 0.01*(initialStake) + 0.01*(psImprovement)\ |\ A = 0.01,\ B = 0.01$$

Figure 6.6 shows the stake recovery of all 25 nodes in Scenario 1 and Scenario 2.

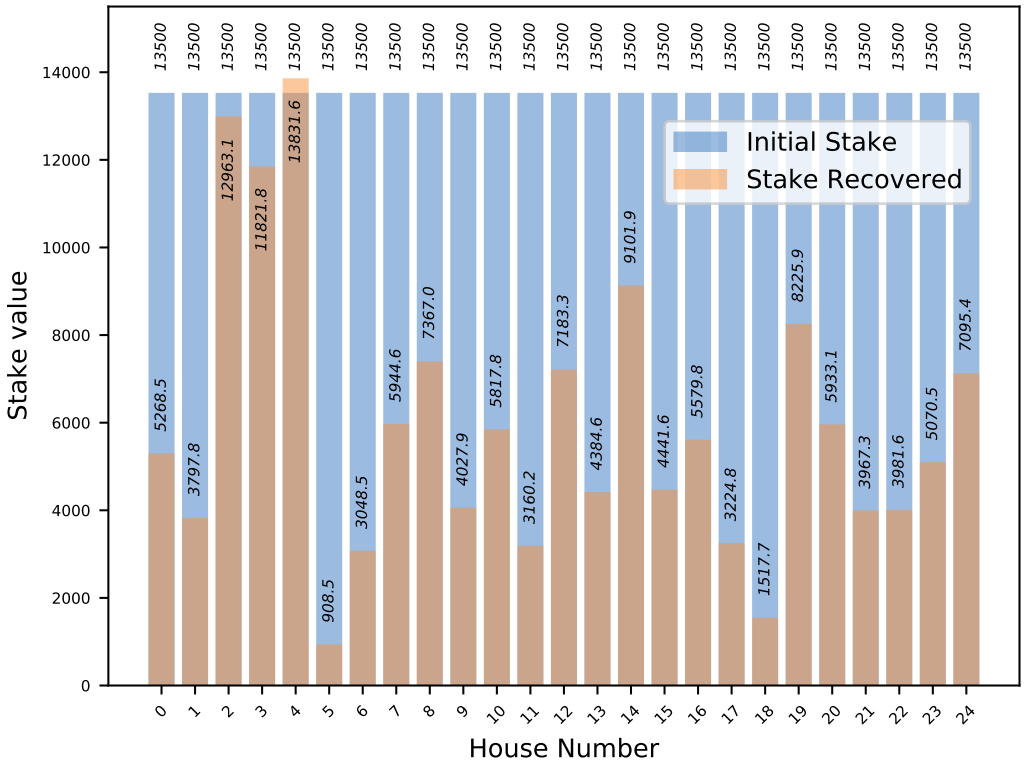


(a)

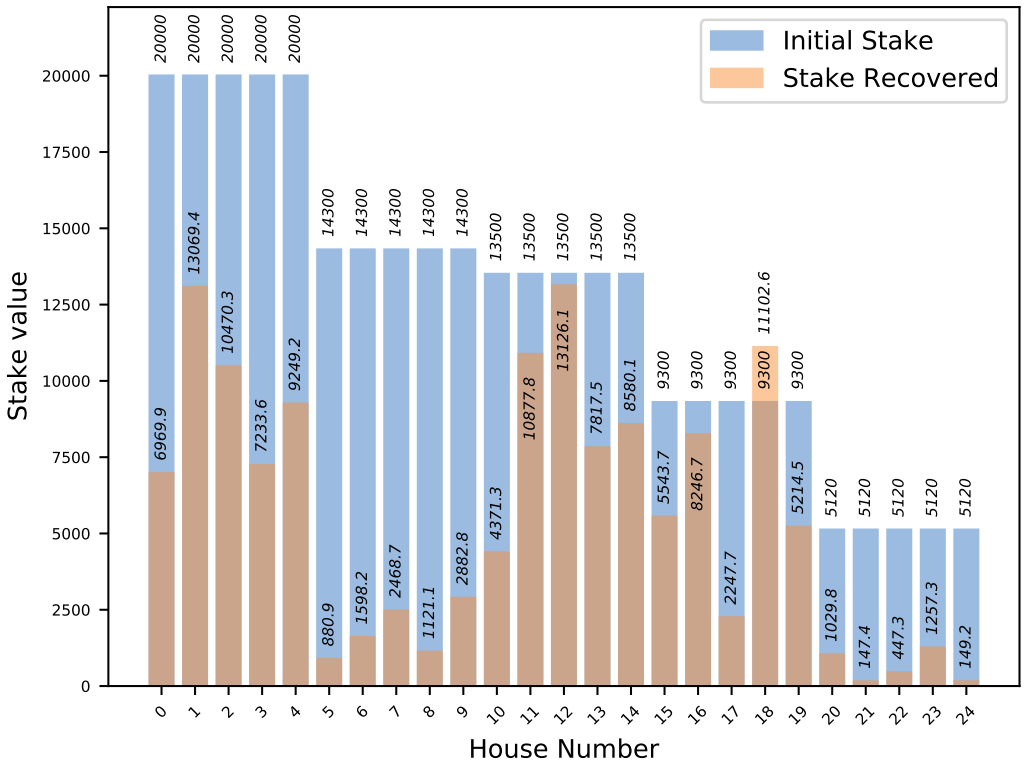


(b)

Fig. 6.5: Stake Recovery of all nodes in Case 1 (a) Scenario 1 and (b) Scenario 2. (Case 1: performance based, only B=0.01)



(a)



(b)

Fig. 6.6: Stake Recovery of all nodes in Case 2 (a) Scenario 1 and (b) Scenario 2. (Case 2: performance based, A= 0.01, B=0.01)

6.5.3 Case 3: *Initial Stake* based, only A=k

The node has the highest initial stake has the highest chance of being chosen as the forging node. Formula used for PoS reward is:

$$PoS\ Reward\ =\ 0.1 * (initialStake)\ |\ A = 0.1$$

Figure 6.7 shows the stake recovery of all 25 nodes in Scenario 1 and Scenario 2.

6.5.4 Case 4: *Initial Stake* based, only C=k

The node has the highest initial stake has the highest chance of being chosen as the forging node. Formula used for PoS reward is:

$$PoS\ Reward\ =\ 500\ |\ C = 500$$

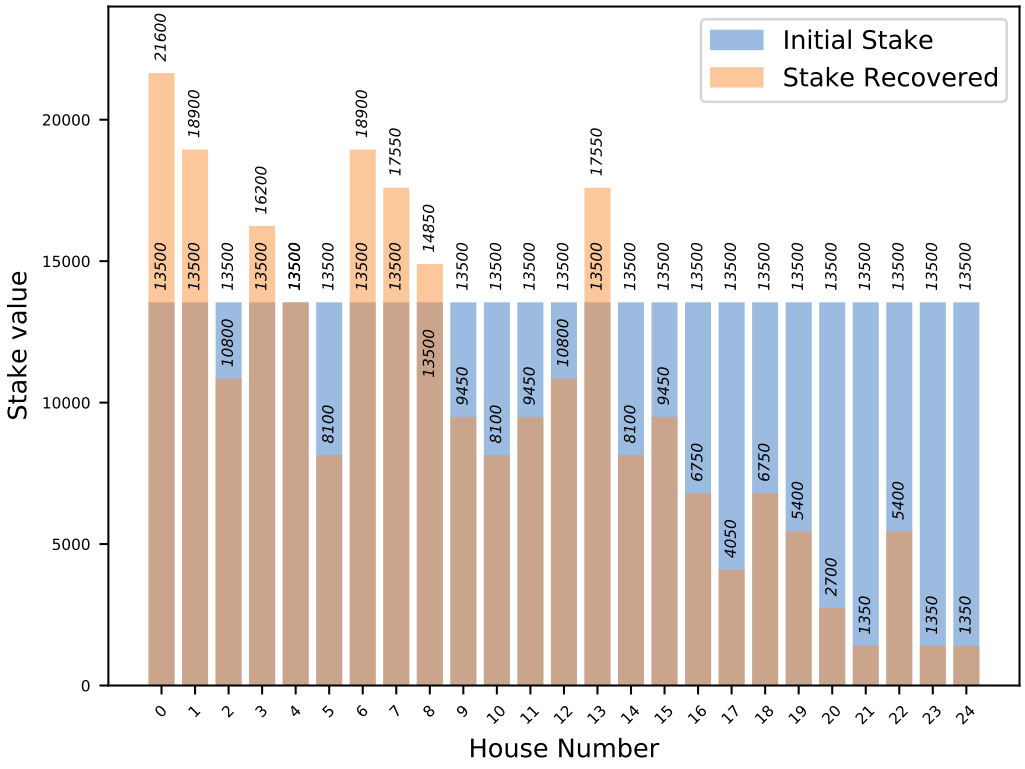
Figure 6.8 shows the stake recovery of all 25 nodes in Scenario 1 and Scenario 2.

Figure 6.9 shows the total stake recovered by each initial stake category of nodes for all four cases for Scenario 1 and Scenario 2.

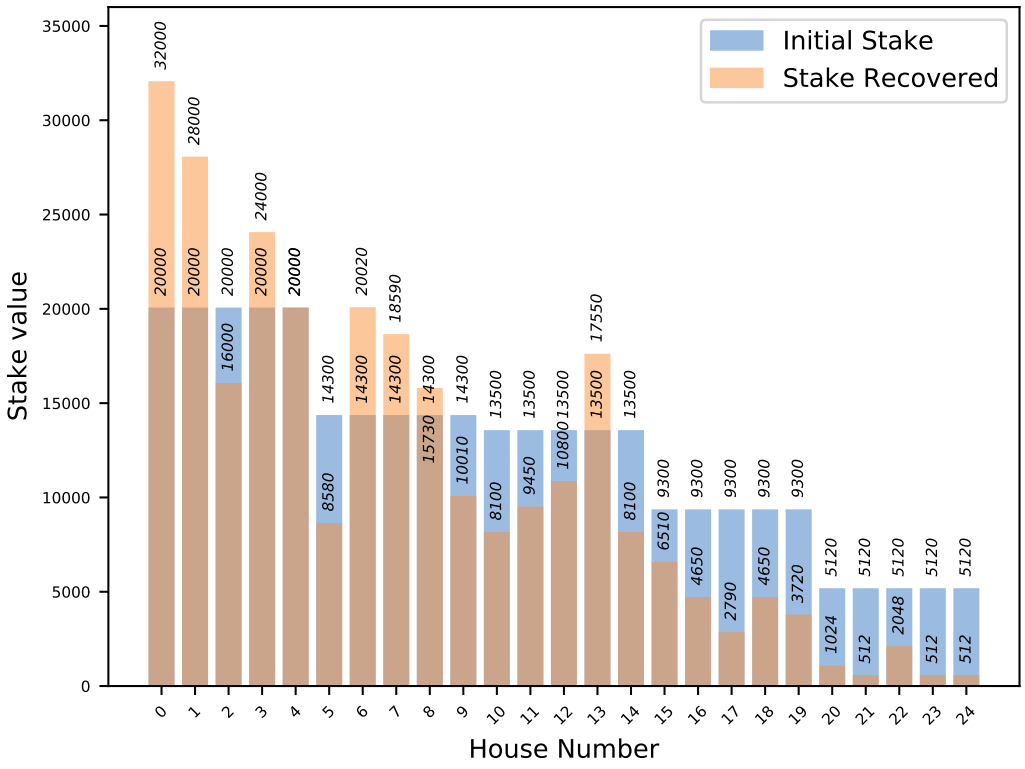
6.5.5 Discussion

In Case 1 (Figure 6.5), the node with the highest PS improvement is awarded block forging rights and the PoS reward is awarded on the basis of DSM performance. In Scenario 2, nodes belonging to battery categories 1, 3 and 4 (Sonnen Eco, Tesla Powerwall 2 and LGChem RESU10H respectively) seem to perform the best. Figure 6.9b confirms the same performance trend in the Total Stake Recovered by each Category for Scenario 2, Case 1. The margin between the total stake recovered by Category 1 and Category 3 nodes is low.

In Case 2 (Figure 6.6), the forging rights are still awarded on the basis of performance but the PoS reward is calculated taking both the initial stake and the PS improvement (performance) in consideration. This gives a boost to the stake recovered by Category 1 nodes (Sonnen Eco) which have the highest

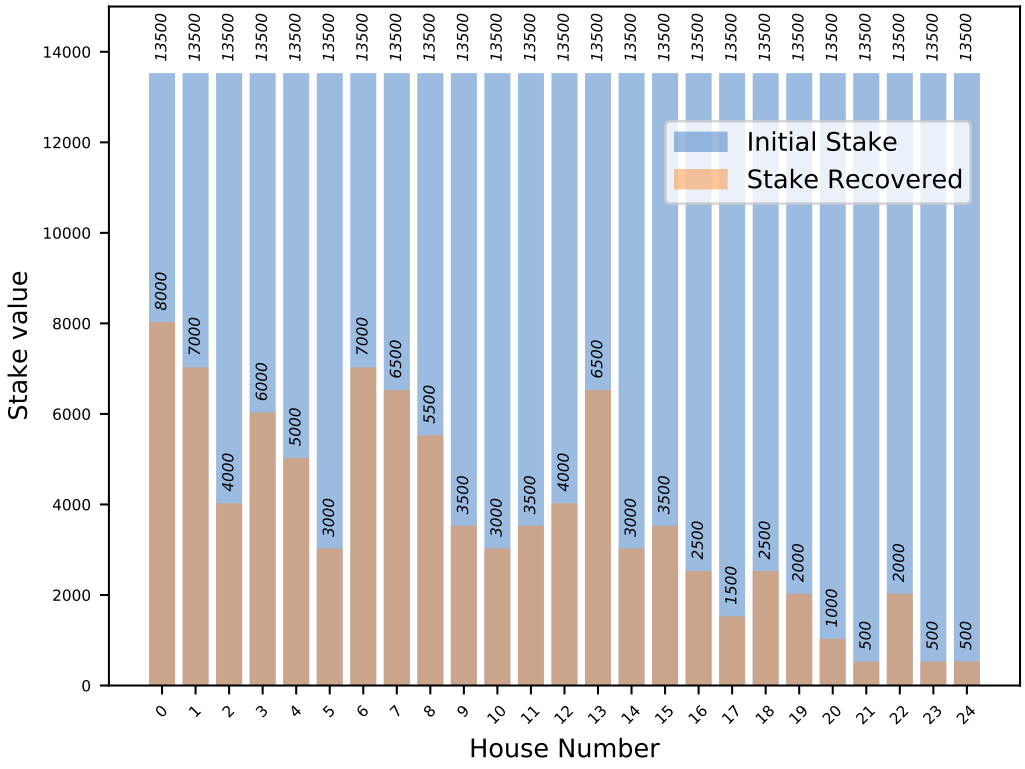


(a)

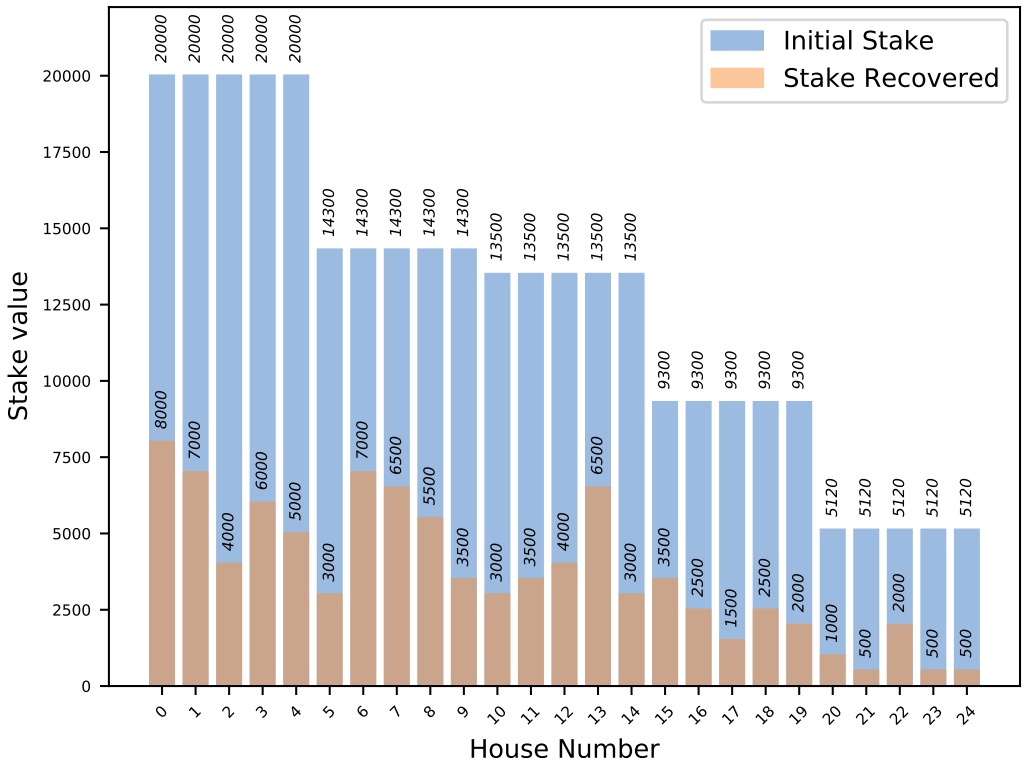


(b)

Fig. 6.7: Stake Recovery of all nodes in Case 3 (a) Scenario 1 and (b) Scenario 2. (Case 3: initial stake based, only A=0.1)

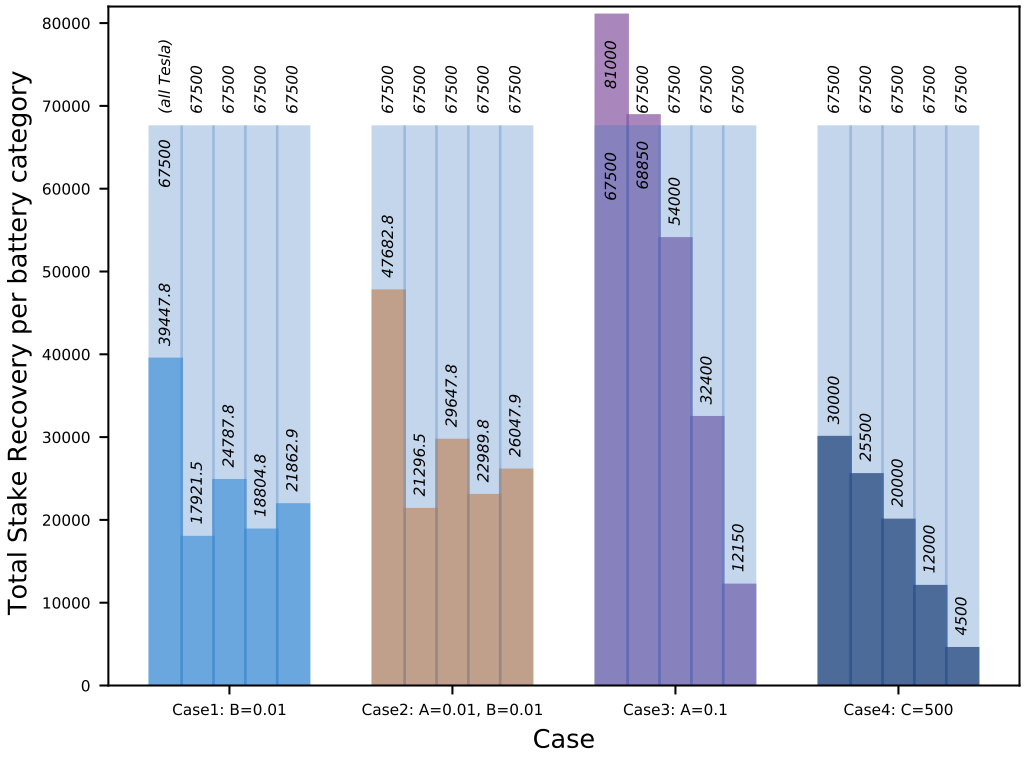


(a)

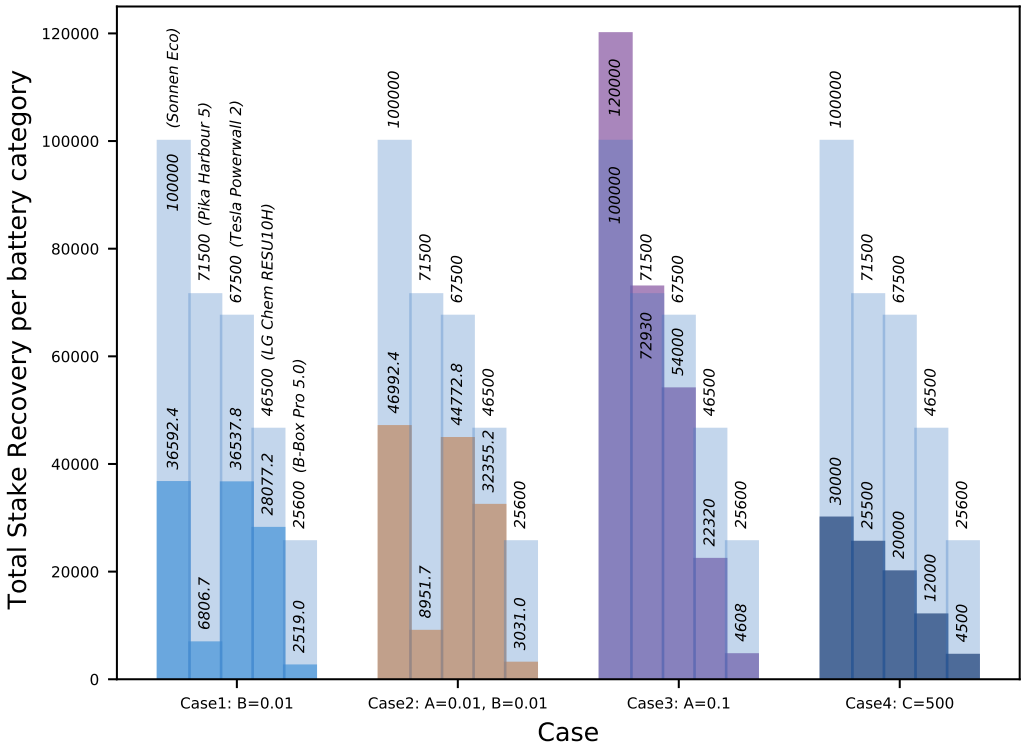


(b)

Fig. 6.8: Stake Recovery of all nodes in Case 4 (a) Scenario 1 and (b) Scenario 2. (Case 4: initial stake based, only C=500)



(a)



(b)

Fig. 6.9: Total Stake Recovery per initial stake category and PoS Cases in (a) Scenario 1 and (b) Scenario 2.

initial stake. As Figure 6.9b shows, the margin of total stake recovered by Category 1 (Sonnen Eco) nodes over Category 3 nodes (Tesla Powerwall 2) is higher in Scenario 2, Case 2 than in Scenario 2, Case 1.

In Cases 3 and 4 (Figure 6.7 and Figure 6.8), both block forging rights and PoS reward are given on the basis of the initial stake without any regard for the daily DSM performance of the nodes. In Case 3, the PoS reward value is dependent on the initial stake (a 10^{th} of the initial stake for each win), while in Case 4, a constant reward of 500 units is given. Given these constant values, Figure 6.9b, reflects the proportional relation between the total stake recovered by each category and the initial stake or battery capacity of that category. Case 3 and Case 4 favour nodes with the higher initial stakes.

6.6 Energy Consumption

About Energy Consumption Simulations (6.6)

1. Simulations for all 4 combinations were run for 184 days with 25 nodes. PoW THV was set to 2^{250}.
2. To calculate the energy consumed, the PyRAPL Python library is used. The PyRAPL library enables measurement of the energy footprint of a host machine when a Python script is run on it. PyRAPL uses the "Intel Average Power Limit (RAPL)" technology that estimates power consumption of a CPU [67].

Table 6.8 shows the energy consumed per iteration by each combination.

Tab. 6.8: Energy consumed per iteration by each combination (going from highest energy consumption to lowest energy consumption). Readings are averages over 184 iterations.

Combination	Energy consumed per iteration (J)	Total Run Time(s)
Method 1 DPS, PoS	845.2531	18.9536
Method 1 DPS, PoW	770.7554	14.2687
Method 2 DPS, PoS	617.2701	43.8490
Method 2 DPS, PoW	542.1310	40.7557

6.6.1 Discussion

The Method 2 implementation of DPS takes more time, due to the higher amount of idle time in comparison to the Method 1 implementation of DPS. However, Table 6.8 shows that the Method 2 implementation of DPS is more energy efficient than the Method 1 implementation of DPS. Additionally, PoW consensus with a difficulty of THV $= 2^{250}$ consumes less energy when compared to PoS.

Summary of all simulation observations

1. *Time Performance:*
   Setup time is higher in Method 1 DPS while PS iterations time is higher in Method 2 DPS. PoW time is dependent on the Network Difficulty. Network difficulty is configurable making it possible to make PoW slower or faster than PoS. PoS time can be reduced by setting a strict timeout in the case of performance based PoS.

2. *Scalability:*
   THV decides Network Difficulty which in turn decides PoW time. Higher THV will reduce difficulty and decrease PoW run time and may even bring it below PoS run time. Decrease in PoW run time negatively affects BFT level of a distributed network. Increase in difficulty, makes networks more secure but more time end energy is consumed in finding hash value.

3. *Winner Distribution in PoW:*
   Chances of winning block mining rights in PoW is proportional to the computational strength of a node.

4. *Stake Recovery in PoS:*
   PoS gives custom control over the Stake Recovery Formula to decide consensus winner and PoS reward. DSM Performance or/and Initial Stake may be incentivized while awarding block forging rights and PoS rewards.

5. *Energy Consumption:*
   As per simulations done using PyRAPL, the Method 2 implementation of DPS is more energy efficient than the Method 1 implementation.

7 Discussion, Conclusion and Future Work

Chapter Objective: Two approaches to distributed DSM and two consensus protocols from blockchain were analysed in the previous chapter. In this chapter, merits and demerits of both implementation methods of distributed DSM and consensus protocols are discussed and future research points are suggested.

Chapter Contents

7.1 Discussion

7.1.1 Distributed DSM

In this section, we discuss Method 1 and Method 2 implementations of distributed DSM. While we use profile steering [1] as the DSM algorithm, either method may be generalized to apply to any DSM algorithm in a distributed manner.

Truthfulness and DSM Redundancy

There is a trade-off between DSM redundancy and truthfulness in distributed DSM networks. In an ideal scenario, with none of the nodes having any scope or desire to engage in malicious behaviour, it would make sense to have only one node acquire the starting data of all other nodes and find

local solutions for all the nodes. This node could also be chosen using a random function written in the Smart Contract. This would reduce the DSM redundancy in Method 1 distributed DSM implementations that require each node to independently find the local solutions of all nodes.

Having one randomly assigned node calculating the local solutions for that day could reduce the total energy usage of the network. Since one randomly chosen node would find all local solutions, the energy usage during this stage would decrease by a factor of $N - 1$, if there are N nodes in the network. However this would subject the transactions mined for that day or the block content to the integrity of the chosen node. In practical networks, the chosen node can engage in malicious behaviour and create block content or profile transactions that are in its favour for that day, but compromise on the aggregate solution and thereby the DSM objectives of the network as a whole. This would be the case of zero DSM redundancy but at the cost of truthfulness of the block content.

On the contrary, when all nodes actively participate in the DSM process by either calculating or validating local solutions, the DSM redundancy is at maximum. In Method 1 implementations, all nodes compute all local solutions and in Method 2, all nodes find the local solution with the highest improvement per iteration. These are independently executed and redundant operations. But doing so ensures that the final block content that shall be mined into the blockchain, is not subject to the tentative integrity of any malicious node, since all nodes exit the DSM stage with access to the block content to be mined. This increases the BFT level of the network (see Subsection 7.1.2).

Data Security

Method 1 implementations of distributed DSM require each node to share their local data with all other network nodes. This creates data privacy issues in Method 1 implementations. In Method 2 implementations, there is no sharing of local data and nodes only update the aggregate solution of the network. This makes Method 2 implementations more secure from a data confidentiality perspective than Method 1 implementations and more suitable for permissionless or open networks.

Crash Fault Tolerance (CFT)

CFT of a network refers to a distributed network's resilience to the sudden crash of a node(s). For example, if a network is able to go ahead with regular operations when a node or a group of nodes crash leading to a sudden loss of communication, then such a network would be regarded as being Crash Fault Tolerant.

> *A network's capability, described by the safeguards or methods it has in place, to handle an abrupt loss of communication with a node(s) is called the Crash Fault Tolerance (CFT) of the network.*

Method 1 distributed DSM implementations require each node to acquire the local data of all other nodes while Method 2 implementations rely on constant communication during the DSM process. Method 1 implementations execute the DSM algorithm in an independent manner while Method 2 implementations execute the DSM algorithm as a collaborative process. This makes Method 1 implementations more crash fault tolerant than Method 2 implementations. However, appropriate solutions may be investigated to make Method 2 implementations more crash fault tolerant. For example, nodes may register a default local solution (daily profile) when entering the network to be used when and if that node crashes. Default values for each stage of the DSM algorithm may be registered within the Smart Contract by each node while entering the blockchain network. These default values can be the safest values that each node prefers for that stage of the DSM algorithm in the case of sudden loss of communication.

7.1.2 Consensus Protocols

In this section, we discuss both consensus protocols, PoW and PoS, with respect to certain parameters of distributed networks.

Byzantine Fault Tolerance (BFT)

Byzantine Fault Tolerance of a network refers to the distributed network's capability to resist malicious attacks from a node that is part of the network. One way a malicious node may attack a the blockchain is by attempting to

manipulate the block content. For example, a node may attempt to alter or delete a transaction (profile) in a block. It is up to the consensus protocols used in distributed networks to disincentivize such malicious behaviour.

> *The capacity of a consensus protocol, and by extension, of the distributed network to disincentivize such malicious behaviour is regarded as the Byzantine Fault Tolerance (BFT) of the network.*

As seen in Subsection 7.1.1, the chances of this is reduced by going for higher DSM redundancy. Though each node exits the DSM stage with the same block content, only one node is chosen to mine or forge the block as per the rules of the consensus round.

In PoW, the node with the highest computational capacity has the highest chance of winning block mining rights. One way then for a malicious node to make sure of its victory, is by acquiring at least 51% of the network's computing power. This would monopolize the action of mining the blocks into the hands of that node (or coalition of nodes). Such an attack is called a 51% attack and is a demerit of PoW.

In PoS, the block forging rights are awarded on the basis of either the node's initial stake or/and the node's DSM performance. If the block forging rights are awarded on the basis of DSM performance, then the only way a node can get block forging rights is by achieving the highest DSM performance, which works in favour of the network. However if the block forging rights depend on the initial stake, then this can lead to the 'rich get richer' problem of PoS networks. Nodes with the highest initial stake, have the highest chance of winning the consensus rounds and may reinvest their PoS reward into the network as initial stake and keep increasing their chances of winning. PoS networks that depend only on initial stake can fall victim to this problem of 'stake run-off'.

Incentive Analysis

Consensus protocols from blockchain play an important role in incentivizing nodal behaviour, and the success of a certain consensus protocol depends on what expected (and unexpected) behaviours they incentivize network nodes to engage in. For distributed DSM networks, consensus protocols that incentivize better nodal DSM performance (better local solutions) are preferred.

That is, consensus protocols that incentivize positive DSM behaviour from nodes are preferred.

PoW

In PoW, nodes are incentivized to either purchase more computing power or form coalitions with other network nodes to gain access to more computing power. PoW fails to incentivize positive nodal behaviour since any reward given in PoW is connected to winning block mining rights which in turn is connected to having more computing power. Consensus round wins are secured with higher computing power regardless of nodal DSM performance.

PoS

PoS on the other hand, offers interesting alternatives to PoW's positive incentive problem, depending on how it's implemented. If PoS is implemented purely on the basis of initial stake (Cases 3 and 4 of Chapter 6, Section 6.5), then this incentivizes nodes to increase their stake in the network. If the initial stake is derived from the nodal battery capacity then this can incentivize nodes to invest in higher battery capacity. This can increase the network's total flexibility and lead to a better aggregate solution. However there is no direct financial incentive for nodes to record higher daily DSM performance values. It must be noted that Case 4 represents the conventional blockchain implementation of PoS. Initial stake based implementations risk triggering the stake-runoff problem in PoS networks as nodes have an incentive to increase their ownership stake in the network. Depending on how the value of the ownership stake is derived, this may or may not benefit the quality of the aggregate solution.

Cases 1 and 2 of Chapter 6, Section 6.5 offer a performance based implementation of PoS. These implementations offer direct incentives to nodes to put forward better daily DSM behaviour in terms of better local solutions. Case 1 directly incentivizes better nodal performance while Case 2 directly incentivizes both better nodal performance and higher ownership stake.

In profile steering for example, a network's ability to achieve an aggregate profile closer to the desired profile is dependent on the network flexibility and by extension, dependent on nodal flexibility. Nodal flexibility itself seems to depend on different storage parameters, such as battery capacity and power rating.

For example, in Scenario 2, Case 1 the margin of total stake recovered between Category 1 and Category 3 nodes is low (the top two performing categories). Category 1 nodes (Sonnen Eco) have a higher battery capacity but a lower power rating than Category 3 nodes (Tesla Powerwall 2). In this case, there is no direct financial incentive for nodal initial stake (battery capacity) even though a higher battery capacity does aid Category 1 nodes in performing well in the PS iterations.

Cases 3 and 4 represent the other extreme, wherein direct financial incentives are given only for initial stake and not daily nodal DSM performance.

Depending on the values of the constants A and B, and how the ownership stake is actually calculated, Case 2 can provide a balance between incentivizing both higher initial investment and better nodal DSM performance. Depending on the DSM algorithm used, achieving this balance may be crucial. In PS, both, a higher battery capacity and a higher power rating seem to play a role in network flexibility. Therefore Case 2, offers direct financial incentives for nodes to invest in both. The values of constants A and B play an important role in achieving this balance. If $A >> B$ then Case 2 becomes Case 3 and if $B >> A$ then Case 2 becomes Case 1.

In this thesis, two methods to implement a DSM algorithm over a distributed network were proposed. Implementation Method 1 requires nodes to share local data with each other and perform all DSM algorithm iterations locally. Each node attempts to find the local solutions for all the network nodes. Implementation Method 2 requires nodes to only find their own local solution and update the aggregate s

Smart Contract

The Smart Contract in a blockchain network offers an important security against some actions that nodes may engage in. For example, it is important for nodes to fulfill the local solutions or profiles recorded in the block. A smart contract can contain penalties for a node that fails to do so. The Network Manager holds and enforces the Smart Contract. Since the only job of the Network Manager is to enforce the Smart Contract, it is in its own self-interest to execute this task with integrity. Additionally, all nodes have a copy of the blockchain, which in turn, makes it nearly impossible for individual nodes to execute a local solution that is different from the one

promised or recorded in the block content and yet not be held accountable for it.

Legal Weight

It is important to note that the legal weight of smart contracts is still under investigation. In the Netherlands, Dutch law does not require any specific formalities for a contract to be established between two or more parties. The contract may be physical or even built in computer code. Therefore, in principle, smart contracts are recognized as 'contracts' under Dutch contract law. However, the complete recognition of smart contracts within Dutch contract law is still pending and dependent on how certain legal standards applied to all contracts can be applied to smart contracts too. We refer to the report 'Blockchain and the Law' by Schellekens et al. [68] for a more detailed discussion.

7.2 Conclusion

The main research question proposed in this thesis was as follows:

> *How can demand side management be implemented in distributed networks?*

In this thesis, two methods to implement a DSM algorithm over a distributed network were proposed. Implementation Method 1 requires nodes to share local data with each other and perform all DSM algorithm iterations locally. Each node attempts to find the local solutions for all the network nodes. Implementation Method 2 requires nodes to only find their own local solution and update the aggregate solution by sharing the delta value. Once nodes arrive at their final local solutions and the aggregate solution, they require a consensus mechanism to decide the node that gets to confirm the solutions into the ledger. For this, two consensus protocols from blockchain, Proof of Work and Proof of Stake were implemented. This results in four approaches to distributed DSM using blockchain.

Distributed DSM

Decentralized DSM relies on a rigid allocation of DSM roles. Given the distributed nature of the communication layer of the energy grid, the avenue for such role distribution is decreasing. Each entity in an MG is capable of

becoming a prosumer. Distributed DSM transforms DSM algorithms into an authority-less form to be implemented in networks wherein each node is given the exact same privileges as all other nodes. This transformation comes at the cost of increased run time in the distributed approaches. In our simulations, the fastest distributed DSM model is found to be 5.77 times slower than PS. Distributed DSM with blockchain also has higher requirements of data throughput given the increased overhead of communicating consensus-specific information in each block. Higher software complexity is required due to the need for maintaining and verifying the blockchain, and competing in the consensus round. Blockchain based DSM, by its nature of block immutability, distributed ledger system and absence of central authority, is more secure than just decentralized DSM.

Distributed DSM implementations

In the Method 1 implementation of distributed DSM, each node finds the local solution for all network nodes while, in the Method 2 implementation of distributed DSM, each node only finds its own local solution. This makes Method 2 more scalable for larger network sizes and reduces the need for faster hardware in larger networks as compared to Method 1. Additionally it also preserves local data privacy and is more energy efficient. With improvements in its actual implementation with better software techniques, the Method 2 run time can be decreased.

PoW and PoS

Between the two consensus protocols, PoS is more preferable than PoW. Security in a PoW based network is achieved by having a challenging THV. But a challenging THV results in more energy usage during the consensus round. Nodes in an MG could spend a considerable amount of energy in just trying to find a solution to the mathematical problem. On the other hand, PoS is not an energy intensive process. Security in a PoS network is achieved by the nature of the consensus algorithm itself. That is, it is counter-intuitive for nodes to compromise a network in which they have invested stake. PoS requires less complex hardware than PoW and is more suited for application in (remote) MGs.

MG Total Stake Value

PoS based consensus is an interesting consensus algorithm for (remote) MGs. The total stake invested in the network and the recovered stake give a numerical way to measure the 'condition' of an MG. In the special case of

remote MGs, this is a critical advantage to have. Remote MGs are usually deployed in regions that are difficult to access. Such regions may also be devoid of advanced communication infrastructure making it necessary to measure the condition of a remote MG from a central location with only minimum data transfer.

Use Case: Imagine a remote MG employing PoS based consensus wherein all nodes are given a daily PoS reward proportional to their DSM performance (Stake Recovery Cases 1 and 2). All nodes are required to invest their daily PoS reward back into the same MG as additions to their previous ownership stake. The total ownership stake in the MG at any instant may be referred to the *Stake Value* of that MG. This means that the Stake Value on $Day\ n+1$ in a network with M nodes is calculated by the formula:

$$StakeValue_{Day\ n+1} = StakeValue_{Day\ n} + \sum_{m=0}^{m=M} PoSReward_m$$

Now the rate at which the stake value of an MG increases in a certain time interval can give interesting insights into the network's overall performance within that time interval. The rate of increase is the slope of the *MG Stake Value VS Day* graph.

A slope of 1 within a certain time interval implies that the daily overall network DSM performance (sum of daily nodal DSM performances) is constant. An exponential growth curve implies that the daily network performance increased in that time interval and an exponential decay curve implies that the daily network performance decreased within that time interval. A steep rate of decay may even imply a situation for caution. In this use case, the MG's stake value becomes analogous to the stock value of a public company.

In addition to the above, stake recovery in PoS gives a much more interesting avenue to directly motivate better nodal DSM performance and investment in better energy storage options. PoS gives direct incentives to increase the MG's resiliency, a much needed advantage for (remote) micro-grids for energy resilient communities.

7.3 Future Work

Scalability and Time Performance: All four approaches are simulated with a maximum of 25 nodes. Simulations with more nodes may provide interesting results in terms of the comparative scalability of all four approaches. Additionally, improvements can be made to the DPS implementations to reduce run time. For example, the Method 2 implementation of DPS can be implemented with file locks to reduce I/O time and idle waiting time resulting from multiple processes writing to the same file.

Smart Contracts in DSM: It is important to investigate the role that the Smart Contracts can play in the DSM stage too, if any. In the four approaches proposed in this thesis, the Smart Contract only comes into play in the consensus rounds. Smart Contracts offer a way to implement a piece of code without the requirement of any external intervention. Therefore, certain DSM operations that would usually be executed by the coordinator node, could be written into the Smart Contract.

For example, in the Method 1 implementation of distributed DSM, nodes could submit all their starting local data to the Smart Contract, which would then execute the coordinator node operations in PS. In the Method 2 implementation of distributed DSM, nodes can submit their improvement values to the Smart Contract. The Smart Contract can choose the node with the highest PS improvement and broadcast the delta profile to all the network nodes.

In the above approaches, the smart contract executes the role of the coordinator. However this places the decision making authority during the transaction phase (DSM stage in this case) into the hands of the Smart Contract. This is contradictory to the role that Smart Contracts are usually supposed to play in a blockchain network. In a blockchain network, Smart Contracts do not involve themselves in the transaction phase. It is the network nodes (specifically light nodes) that validate transactions. The reasoning behind this design decision is to prevent any localization of authority into the hands of the smart contract, which would in effect, make it a centralized network. Smart contracts are only supposed to enforce network guidelines, not actively participate in the network's transactions.

It is possible to allocate the coordinator role to the Smart Contract but to retain the distributed ledger concept from blockchain. This may be investigated as a third method of implementing DSM.

Security: Commercial blockchain networks implement Public-Key Cryptography (PKC) to verify nodal identity on a distributed network. Specifically, PKC is used by the receiver node to verify the source node when it receives a submission. The implementation of PKC may result in interesting results with respect to time performance in Method 1 and Method 2 implementations of distributed DSM since both methods differ in the instances of inter-node data sharing.

Economic Analysis: An economic comparison of the four approaches, especially the two consensus protocols is important. Setting practical values for financial rewards offered in the implementation of either protocol is important. This can give important insights especially in the case of Proof of Stake wherein, knowledge of the practical values of the reward constants A, B and C and the nature and value of the initial stake is required.

In this thesis, scenarios wherein only the winning node receives a PoS reward have been studied. A stake recovery analysis of scenarios wherein all nodes gain some PoS reward in every round may also be investigated. Another interesting aspect of PoS is to implement the ownership stake simultaneously with local energy trading. For example, node's can be incentivized to trade their surplus energy locally by gaining a proportional increase in their network stake.

DSM algorithms: The two proposed methods of distributed DSM are tested with only the profile steering algorithm. While the implementation methods may be applied to any DSM algorithm, this aspect needs to be simulated with different DSM algorithms. Using different DSM algorithms may lead to different time performance results and may also give rise to different incentives to take advantage of during the consensus rounds.

Specifically for PS, it would be interesting to study the exact dependency between energy storage parameters and nodal flexibility. Such a study would give a better idea about which parameters affect nodal flexibility and by how much. This would aid in forming more informed decisions about the values of the PoS reward constants A, B and C and in designing better incentive models.

Commercial Implementation: Finally, the proposed approaches to distributed DSM with blockchain may be tested on a public blockchain platform and to take it a step further, the model may be deployed in an actual neighbourhood. Ethereum offers frameworks such as Truffle and Brownie to deploy a local development environment based on PoW consensus [69]. Ethereum recently started a PoS based blockchain implementation called The Beacon Chain and offers frameworks to deploy a local client and host DApps (Smart Contracts) on the same [70].

Bibliography

[1]M. E. T. Gerards, H. A. Toersche, G. Hoogsteen, *et al.*, "Demand side management using profile steering," *IEEE Eindhoven PowerTech*, pp. 1–6, 2015. DOI: 10.1109/PTC.2015.7232328 (cit. on pp. v, 20, 24, 29, 32, 33, 37, 105).

[2]"The history of electrification," *Edison Tech Center*, (cit. on p. 7).

[3]"Perl street station," *Engineering and Technology Wiki*, (cit. on p. 7).

[4]R. Alfred, *Aug. 14, 1888: I sing the meter electric* (cit. on p. 8).

[5]M. Whelan, S. Rockwell, and T. Blalock, *Great barrington 1886* (cit. on p. 8).

[6]E. Nix, "How edison, tesla and westinghouse battled to electrify america," Oct. 2019, [Online; accessed 25. Sept. 2020] (cit. on p. 8).

[7]"Microgrids are on the rise," *Energy Planet*, Apr. 2018 (cit. on p. 8).

[8]B. L. Capehart, "Distributed energy resources (der)," Oct. 2016 (cit. on p. 9).

[9]*P1547.2 - application guide for ieee std 1547(tm), ieee standard for interconnecting distributed resources with electric power systems*, Sep. 2017 (cit. on p. 9).

[10]A. Meyer, "Why a distributed energy grid is a better energy grid," May 2016 (cit. on p. 9).

[11]"Distributed generation of electricity and its environmental impacts," *EPA*, (cit. on p. 9).

[12]D. Proctor, "Distributed energy resources bring benefits, challenges and new opportunities," Feb. 2018 (cit. on p. 9).

[13]"Trends in the netherlands in 2018: Economy," *CBS*, 2018 (cit. on p. 10).

[14]M. Nazari-Heris, S. Madadi, and B. Mohammadi-Ivatloo, *Classical and Recent Aspects of Power System Optimization*. Academic Press, Jun. 2018, pp. 407–420. DOI: 10.1016/B978-0-12-812441-3.00015-X (cit. on p. 9).

[15]NL Agency, "Smart grids and energy storage," p. 2, Nov. 2012 (cit. on p. 10).

[16]"Smartgrids: Strategic deployment document for europe's electricity networks of the future," *European Technology Platform*, Apr. 2010 (cit. on p. 10).

[17]US Department of Energy, "Smart grid system report: 2018 report to congress," Nov. 2018 (cit. on p. 10).

[18]Heinrich Böll Stiftung Foundation, "Energy atlas: Facts and figures about renewables in europe," *European Energy Atlas 2018*, Apr. 2018 (cit. on p. 12).

[19]N. Routley, “Mapped: The 1.2 billion people without access to electricity,” Nov. 2019 (cit. on p. 13).

[20]F. Zhang, “In the dark: How much do power sector distortions cost south asia?,” 2019 (cit. on p. 13).

[21]*Energy access outlook 2017*, [Online; accessed on 27. Sept. 2020], Oct. 2017 (cit. on p. 13).

[22]*Solving the energy access problem with renewable mini-grids*, Oct. 2016 (cit. on p. 14).

[23]V. Ramachandran, J. Obado-Joel, R. Fatai, J. Masood, and B. Omakwu, “The new economy of africa: Opportunities for nigeria’s emerging technology sector,” Nov. 2019 (cit. on p. 14).

[24]B. Oluwafemi, “Mtn nigeria warns that diesel shortage could soon jeopardise network services,” May 2015 (cit. on p. 15).

[25]M. Shuaia, W. Chengzhib, Y. Shiwena, *et al.*, “Review on economic loss assessment of power outages,” *Procedia Computer Science*, vol. 130, pp. 1158–1163, Apr. 2018. DOI: 10.1016/j.procs.2018.04.151 (cit. on p. 15).

[26]L. Weifang, T. Yong, S. Huadong, *et al.*, “Blackout in brazil power grid on february 4,2011 and inspirations for stable operation of power grid,” *Automation of Electric Power Systems*, vol. 35, no. 9, pp. 1–5, 2011 (cit. on p. 15).

[27]A. Ricci, S. Faberi, N. Brizard, *et al.*, “Smart grids/energy grids,” Sep. 2012 (cit. on p. 15).

[28]M. Metcalfe, “Grid efficiency: An opportunity to reduce emissions,” Aug. 2017 (cit. on p. 16).

[29]D. Seavers, “The democratization of energy,” Jan. 2017 (cit. on p. 16).

[30]Y. Luo, S. Itaya, S. Nakamura, and P. Davis, “Autonomous cooperative energy trading between prosumers for microgrid systems,” *39th Annual IEEE Conference on Local Computer Networks Workshops*, pp. 693–696, Sep. 2014. DOI: 10.1109/LCNW.2014.6927722 (cit. on p. 16).

[31]O. Longe, K. Ouahada, S. Rimer, H. Ferreira, and A. Vinck, “Distributed optimisation algorithm for demand side management in a grid-connected smart microgrid,” *Sustainability*, vol. 9, no. 2, p. 1088, Jun. 2017. DOI: 10.3390/su9071088 (cit. on pp. 22, 28).

[32]S. Jha and D. Kumar, “Demand side management for stand-alone microgrid using coordinated control of battery energy storage system and hybrid renewable energy sources,” *Electric Power Components and Systems*, vol. 47, pp. 1261–1273, Jun. 2019. DOI: 10.1080/15325008.2019.1661544 (cit. on pp. 23, 28).

[33]M. Marzband, E. Yousefnejad, A. Sumper, and J. Domínguez-García, “Real time experimental implementation of optimum energy management system in standalone microgrid by using multi-layer ant colony optimization,” *International Journal of Electrical Power Energy Systems*, vol. 75, pp. 265–274, 2016. DOI: 10.1016/j.ijepes.2015.09.010 (cit. on pp. 23, 29).

[34]S. Noor, W. Yang, M. Guo, K. van Dam, and X. Wang, "Energy demand side management within micro-grid networks enhanced by blockchain," *Applied Energy*, vol. 228, pp. 1385–1398, 2018. DOI: 10.1016/j.apenergy.2018.07.012 (cit. on pp. 23, 29).

[35]E. Münsing, J. Mather, and S. Moura, "Blockchains for decentralized optimization of energy resources in microgrid networks," *IEEE Conference on Control Technology and Applications (CCTA)*, pp. 2164–2171, 2017. DOI: 10.1109/CCTA.2017.8062773 (cit. on pp. 25, 30).

[36]T. Alskaif and G. van Leeuwen, "Decentralized optimal power flow in distribution networks using blockchain," *International Conference on Smart Energy Systems and Technologies (SEST)*, pp. 1–6, 2019. DOI: 10.1109/SEST.2019.8849153 (cit. on pp. 25, 30).

[37]*Power ledger*, [Online; accessed 28. Sept. 2020] (cit. on pp. 26, 30).

[38]*Lo3 energy*, [Online; accessed 28. Sept. 2020] (cit. on pp. 26, 30).

[39]M. Rizwan, *Nrgcoin – smart contract for green energy*, May 2020 (cit. on pp. 26, 30).

[40]*World's first full smart energy community with blockchain technology*, Dec. 2017 (cit. on p. 27).

[41]*Codeveloping the first large-scale peer-to-peer energy trading market in the netherlands* (cit. on p. 27).

[42]C. Burgahn, *Launch of the open charging network*, Mar. 2020 (cit. on pp. 27, 30).

[43]*Dynamic pricing in electricity supply*, Feb. 2017 (cit. on p. 33).

[44]T. van der Klauw, "Decentralized energy management with profile steering: Resource allocation problems in energy management," CTIT Ph.D. thesis series no. 17-424, Ph.D. dissertation, University of Twente, Netherlands, May 2017. DOI: 10.3990/1.9789036543019 (cit. on p. 35).

[45]*Pecan street: Residential data*, [Online; accessed 9. June. 2020], Oct. 2019 (cit. on p. 38).

[46]S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Cryptography Mailing list at https://metzdowd.com*, Mar. 2008 (cit. on p. 42).

[47]*Bitcoin*, [Online; accessed 13. Oct. 2020] (cit. on p. 42).

[48]*Classification and importance of nodes in a blockchain network*, Aug. 2020 (cit. on p. 43).

[49]*Global bitcoin nodes distribution*, [Online; accessed 23. Dec. 2020] (cit. on p. 43).

[50]L. Frost, "Bitcoin blockchain grows to 300 gigabytes in size," Sep. 2020, [Online; accessed 23. Dec. 2020] (cit. on p. 43).

[51]*What is a merkle tree and how does it affect blockchain technology?* Nov. 2019 (cit. on p. 48).

[52]B. Whittle, “What is a nonce? a no-nonsense dive into proof of work,” Dec. 2018 (cit. on p. 53).

[53]*Bitcoin energy consumption index*, [Online; accessed 14. Oct. 2020] (cit. on pp. 54, 55).

[54]“Proof of stake explained,” Feb. 2020 (cit. on p. 55).

[55]A. Shevchenko, “Proof-of-stake vs. proof-of-work: Which one is ‘fairer’?,” Apr. 2020 (cit. on p. 56).

[56]*A list of blockchain protocols - explained and compared*, May 2019 (cit. on p. 57).

[57]*Proof of weight (poweight)*, 2018 (cit. on p. 57).

[58]W. Kenton, “Proof of burn (cryptocurrency),” Feb. 2020 (cit. on p. 57).

[59]*Proof of participation*, Nov. 2019 (cit. on p. 58).

[60]J. Frankenfield, “Proof of elapsed time (poet) (cryptocurrency),” Aug. 2020 (cit. on p. 58).

[61]S. Seth, “Proof of assignment (poa),” Jul. 2020 (cit. on p. 58).

[62]*What is a directed acyclic graph (dag) in cryptocurrency?* Aug. 2020 (cit. on p. 58).

[63]S. Daley, “15 examples of how blockchain is reviving healthcare.,” Jul. 2019 (cit. on p. 59).

[64]K. Leslove, “Blockchain enters the uk video games industry with a bang,” Jan. 2019 (cit. on p. 59).

[65]F. Þ. Hjálmarsson, G. K. Hreiðarsson, M. Hamdaqa, and G. Hjálmtýsson, “Blockchain-based e-voting system,” *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pp. 983–986, Sep. 2018. DOI: 10.1109/CLOUD.2018.00151 (cit. on p. 59).

[66]*Network difficulty*, [Online; accessed 16. Dec. 2020] (cit. on p. 86).

[67]*Pyrapl*, [Online; accessed 16. Dec. 2020] (cit. on p. 102).

[68]M. Schellekens, E. Tjong, T. Tai, *et al.*, “Blockchain en het recht,” Jun. 2019 (cit. on p. 111).

[69]Ethereum, *Set up your local development environment*, [Online; accessed 31. Dec. 2020], 2020 (cit. on p. 116).

[70]Ethereum Foundation, *The beacon chain*, [Online; accessed 31. Dec. 2020], 2020 (cit. on p. 116).

Colophon

This thesis was typeset with LaTeX 2$_\varepsilon$. It uses the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

Download the *Clean Thesis* style at http://cleanthesis.der-ric.de/.