BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF INTELLIGENT SYSTEMS
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

MODELING AND SIMULATION OF INCENTIVE MECHANISMS IN ETHEREUM

MODELOVÁNÍ A SIMULACE VLASTNOSTÍ POPLATKOVÝCH MECHANISMŮ V SÍTI ETHEREUM

BACHELOR’S THESIS
BAKALÁŘSKÁ PRÁCE

AUTOR TEREZA BURIANOVÁ
AUTOR PRÁCE

SUPERVISOR Ing. MARTIN PEREŠÍNI
VEDOUCÍ PRÁCE

BRNO 2022

Department of Intelligent Systems (DITS) Academic year 2021/2022

Bachelor's Thesis Specification

24237

Student:	**Burianová Tereza**
Programme:	Information Technology
Title:	**Modeling and Simulation of Incentive Mechanisms in Ethereum**
Category:	Modelling and Simulation

Assignment:

1. Study the background related to blockchains and cryptocurrencies, specifically their unique properties, application layer, and consensus protocols.
2. Study and analyze a current incentive mechanism of Ethereum as well as its improvement proposal EIP 1559. Acquaint yourself with literature investigating EIP 1559.
3. Propose simulation experiments investigating the behavior of both incentive schemes (original and extended ones).
4. Implement the simulation model and run different simulation scenarios focusing on two incentive schemes exploring corner cases.
5. Discuss your results and compare them with the existing literature.

Recommended literature:

- I. Homoliak, S. Venugopalan, D. Reijsbergen, Q. Hum, R. Schumi and P. Szalachowski, "The Security Reference Architecture for Blockchains: Toward a Standardized Model for Studying Vulnerabilities, Threats, and Defenses," in IEEE Communications Surveys & Tutorials, vol. 23, no. 1, pp. 341-390, Firstquarter 2021, doi: 10.1109/COMST.2020.3033665.
- S. Leonardos, B. Monnot, D. Reijsbergen, S. Skoulakis and G. Piliouras, "Dynamical Analysis of the EIP-1559 Ethereum Fee Market", arXiv: https://arxiv.org/abs/2102.10567
- Tim Roughgarden, "Transaction Fee Mechanism Design for the Ethereum Blockchain: An Economic Analysis of EIP-1559", arXiv: https://arxiv.org/abs/2012.00854
- EIP-1559 Proposal: https://eips.ethereum.org/EIPS/eip-1559
- EIP-1559 Resources: https://hackmd.io/@timbeiko/1559-resources
- ETH with EIP-1559 burned fee meter: https://etherchain.org/burn

Requirements for the first semester:

- Items 1 to 3.

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor:	**Perešíni Martin, Ing.**
Consultant:	Homoliak Ivan, Ing., Ph.D., UITS FIT VUT
Head of Department:	Hanáček Petr, doc. Dr. Ing.
Beginning of work:	November 1, 2021
Submission deadline:	May 11, 2022
Approval date:	November 4, 2021

Abstract

The topic of this thesis is the Ethereum incentive mechanism, in particular the changes introduced in EIP-1559. The aim of the thesis is to investigate the behaviour and propose any potential improvements in case of discovered flaws. The previously used first price auction mechanism required users to choose the incentive arbitrarily, which led to overpaying and high fee volatility. These problems occurred mainly due to higher network utilization after the popularization of projects such as decentralized finance, NFT collections, and the metaverse. The new incentive mechanism introduced the variable block size, which can adapt to the current network usage. Base fee, a value that indicates the minimum fee needed to include the transaction in the block, is then calculated based on the utilization of the previous block, making the fees more predictable. Several simulation experiments were proposed to investigate the typical behaviour and possible weaknesses of the mechanism. Finally, a possible improvement was found, and future research was proposed. The goals of the thesis were achieved, and the results were presented in the thesis.

Abstrakt

Tématem této bakalářské práce je poplatkový mechanismus v síti Ethereum, zejména změny představeny v EIP-1559. Cílem práce je zkoumat chování mechanismu a navrhnout případná možná vylepšení v případě nalezených nedostatků. Dříve používaný aukční systém vyžadoval libovolné nastavení výše poplatku uživatelem, což vedlo k přeplatkům a vysoké volatilitě výše poplatků. Tyto problémy nastaly převážně kvůli vyššímu vytížení sítě po popularizaci projektů jako například decentralizované finance, NFT kolekce a metaverse. Nový poplatkový mechanismus zavedl proměnlivou velikost bloku, která se dokáže přizpůsobit aktuálnímu vytížení sítě. Base fee, hodnota značící minimální výši poplatku potřebnou pro zahrnutí do bloku, je pak vypočítána na základě zaplněnosti předchozího bloku, což dělá poplatky více předvídatelnými. Bylo navrženo několik simulačních experimentů, které zkoumají typické chování a možné slabiny mechanismu. Nakonec bylo nalezeno možné vylepšení a byl navržen další výzkum. Cíle práce byly splněny a výsledky byly prezentovány.

Keywords

blockchain, cryptography, Ethereum, incentive mechanism, EIP-1559, London Hard Fork, modeling and simulation, base fee, block utilization, gas price volatility, tokens, ETH burn, Ethereum network congestion, DeFi, NFT

Klíčová slova

blockchain, kryptografie, Ethereum, poplatkový mechanismus, EIP-1559, London Hard Fork, modelování a simulace, base fee, zaplněnost bloku, volatilita ceny za gas, tokeny, pálení ETH, zahlcení Ethereum sítě, DeFi, NFT

Reference

BURIANOVÁ, Tereza. *Modeling and Simulation of Incentive Mechanisms in Ethereum.* Brno, 2022. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Martin Perešíni

Rozšířený abstrakt

Tématem této bakalářské práce je poplatkový mechanismus v síti Ethereum, zejména změny představeny EIP-1559 v srpnu 2021. Ethereum je protokol umožňující fungování decentralizovaných produktů na blockchainu. Nativní měnou sítě Ethereum je Ether. Zájem o blockchain technologie každým dnem stoupá, na síť Ethereum uživatele lákají projekty zabývající se například decentralizovanými financemi, NFT kolekcemi či metaverse hrami. Vzhledem k rostoucímu počtu uživatelů začal být předchozí poplatkový mechanismus neefektivní a uživatelsky nepřívětivý. Změny představeny v EIP-1559 mají za úkol zefektivnit fungování celé sítě a udělat poplatkový systém více předvídatelným a uživatelsky přívětivým. Cílem práce je provést experimenty, které umožní zkoumat chování mechanismu a navrhnout případná možná vylepšení v případě nalezených nedostatků.

Pro odeslání transakce na síti Ethereum je třeba zaplatit poplatek, který odměňuje těžaře za zpracování transakce. Dříve používaný aukční systém vyžadoval libovolné nastavení výše poplatku uživatelem, což vedlo k přeplatkům a vysoké volatilitě výše poplatků. Nový poplatkový mechanismus zavedl proměnlivou velikost bloku, která se dokáže přizpůsobit aktuálnímu vytížení sítě. Zatímco cílová zaplněnost bloku zůstává stejná, maximální limit se může navýšit až na dvakrát větší hodnoty. Base fee, hodnota značící minimální výši poplatku potřebnou pro zahrnutí do bloku, je pak vypočítána na základě zaplněnosti předchozího bloku, což dělá poplatky více předvídatelnými. Lze také určit libovolné spropitné pro těžaře, který pak transakci zpracuje rychleji na základě výše zvoleného spropitného. Část poplatku určena base fee je spálena a spropitné je proplaceno těžaři společně s odměnou za vytěžení bloku.

Bylo navrženo několik simulačních experimentů, které zkoumají typické chování a možné slabiny mechanismu. Pro vykonání těchto simulačních experimentů bylo použito simulační prostředí abm1559, které se věnuje přímo poplatkovému mechanismu EIP-1559. V experimentech je nejprve zkoumáno chování mechanismu a následně jsou změněny vlastnosti mechanismu EIP-1559, využité k výpočtu klíčových hodnot, jako například maximální velikost bloku nebo maximální změna hodnoty base fee mezi dvěma bloky. První sekce se zabývá běžným každodenním využitím sítě bez větších výkyvů v počtu přicházejících transakcí. Všechny experimenty jsou vykonány na základě reálných historických dat, získaných z mempool statistik a přímo z blockchainu. Podobný postup byl zvolen i v druhé sekci, která pracuje s náhlým nárůstem přicházejících transakcí. Experimenty byly inspirovány situací v dubnu 2022, kdy vysoký zájem o vydání NFT metaverse hry Otherside, projektu od tvůrců známé NFT kolekce Bored Ape Yacht Club, způsobil zvýšení poplatků a přehlcení sítě, které přetrvalo po dobu několika hodin. Tato událost byla nasimulována s aktuálně platnými hodnotami parametrů. Následně bylo provedeno několik experimentů s odlišnými parametry, které měly ukázat, zda by síť takový zájem zvládla lépe při nastavení odlišných parametrů.

Porovnání běhů s různými klíčovými vlastnostmi bylo vyhodnoceno a na základě výsledků bylo nalezeno možné vylepšení spočívající v navýšení limitu velikosti bloku. Tato změna musí být dále pečlivě zkoumána před samotnou implementací, protože by mohla zvýšit požadavky na potřebný hardware a tím negativně ovlivnit počet uzlů v síti a ohrozit tak decentralizaci a bezpečnost. Z toho důvodu bylo navrženo provedení dalšího výzkumu se zaměřením na chování ostatních vrstev při změnách poplatkového mechanismu. Cíle práce byly splněny a výsledky byly prezentovány.

Modeling and Simulation of Incentive Mechanisms in Ethereum

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Ing. Martin Perešíni. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

. . . .
Tereza Burianová
May 17, 2022

Acknowledgements

I would like to thank my supervisor Ing. Martin Perešíni, who regularly provided me with advice, feedback, and materials that were crucial in writing this bachelor thesis. I also wish to thank my family and friends for supporting me not only during the writing of this thesis, but also during the whole three years of my studies.

Contents

Chapter 1

Introduction

In the last few years, blockchain technology has gained a lot of popularity. It provided for the establishment of cryptocurrencies, digital money independent of a central authority. Users from the whole world are attracted by properties like transparency or pseudo-anonymity, which offer more privacy, equality and safety regardless of their location, living conditions or government control.

The first successful project is called Bitcoin. It offers the option of opening an account and sending transactions, similar to a bank account, but it does not really offer an infrastructure that would allow for the development of automatized contracts and financial services. That was the main reason for creating Ethereum - a protocol allowing smart contract functionality.

One of the biggest problems is the high volatility of transaction fees. The fee directly affects the speed of transaction processing, motivating users to pay significantly higher fees if speed is important to them. The decentralized finances, NFTs and other popular projects have increased the interest in the Ethereum network, causing occasional peaks in the requests, volatile fees and network congestion.

One of the possible solutions is the new incentive mechanism proposed in EIP-1559. The block does not have a fixed size, but rather a maximal limit and a target size. The fee is now divided into two parts, the base fee and the tip. The base fee is calculated for each of the blocks based on the previous block utilization and serves as a tool to regulate the current demand. The base fee gets burned after the processing. The arbitrary tip is for users who wish to process their transactions faster because it serves as an incentive to the miner. The aim of this thesis is to investigate the behaviour of the new incentive mechanism using simulations, detect possible weaknesses that need to be taken into account in further development and propose possible improvements.

The thesis is divided into several chapters. Chapter 2 explains the core principles of blockchain, the important properties and the basic structure, valid for all of the well-known blockchains, including Ethereum. It describes all of the layers of the technology. Chapter 3 is closely focused on Ethereum and its implementation details. The incentive mechanisms, which are the topic of this thesis, are explained in this chapter. In Chapter 4, the general process of modelling and simulations is described. Part of the chapter contains a description of the tools used to gain knowledge about the incentive mechanism using simulations. Chapter 5 consists of two parts, one investigating the behaviour of the network under the typical everyday demand, and the other one discovering possible weaknesses in case of very high demand. Each of the parts consists of several experiments and an evaluation of the results.

Chapter 2

Blockchain

Blockchain, as the name implies, is a chained list of blocks containing various information. These blocks cannot be removed and are not managed by a third party. Together they create a digital ledger of transactions, which can be seen by the participants or even the public [31].

The initial conception was introduced in the 1990s. At that time, it was described as a consensus model (explained in Section 2.3.5) allowing participants in the computer network to come to an agreement, taking into account the possible unreliability of some of the participants. The concept was used to digitally sign documents.

At the same time, a group called cypherpunks launched a mailing list discussing cryptography. The members of this group predicted the future importance of the internet. They determined cryptography as the only tool that could stop government surveillance and censorship. A sovereign economy, independent of the central banks, was a part of their philosophy. That inspired David Chaum, a member of the cypherpunks, to create the first digital currency.

In 2008, blockchain was first utilized for creating **decentralized electronic cash**. This use case was described in the paper Bitcoin: A Peer to Peer Electronic Cash System [22]. The Bitcoin network emerged a year later, in 2009. Most of today's cryptocurrencies are based on the same concepts.

Although cryptocurrencies are quickly gaining popularity, there are other ways to utilize this technology. One of them is called **smart contract** (Section 3.3), which is software deployed on the blockchain. Even extensive decentralized applications, allowing for electronic voting, auctions, notaries, trading or loans, can be developed using smart contracts. These are secure thanks to cryptography, features inherited from blockchain and peer-to-peer networking [17].

There are two categories of blockchain, based on the level of permissions. In the **permissionless model**, anyone can publish and see blocks. No authority is needed to permit activity. The disadvantage is the possibility of malicious participants who may want to influence the system for their own benefit. This can be partially prevented by using various consensus mechanisms (Section 2.3.5). In the **permissioned model**, all transactions need to be authorized. The participants are considered to be trusted because their identity is known by the authority. Therefore, the transactions are not anonymous.

2.1 Cryptography

Cryptography is a key principle used in blockchain [32, 34, 21]. It secures data secrecy and protects it against unauthorized changes. The most basic techniques are called symmetric and asymmetric encryption. Both of them utilize encryption algorithms, which use various mathematical operations to transform input data (plaintext) to another form called ciphertext. The algorithm transforms every bit into a parallel bit in the ciphertext. Decryption algorithms ensure transformation reversibility, meaning it is possible to get the initial data from the changed form. The way blockchain utilizes cryptography principles is further described in Section 2.3.

In the **symmetric encryption**, data integrity is provided using an identical secret key as a parameter in encryption and decryption algorithms. It is generally faster than asymmetric encryption due to using a shorter key, but the need for key distribution makes it prone to security breaches. Some examples of symmetric encryption are DES or AES.

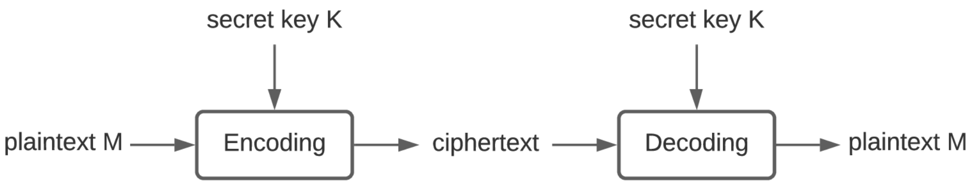

Figure 2.1: Diagram of the symmetric encryption.

The **asymmetric encryption** requires two different keys: the public key and the private key. The public key, used to encrypt messages, can be accessible to the public. Decryption is done using the private key, which must not be shared. Some of the asymmetric encryption algorithms include RSA, DSA and the Diffie-Hellman method.

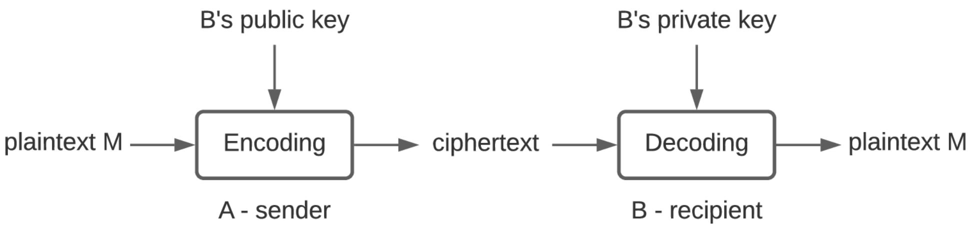

Figure 2.2: Diagram of the asymmetric encryption.

These two methods are frequently used together, symmetric for encryption and asymmetric for digital signatures and key exchange. The secret key used for faster encryption can be transferred securely using the public and private keys.

Another form of cryptography algorithms are the **cryptographic hash functions**, which map a message of variable length to a fixed-length value. The hashed value cannot be transformed back to the initial message and is never the same for two different inputs, therefore it allows for detecting unauthorized changes in data. Examples of cryptographic hash functions are MD4, MD5, SHA-1 or SHA-2. This method can also be used in combination with symmetric cryptography, which adds the shared secret key to the function input. The output is then called HMAC (Hash-Based Message Authentication Code). Examples of these functions are HMAC-MD4, HMAC-MD5, HMAC-SHA-1 or HMAC-SHA-2.

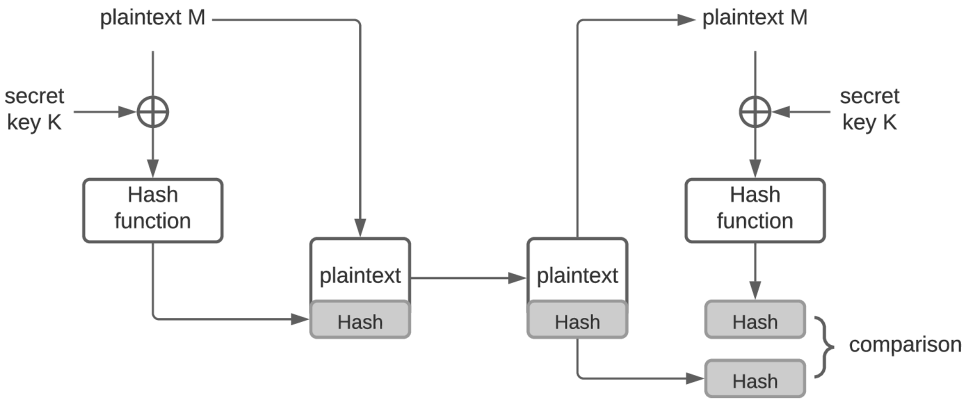

Figure 2.3: Diagram of the HMAC cryptographic hash function.

2.2 Properties

The Blockchain Trilemma

The Blockchain Trilemma is a concept, first described by the Ethereum co-founder Vitalik Buterin. It describes the relations among important blockchain properties, scalability, safety and security, which cannot be achieved all at once and one of them must be sacrificed in favour of the other two, as seen in Figure 2.4. This phenomenon occurs on the **monolithic blockchains**, including Ethereum, which try to reach all of these features present in one place - Layer 1. To speed up the transaction processing, the number of nodes executing the computations needs to be lower, but that would lead to decreased decentralization. The few nodes would need to be trusted with all the transactions. To make the network secure and decentralized, more participants in the network are needed, but in that case, transactions would take much longer. The solution to this problem may be **modular blockchains**, which divide the main monolithic blockchain into several layers [25, 16].

Decentralization

The key feature of blockchains is called decentralization, which means they are not managed by an individual, but rather by a collection of nodes, typically computers of people who partake in the activity. This makes the technology more accessible, portable and under the control of the users [18, 23].

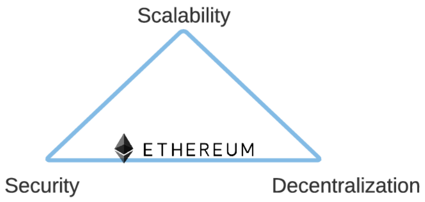

Figure 2.4: The blockchain trilemma and the properties, which are unachievable all at once.

Scalability

The term scalability stands for the number of transactions that can be processed per unit of time. This is the most problematic property for blockchains like Ethereum or Bitcoin considering the big number of nodes which need to execute various tasks to validate the transactions. There are several different solutions addressing the issue, for instance, sharding, the Proof-of-stake consensus (Section 2.3.5) or the Lightning Network [18, 19].

Security

Properties such as decentralization and the use of encryption ensure increased security of the network. Information stored in the blockchain cannot be changed and therefore is secured from fraud (as described in Section 2.2). Another layer of security is added through cryptography - an algorithm that hashes all information on the blockchain [18]. Cryptography is further explained in Section 2.1.

Immutability

Once the majority of the nodes confirms the validity of a transaction, which is then added to the chain, it cannot be removed or tampered with. Every node has a copy of the ledger, therefore the probability of fraud is not as high as when using a third party. That is why blockchain can be an important technology in the fight against corruption. However, the immutability of transactions is not immediate [18, 17].

One of the disadvantages of immutability can be the size of the data. With the rising number of transactions executed every day, this can quickly become a problem. For example, the Ethereum blockchain, which is still a relatively new technology, has already crossed the size limit of 1 TB. Another problem may occur during the development of various smart contracts. These need to be thoroughly checked for vulnerabilities, considering that after the deployment of the smart contract, all the information is irreversibly stored in the blockchain. Sufficient testing is therefore crucial to avoid future security breaches. One of the most discussed hacks is the smart contract of „The DAO“ [30, 33]. In this project, participants could exchange Ether for DAO tokens, and thanks to its success, the organisation was worth over $250 million at one point. Later, attackers were able to find a loophole in the smart contract code and steal $60 million from the investors.

Transparency

All transactions stored in the blockchain can be seen by all the participants and in some cases even by the public, depending on the permission model. This is another blockchain property that allows it to be mostly corruption-proof [18].

Pseudo-anonymity

All the blockchain transactions are pseudo-anonymous, meaning users are only identified by the address of their crypto wallet or other account identifiers, but not by their personal data, e.g. their name or house address. This blockchain property does not always mean that all blockchain participants and their transactions are completely anonymous. Most stock exchanges and other services require a form of identification due to the „Know Your Customer“ laws (prevention of money laundering, fraud etc.) [31].

2.3 Structure

This section describes the components and the way they cooperate to achieve a successful implementation. On a large scale, there are several layers working together. These manipulate various data structures that carry out asset transfers and store important data, like the accounts and their balances or the transaction history, in a secure manner [17].

2.3.1 Stacked model

Similarly to the OSI model, a networking system description, blockchain can also be divided into several layers, each having its own significant role in the whole functionality, as seen in Figure 2.5.

Data Layer handles data representation. It describes the block formation from a batch of transactions. The blocks are then connected together, forming a ledger. The whole process uses various cryptography methods to protect data, making the chain tamper-proof.

Network Layer secures the communication with peers in the peer-to-peer networking. Although not directly related to the blockchain network, the lower OSI layers are also needed for a successful implementation. That includes functionality like routing, domain name resolution (DNS) or addressing.

Consensus Layer describes the exact rules for mutual agreement, allowing to maintain consistency across the network. Topics like transaction order, security or incentive mechanisms, which are the subject of this thesis, are included in the consensus layer. Some of the consensus mechanisms are described in detail in Section 2.3.5.

Replicated State Machine Layer defines the way transactions are processed. The goal of this process is to add the new transactions, represented as a block, into the ledger. There are high-level programming languages allowing developers to implement smart contracts, if possible for the particular blockchain implementation. The codes need to be compiled and later executed, generating new transactions that need to be processed.

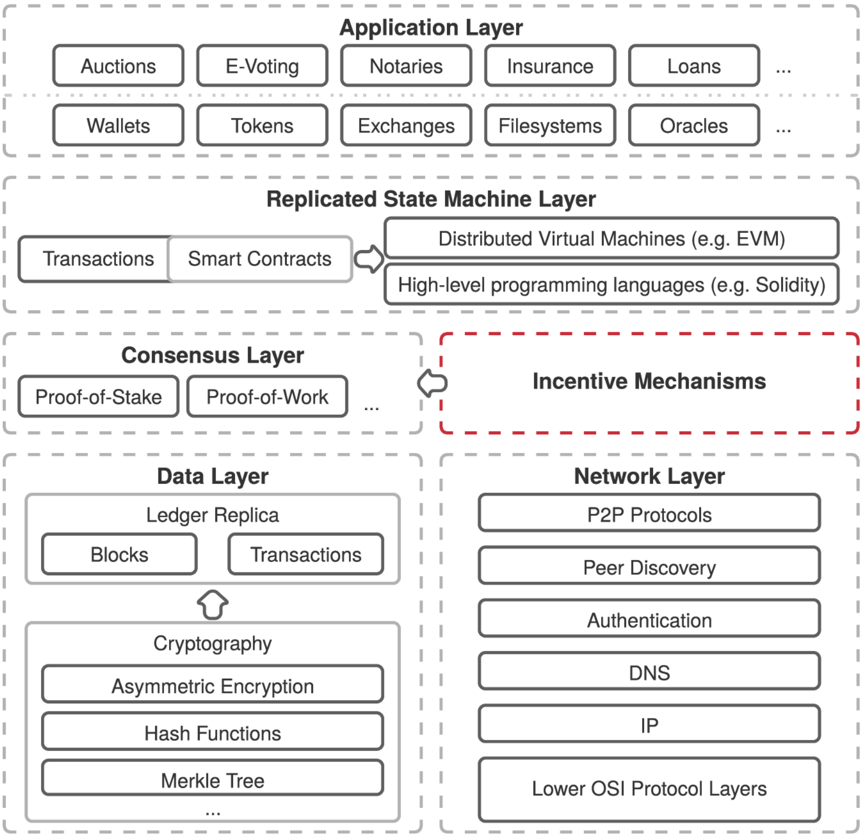

Figure 2.5: A model of the blockchain network implementation [17, 35].

Application Layer includes the end-user services, both basic functionalities and high-level applications. These interfaces can be used by users to communicate with the blockchain network, separating the user from the lower layers.

2.3.2 Blocks

Blockchain is a list of blocks chained together, where each of these contains a batch of **transactions** (Section 2.3.3) and a **header** with other important information. Blocks are identified by a cryptographic hash containing the block header encoded using a secure hashing algorithm, for example, SHA256 or other. Cryptography is further explained in Section 2.1. The identifying hash does not have to be stored in the block header, as it can be computed by every node individually. Another form of identification is the block height, meaning the position of the block in the blockchain (where the position of the first block, which is usually called the genesis block, is 0). The chained structure is created by referencing the previous (parent) block in each block's header [3].

2.3.3 Transactions

In a bank, a transaction is a process of **transferring assets** from one account to another account, changing both accounts' balances. In the blockchain, the meaning remains, but technically, a transaction is a cryptographically signed set of information needed to execute the transfer of currencies and tokens between accounts.

The object needs to be signed using the sender's private key to confirm his identity before being included in the transaction pool as a transaction request. This is achieved using cryptography, which is further explained in Section 2.1. After the transaction is created, a request is added to a list and later gets placed into a block and processed based on the used consensus protocol (Section 2.3.5). Sending a transaction is a paid service and the speed of transaction processing is usually affected by the fee, making the processing faster when the fee is higher [3, 2].

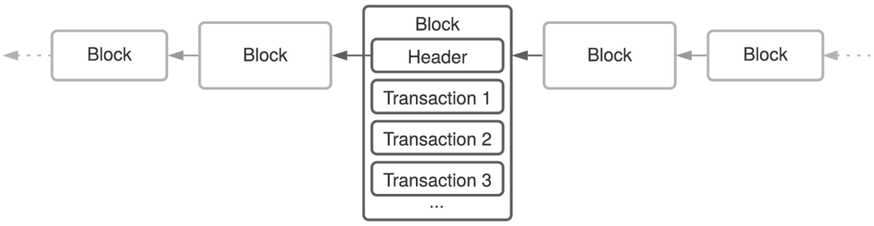

Figure 2.6: Structure of blockchain, showing blocks and transactions.

2.3.4 Nodes

Nodes are computers in the network, which communicate with each other and share information, being the peers in the **peer-to-peer** network. The aim is to connect a large number of independent nodes, allowing the network to function securely and reliably, without the danger of a security attack or censorship.

Nodes also hold information regarding pending transactions. When a node receives an incoming transaction, it gets stored in the **mempool**. The transaction is then sent to other nodes in the network and stays in the mempool until it gets processed and included in a block.

There are two main types of nodes: a full node and a light node. The **full node** contains a copy of the whole blockchain, including all the blocks and the corresponding transactions. It can validate all processed transactions, and query blockchain data, but the hardware requirements are significant. The solution to this problem is the **light node**, which only holds the headers of blocks, without the included transactions. They can validate the block headers and also determine the effect of the included transactions on the network, making the nodes capable of validating them [2].

2.3.5 Consensus mechanisms

A consensus mechanism is a technology that allows distributed systems (e.g. the collection of nodes mentioned in Section 2.2) to securely work together. It affects the way blocks are processed, the system of rewards and the safety measures against fraud [6].

Proof-of-work (PoW)

In the proof-of-work consensus, the blocks are created and chained together by special nodes in the network called **miners**, who use their hardware to solve computationally difficult puzzles [28, 3]. First, the miner gathers a batch of transactions to be put into the new block and checks their validity. The puzzle required to connect the block to the chain is based on finding a pseudo-random value, the block's **nonce**, by random chance. To do that, the miner needs to transform the found value using a cryptographic hash function and see if the final value matches the conditions given by the difficulty, which is periodically recalculated based on the previous blocks' solution search time. In case of higher difficulty, the number of nonces that meet the conditions is lower, therefore it takes longer to find a solution. Once a solution is found, it gets broadcast to other nodes, who verify its validity, accept the new state of the blockchain if valid and remove the processed transaction requests from the list. The successful miner is rewarded according to the incentive mechanism.

Mining requires **computation power** and time. Increased computational power means a bigger chance of finding the solution because more nonces can be generated and checked in a shorter amount of time. The needed hardware and energy costs are too high for individuals to succeed in this competition and miners need to collaborate. The collaborating groups are called mining pools. Participants make use of their united computation power and divide the reward. That makes the individual rewards lower, but the income is steady.

A situation called soft fork may arise when several miners find the solution at once. That results in inconsistencies among the nodes. This state must be resolved, as only one child block (the new block created by the miner) can be chained to the parent block (already existing in the blockchain). To solve this issue, all nodes select the chain with the greatest total amount of executed work to achieve consistency.

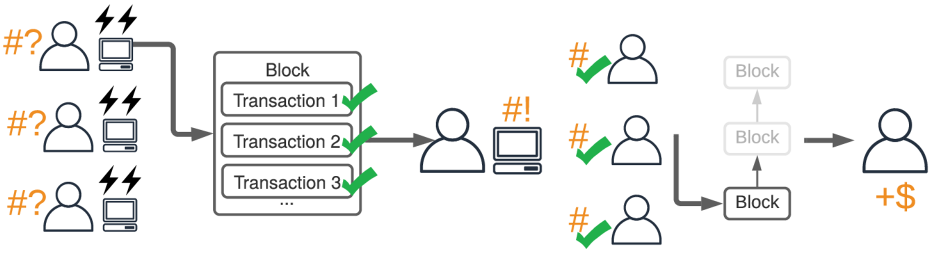

Figure 2.7: Proof-of-work consensus mechanism.

Proof-of-stake (PoS)

To become a validator and start validating blocks in the proof-of-stake consensus, a fixed amount of assets must be locked as a **stake** [27, 26]. Therefore, the main requirement is to have enough assets in the account. **Validators** with higher stakes have a better chance to be chosen to forge (process) the new block. To make the process fair and more randomized, other conditions are usually added to the validator selection process. In the „randomized block selection“ mechanism, the size of the stake is combined with the lowest hash value as an additional condition. The „coin-age based selection“ method adds the number of days the stake has been held. After forging a block, the counter resets and sets a limit when the validator cannot be chosen again. The set of rules is set individually by every cryptocurrency using proof-of-stake.

The chosen validator must check the validity of all transactions in the block. If the validated transactions were fraudulent, part of the stake and the right to further participate as a validator would be taken away. To disrupt the blockchain security without consequences, a node would have to own 51% of the assets. This phenomenon is called the „51% attack“. The risk of this attack is higher for smaller networks, where the investment would be levelled out by the profitability.

After the transactions are validated and the block is created, the validator adds it to the blockchain and is rewarded with the transaction fees.

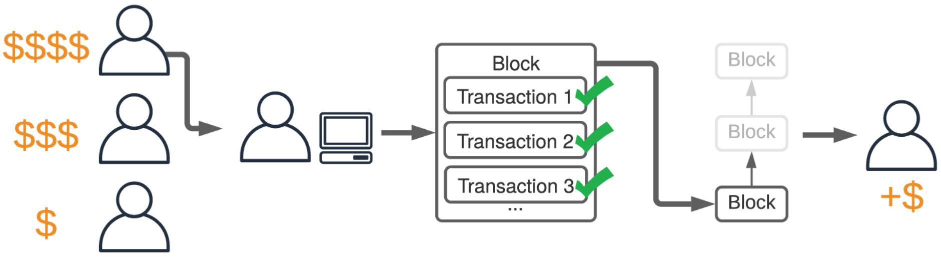

Figure 2.8: Proof-of-stake consensus mechanism.

These two consensus mechanisms, used by many publicly well-known blockchains like Bitcoin, Ethereum, Cardano or Solana, work differently in the sense of block processing and also require the blockchain to be in different parts of the life cycle. A comparison of the two consensus mechanisms is shown in Table 2.1.

Table 2.1: Comparison of properties and behaviour of PoW and PoS.

Proof-of-work	**Proof-of-stake**
Miners need to join big mining pools to successfully mine a block, lowering the number of independent nodes.	It is easier to participate, more participants can join as individual validators.
A lot of energy is required to mine a block.	A lot less energy is needed, making the process cheaper and environmentally friendly.
The process is thoroughly tested, as a lot of big technologies using the consensus have been running for years.	The consensus is still new and has not been tested in difficult conditions.
Miners need hardware and energy, the cost is high. They need to join mining pools to succeed.	Validators need assets to stake. Unavailability (going offline) can lead to the loss of some of the stake.
Once the first (genesis) block is created, the blockchain is ready to run, and transactions can be created.	Existence of virtual assets, that can be staked, is expected.
Sharding (splitting the blockchain into smaller partitions) would mean less security.	Stronger support for shard chains.

Chapter 3

Ethereum

Ethereum is a protocol, whose main purpose is to provide smart contract functionality to various decentralized applications. It can also be used to send and receive the cryptocurrency Ether and other digital assets, thanks to being programmable. The protocol operates a state machine, which hosts all accounts and smart contracts. The changes and computing of new states are performed by the Ethereum virtual machine (Section 3.2). The states are then stored in a blockchain [12].

Ether (also ETH) is a cryptocurrency native to the Ethereum network [11]. It serves as an incentive for the miners (Section 2.3.5), who keep the network functional and safe. Every use of the Ethereum network is therefore conditioned by paying a small transaction fee in ETH. The cryptocurrency can also be traded and invested, similar to other cryptocurrencies like Bitcoin.

The first thoughts behind Ethereum arose at a time when the thoughts and technology behind Bitcoin had already been recognized. Developers needed an infrastructure that is more flexible considering different data types, transaction types and various storage sizes. Developing on Bitcoin meant creating workarounds, possibly working on another off-chain layer or even a completely new blockchain. In 2013, this was the impulse for Vitalik Buterin, who proposed a Bitcoin extension. The proposal would bring contracts as a new technology, but it was not approved due to the changes being too extensive. Vitalik then started working on the Ethereum Whitepaper [36] with Dr Gavin Wood, who joined him as a CTO in an effort to create a programmable blockchain technology, now known as Ethereum.

3.1 Structure

This section describes structure details exclusive to the Ethereum protocol, extending the information provided in Section 2.3.

Blocks

The Ethereum blocks are added by changing the EVM state after being processed (further explained in Section 3.2). They contain the following fields[1]:

- **timeStamp** - time of the block processing

[1] https://ethereum.org/en/developers/docs/blocks/

- **blockNumber** - number of blocks in blockchain
- **baseFeePerGas** - transaction's minimum fee per gas to be included
- **difficulty** - how difficult was the block's processing
- **mixHash** - block's identifier
- **parentHash** - reference to the parent block
- **transactions** - included transactions
- **stateRoot** - state of the system
- **nonce** - proof of the carried out work (explained in Section 2.3.5)

Transactions

A transaction can be initiated by an externally-owned account, an entity with a balance owned and controlled by a user. The transfer does not always have to take place between two accounts owned by users, in Ethereum, a user can also transfer ETH to a smart contract account, executing a code. The two types of accounts are further explained in Section 3.1.

After the initiation, the user's node broadcasts the transaction request. At that time, the transaction is not yet performed and valid, it needs to get verified first according to the consensus rules (further explained in Section 2.3.5). Once valid, the changes can get broadcast to the whole network [2].

The transaction object contains various information important for the transfer. For Ethereum, the values are following[2]:

- **recipient** - receiver's address
- **signature** - confirmation of the transaction being authorized by the user
- **value** - amount of transferred ETH
- **data** - optional additional data
- **gasLimit** - maximum possible amount of gas consumed
- **maxPriorityFeePerGas** - maximum possible amount of gas included as a miner's tip
- **maxFeePerGas** - maximum amount of gas to be paid for the transaction (incl. baseFeePerGas and maxPriorityFeePerGas)

Accounts

An account is defined as an entity with an ETH balance that can transfer the assets to other accounts, similar to a bank account [1]. There are two types depending on the entity managing the account. An **externally-owned account** is managed by an authorized person. The creation of this account is cost-free and allows to send and receive ETH or tokens. On the other side, a **contract account** is paid due to the smart contract

[2] https://ethereum.org/en/developers/docs/transactions/

code utilizing storage. Transactions can be sent to this account, but the account itself can only send transactions if triggered by an incoming transaction. The incoming transactions execute the contract's code.

The account data structure contains following fields:

- **nonce** - counter of sent transactions/created contracts
- **balance** - ETH balance
- **codeHash** - identification of the account on the EVM (Section 3.2)
- **storageRoot** - hash value, root of the Merkle Patricia tree that contains the account's storage contents

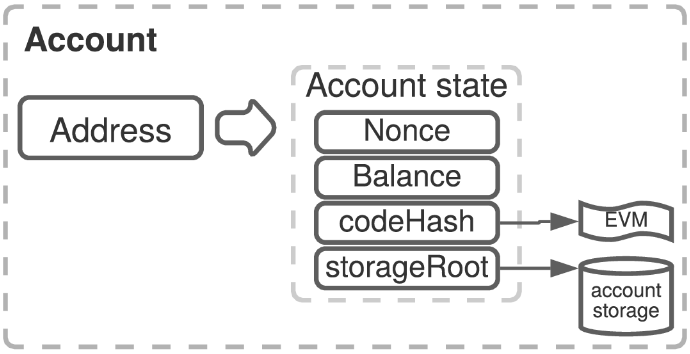

Figure 3.1: The account data structure [1].

3.2 Ethereum Virtual Machine (EVM)

The Ethereum Virtual Machine (EVM) in an environment simulating a computer. It works as a **virtual CPU** managed by each of the nodes in the network. It can also be described as a program, which can be run using the user's computer, creating a new node. As a state machine, its role is to maintain data stored as a tree structure, which is also called the Merkle Patricia Tree. This data structure keeps a list of accounts and a state, which changes with new blocks. The change is accomplished using the state transition function, which processes the previous state and a set of new transactions into a new valid state. EVM can also process and execute smart contracts [37, 13].

3.3 Smart contracts

Smart contracts allow users to create payment agreements, similar to regular legal contracts regarding fiat money or other property [39, 8]. On the Ethereum network, they are represented as scripts that can be executed by users interacting with the smart contract. The main task is to transfer assets if the defined conditions are fulfilled. There are high-level languages created specifically to develop smart contracts, like Vyper

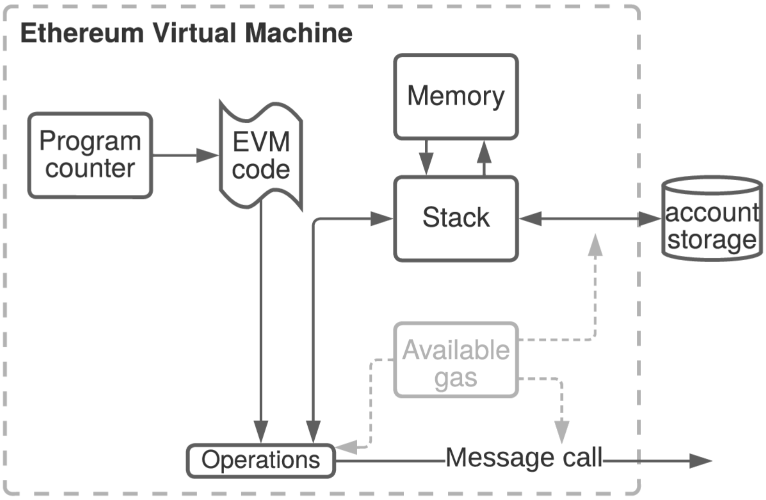

Figure 3.2: Structure of the EVM [14].

or Solidity. These tools enable the creation of very complex codes, which are essential for creating decentralized financial services (DeFi, further described in Section 3.3.2).

3.3.1 Tokens

Tokens can represent many different things: financial assets, fiat currencies, game points, collectables, certificates or even real-life items. They are implemented as smart contracts, which hold values and define basic operations, like balance checks or token transfers. An example of a token implementation [3] can be seen in Listing 3.1. The values are stored in a contract as a structure of values mapped to users' addresses. There are several standards, which serve as a guideline for token implementation and ensure compatibility with existing exchanges and wallets. Contracts can trade with each other without the need of creating new interfaces for each one of them. These standards are also called **ERC**, which stands for „Ethereum Request for Comment“ [7].

ERC-20 Tokens

The ERC-20 standard [4] describes Fungible Tokens, meaning non-unique interchangeable assets, where 1 Token has exactly the same value as another Token. One of the use cases are the stablecoins, tokens maintaining fixed value according to a fiat currency. The native Ethereum token, Wrapped ETH (WETH), is an ERC-20 analogue of the ETH currency. Its value is the same as the value of ETH and it can be traded for other tokens.

For a smart contract to abide by the ERC-20 standard, the following methods must be implemented:

- carry out token transfers
- find out the current token balance for one account

[3] https://github.com/OpenZeppelin/openzeppelin-contracts

[4] https://ethereum.org/en/developers/docs/standards/tokens/erc-20/

- find out the total token amount
- amount of tokens in an account allowed to be spent by a third-party

```
pragma solidity ^0.4.24;
import "./IERC20.sol";
contract ERC20 is~IERC20 {
    mapping (address => uint256) private _balances;
    mapping (address => mapping (address => uint256)) private _allowed;
    uint256 private _totalSupply;
    function name() public view returns (string) {...}
    function symbol() public view returns (string) {...}
    function decimals() public view returns (uint8) {...}
    function totalSupply() public view returns (uint256) {...}
    function balanceOf(address _owner) public view returns (uint256 balance) {...}
    function transfer(address _to, uint256 _value) public returns (bool success) {...}
    function transferFrom(address _from, address _to, uint256 _value) public returns (bool
        success) {...}
    function approve(address _spender, uint256 _value) public returns (bool success) {...}
    function allowance(address _owner, address _spender) public view returns (uint256
        remaining) {...}
}
```

Listing 3.1: An example of ERC-20 token implementation using Solidity language and the OpenZeppelin library.

ERC-721 Tokens (NFTs)

The ERC-721 standard [5] is used for Non-Fungible Tokens (also NFTs), which represent unique items. They are not interchangeable and tokens can have different values in one smart contract. Items like collectables, access keys, certificates, real estate, copyright or gaming assets can be offered this way. The standard offers a new field called tokenId (type uint256). This value, in combination with the contract address, uniquely represents the token. In most cases, the token is also linked to an image. One of the currently most well known NFT projects, that inspired the ERC-721 standard, is called CryptoPunks.

For the ERC-721 standard, the following methods must be implemented:

- find out the number of NFTs in one account
- find out the owner of an NFT
- transfer the ownership of an NFT to another account
- get or change the approved contract address for an NFT
- approve another account or a third party to manage an account's assets

ERC-777 Tokens

The ERC-777 standard [6] expands the ERC-20 standard, meaning it deals with Fungible Tokens. The standard is backwards compatible, therefore the ERC-777 contracts allow for the same interaction as if they were ERC-20 abiding contracts. The main offered

[5] https://ethereum.org/en/developers/docs/standards/tokens/erc-721/

[6] https://ethereum.org/en/developers/docs/standards/tokens/erc-777/

improvement in comparison with ERC-20 is the hooks. They are special functions, which get executed once a token transfer is detected. The token transfer and the following notifies or rejects can be executed as one transaction and do not require two method calls. Decimals are also slightly improved to eliminate confusion in the development phase.

ERC-1155 Tokens

The ERC-1155 standard [7] is used for contracts working with multiple token types, including both Fungible and Non-fungible tokens, and therefore combining the functionality of the ERC-20 and ERC-721 standards. As a result, most of the methods work with arrays of values.

It supports the following features:

- batch transfer - transfer of multiple assets
- batch balance - find out the balance of multiple assets
- batch approval - set approved operators
- hooks - receive hooks to ensure successfully completed transfer
- NFT support
- safe transfer rules

3.3.2 Decentralized Finance (DeFi)

DeFi is a financial system that can be an alternative to fiat money and centralized financial services, which provide a way to loan, trade or invest [10]. Thanks to being decentralized, it provides access to the services and products to everyone with an internet connection. These products use smart contracts to manage the money, meaning all the processes are automatized and mostly free of long processing time and possible human errors. The activity is pseudonymous (i.e. linked to the address rather than to a person's identity) and the system is transparent - there is no need to trust companies, whose internal processes are usually not available to the public. DeFi first became popular during the summer of 2020, increasing the Ethereum network activity and making the fees more volatile. Since then, it has significantly gained popularity, initiating the discussion regarding the incentive mechanism and ways to make it more efficient and user-friendly.

3.4 Incentive mechanism

As explained in Section 2.3.5, new transactions are processed by miners or validators. To compensate them for the used computational power, time or stakes, an incentive mechanism is introduced [14]. The unit used to measure the required computational power for the transaction processing is called **gas**. The incentives are paid by users who send a transaction, which means that the price must be low enough to motivate users to use the network, but also high enough to motivate miners to keep the network running and make security attacks less advantageous. The price per gas is usually given in gwei, which equals 10^{-9} ETH.

[7] https://ethereum.org/en/developers/docs/standards/tokens/erc-1155/

3.4.1 The first price auction mechanism

Before the London hard fork, which brought many changes regarding the incentive mechanism, a system called first price auction was used [29, 20]. The user has to submit two values, the gas limit and the gas price.

The **gas limit** signifies the total computational work limit for the transaction. In case of a limit higher than what is actually needed, the difference is given back to the user. If the limit is too low, the changes get reverted, but the fees paid for the work already carried out do not get returned.

The **gas price** is arbitrarily chosen by the user. This system can lead to high gas price volatility and overpaying, because users are unaware of the current average price and can only determine the value based on predictions and suggestions, using for example oracles. The gas price can also be affected by time-sensitive transactions, for which users set a higher gas price to speed the process up. The total fee paid for the transaction processing is compared using the gas limit and gas price:

$$Fee = GasLimit * GasPrice \tag{3.1}$$

A standard transaction, such as sending ETH to another address, requires 21 000 gas units. If the gas price chosen by the user is 20 gwei, the total amount paid by the user is $21000 * 20 * 10^{-9}$ ETH, i.e. $42 * 10^{-5}$ ETH.

The blocks had a **fixed size** of 15 million gas, meaning the users had to wait longer for their transactions to get completed during high demand, especially if the gas price was lower than the gas price of other transactions. Users would often overbid in case of time-sensitive transactions, making the network more unpredictable for other users. One of the solutions was proposed and implemented in EIP-1559.

3.4.2 EIP-1559

EIP-1559 is an improvement proposal regarding the incentive mechanism included in the upgrade called London, which came into effect on 5 August 2021. The goal is to lower the volatility of fees and make them more predictable and user-friendly, in comparison to the previously used incentive mechanism [38, 29].

To resolve the network congestion during high demand, the **variable block size** got introduced. While the **target gas** still keeps its value (15 million gas), the new block size, also called the block **gas limit**, is a value two times higher than the target (30 million gas). The completely full blocks, which reach the block gas limit, are not very common under usual demand. The fee gets increased if blocks are fuller than the target and decreased in the opposite case, motivating the users to send their transaction immediately or to wait for lower demand.

The fee got divided into two parts: the base fee and the tip. The **base fee** is the minimum gas price that would allow the transaction to get included in a block, serving as a guideline for new transaction requests. The amount is recalculated for every block based on the utilization of the previous block, as shown in Equation 3.2.

$$BaseFee_{h+1} = BaseFee_h * (1 + \frac{1}{d} * \frac{GasUsed_h - GasTarget}{GasTarget}) \tag{3.2}$$

The value d refers to the base fee denominator, which determines the largest change of the base fee between two blocks. Currently, $d = 8$, meaning the base fee can increase or decrease by as much as $1 + \frac{1}{8}$, or 1.125x.

After the processing of the transaction, the base fee gets **burned** (destroyed). To prioritize their transactions, users can also offer a **tip** for the miner. This is a necessary addition to the mechanism, motivating the miners not to mine empty blocks for the same block reward and less computational power. The block reward is an ETH amount that gets generated as an additional incentive for each of the mined blocks. The total fee paid is then calculated according to the following equation:

$$Fee = GasLimit * (BaseFee + Tip) \tag{3.3}$$

Users can also optionally set the **maximal fee per gas**, which refers to the largest amount the user is willing to pay for the transaction. The gas price is then the lower value of the maximal fee and the sum of the base fee and the tip. This way, users can choose the highest fee they are willing to pay without actually overpaying [14].

The changes in the incentive mechanism can be seen in Table 3.1, which compares the new mechanism, introduced in the London upgrade, to the previously used first price auction mechanism.

Table 3.1: Comparison of the incentive mechanism before and after EIP-1559.

Legacy mechanism	**EIP-1559**
The fee consists of the gas price set by the user based on the current market and transaction priority.	The fee consists of a base fee (based on the previous block) and a preferential tip.
The whole fee goes to the miner.	The tip goes to the miner and the base fee gets burned.
All blocks have a fixed size.	The block size can change based on current demand (2x the target size).
Users set fees based on predictions and transaction priority, causing high volatility.	Part of the fee is based on previous blocks, making it more predictable and less volatile.

The EIP-1559 incentive mechanism is still new and can act in unexpected ways. In the following chapters, the behaviour of the mechanism is investigated and demonstrated using simulation experiments, with a focus on possible weak points and improvements.

Chapter 4

Simulation and implementation

To be able to successfully investigate the behaviour of the incentive mechanism, several simulation experiments have to be proposed. In order to carry out the experiments, a simulation model, suitable data and data processing scripts are needed. This chapter describes the process of choosing and implementing necessary tools, as well as finding relevant data.

4.1 Simulation modelling

The process of modelling and simulations consists of several parts and steps [24]. The first important part is the **system**, a set of components related to each other in various ways. In this case, the system is the new incentive mechanism, which consists of parts such as the base fee, the tip or the block utilization, connected to each other by mathematical relations.

In order to implement a copy of the system, a **model** has to be created. The model is a simplified way to describe the important attributes of the system. A lot of models of the same system can exist, each one of them considering different attributes, which are important for the purpose of the research. The draft of the model given, for example, in the form of mathematical equations, is called an abstract model. The implementation of the abstract model is then called a simulation model.

Finally, new knowledge can be obtained using the simulation model, further also referred to as a **simulation environment**. First, simulation experiments have to be proposed, defining the goal of the experiment and the used parameters. The simulation environment is run using the given parameters. The results can be evaluated and discussed.

The verification and the validation are also important during the model implementation. The **verification** of the model indicated the determination of whether the simulation model is correct and the behaviour and the structure match the abstract model. The **validation** of the model determines whether the abstract model matches the system. A model can never be fully validated, as the correctness of the results and a lack of errors cannot be proved. The whole process is described in Figure 4.1.

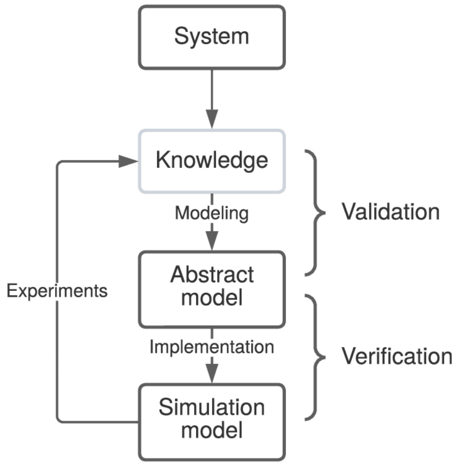

Figure 4.1: The process of modelling and simulation [24].

4.2 Simulation environment

Two simulation models were considered based on the goal of this thesis and the following propositions of the experiments. **BlockSim** [1], created by Maher Alharby and Aad van Moorsel, is a simulation model of blockchain implemented in Python. In the input, users can choose from a base model, a Bitcoin model and an Ethereum model. In the Ethereum model, users can set parameters regarding the network layer, the consensus layer and the incentives layer. The model was designed to work for various blockchain systems in general and can be easily extended based on the purpose of the research.

The **abm1559** simulation environment [2] was created by Barnabé Monnot, a member of the Ethereum Foundation, a non-profit organization focused on technologies related to Ethereum. It was created before the adoption of the EIP-1559 improvement to demonstrate the basic functionality and rules, the transition between the two incentive mechanisms and the behaviour of different user types, as well as various changes to the principle. The simulation input is an array of values representing the assumed number of new transaction requests between two blocks and also several parameters regarding EIP-1559 and the behaviour of the users.

For the purpose of this thesis, the **abm1559** simulation environment was selected. It is more focused on the details of the EIP-1559 incentive mechanism, disregarding the details of the network and consensus layers. The only change made to the simulation environment was an addition facilitating the setting of the parameters, which can now be done in a notebook rather than directly in the files of the library. The simulation model was previously used in several research papers, making it functional and credible.

[1] https://github.com/maher243/BlockSim

[2] https://github.com/ethereum/abm1559

4.3 Data and data processing

In order to propose true to life experiments, historical data and statistics were used as a simulation input, as well as a comparison regarding the results. Several scripts were implemented and used to process the data, preparing them for further use in the simulation model.

Mempool Statistics by Jochen Hoenicke contain the number of pending transactions, paid fees and total mempool weight in gas, collected by the author's full node. While no data regarding mempool can take into account all pending transactions, because the Ethereum network does not use a centralized global mempool, it can serve as an example of the general state of the network during various network utilization. For this reason, mempool weight statistics are used to calculate the demand scenarios used directly in the following experiments or as a hint in the mathematically computed distributions of incoming transactions. The raw statistics and the file explaining the data structure are available on the author's website [3].

Blockchain Data regarding blocks and transactions were used in combination with the mempool statistics to calculate input data and statistics for the comparisons. A collection of scripts, called Ethereum ETL [4], was used to get the raw blockchain data in the CSV format. A node client or a node provider is necessary for this step.

Script mempoolstats.py uses mempool statistics to estimate the number of new transaction requests between every two blocks. Input files used by this script are the mempool statistics, which should be restricted to the given time period if possible for execution time purposes, and the file containing mined block data for the given time period. For simulation purposes, all transactions are assumed to be basic transactions using 21 000 gas, therefore the number of transaction requests is calculated as $transactions = total\ gas/21000$. The frequency of mempool statistics records is about 1 per minute. Considering the average block time (amount of time it takes to process a new block) being generally about 10 to 20 seconds, the blocks are mapped to the corresponding mempool statistics records and the calculated value is used for all of the mapped blocks. The formula for calculating the number of new requests is as follows:

$$new = mempool\ state_{n+1} - mempool\ state_n + processed\ txns \tag{4.1}$$

Statistics concerning cancelled transactions were not taken into consideration due to the unavailability of data. The resulting array of values is in the *results/demand-scenario.txt* file.

Script txnsfees.py calculates the average gas price on historical data. Input files used by the script are the file containing mined block data for the given time period and the file containing transaction receipts for a matching time period. The transaction receipts are mapped to the corresponding blocks using block numbers and the gas price of every transaction is collected. The values are then represented as both the arithmetic

[3] https://jochen-hoenicke.de/queue/mempool.js

[4] https://github.com/blockchain-etl/ethereum-etl

mean and the arithmetic median in the *results/average-fees.txt* file. The selected array can be plotted on a graph to see the average gas price for each of the blocks.

simulation.ipynb Jupyter Notebook creates the demand scenario and allows to set the parameters, run the simulation and export the results. It contains the main simulation function used by the author of the abm1559 simulation environment.

graphs.ipynb Jupyter Notebook processes the exported simulation results and plots them on graphs, which are later used in this thesis.

Chapter 5

Simulation experiments

The experiments are divided into two parts, the **typical demand** and the **high demand** with a peak in incoming transactions. Each of these sections contains a description of the simulation input, which is a list of new incoming transaction requests, further referred to as a „demand scenario“. The paragraph contains the source of the data and the motivation to perform simulation experiments using this particular scenario, as well as the input displayed in form of a graph.

The general description is followed by several experiments. Each of them consists of the goal of the experiment, the methodology, a table with parameters and the results, that consist of the final graphs and an evaluation of the results. Finally, two solutions are proposed, including both a proposal for a change in the mechanism based on the experiments and a suggestion on how to reduce the network load in general.

5.1 Typical Ethereum network utilization

In this group of experiments, the demand scenario is based on historical data and statistics collected when the first price auction incentive mechanism was being used. Data collection and processing are further described in Section 4.3. The selected time period generally represents the usual utilization of the network, making the behaviour predictable during other time periods, as all the same rules apply. However, an unexpected behaviour change during various market changes cannot be ruled out, and the results cannot be represented as a universal truth regarding any unexpected situation.

The selected time period is one day, 6 June 2021, based on the high volatility of transaction requests in the mempool, most likely caused by the price of ETH significantly dropping during May and June 2021 [1]. Volatile prices are a common occurrence and the reaction of the market falls within the usual and expected network utilization. The demand scenario can be seen in Figure 5.1.

5.1.1 Experiment I

Goal. The goal of the experiment is to compare the two incentive mechanisms using historical legacy data in an EIP-1559 simulation from the point of view of gas prices.

[1] https://www.coingecko.com/en/coins/ethereum

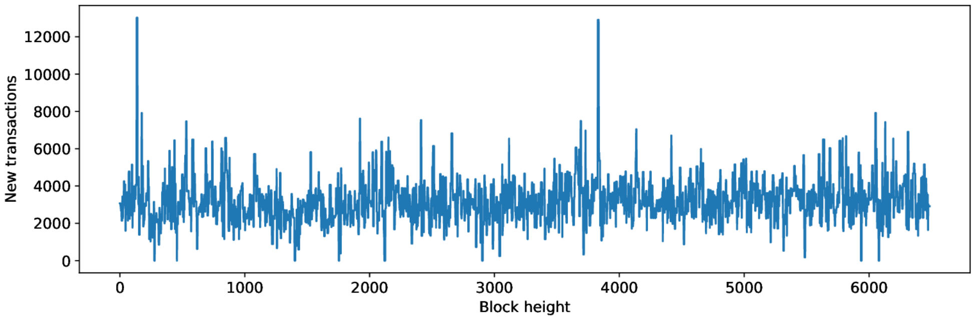

Figure 5.1: Used demand scenario, 6 June 2021.

Methodology. The parameters used in this experiment are coincident with the parameters currently used for all EIP-1559 transactions. The users send regular transactions and expect them to be included in the next 5 blocks. The selected initial base fee is 10 gwei. The simulation run length, 6 486 blocks, matches the number of blocks mined during the selected day. An overview of the used parameters can be seen in Table 5.1.

Table 5.1: Parameters used in Experiment I.

Parameter	**Value**
Target gas	15 000 000
Maximal gas limit	30 000 000
Maximal base fee change denominator	8
Initial base fee (Wei)	$10 * (10^9)$
Expected transaction processing time (blocks)	5

Results. The comparison of gas prices during the legacy and the EIP-1559 incentive mechanism can be seen in Figure 5.2 and Figure 5.3. While the average price between blocks 3000-3500 and 4500-5000 could get as high as 150 gwei compared to the usual 25–50 gwei in the previous blocks, the largest difference between a low and a high spike is ~16 gwei, making the minimum transaction fee more **predictable**. The EIP-1559 incentive mechanism lowers the gas price volatility for this data sample. The volatility of the average tip in Figure 5.2 is low, but the simulation only takes regular transactions into consideration. Generally, the tip can get much more volatile, due to various time-sensitivity of transactions and other, often unpredictable, factors.

While the fee volatility got lower, the average fees by themselves maintained the **same average values** as before during the usual network utilization. According to the simulated results, the EIP-1559 incentive mechanism does not lower the gas price in general.

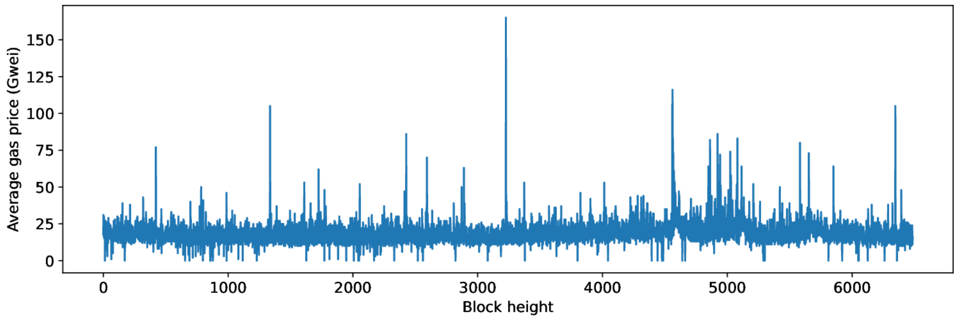

Figure 5.2: Historical average gas prices, 6 June 2021.

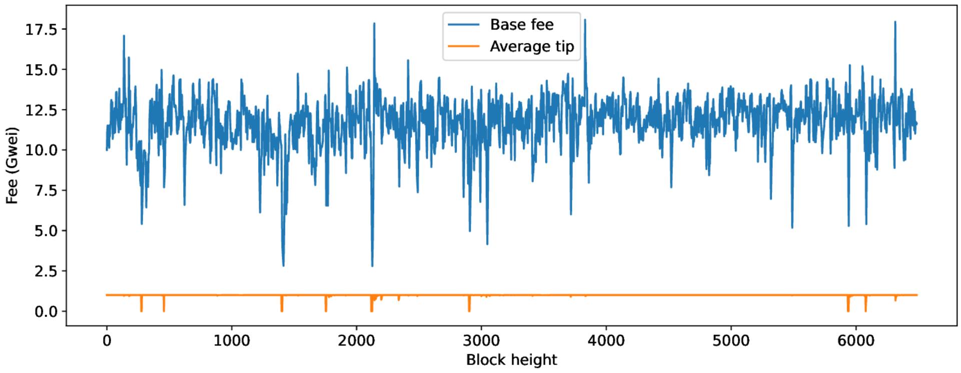

Figure 5.3: EIP-1559 gas prices simulation using data from the same day.

5.1.2 Experiment II

Goal. The goal of this experiment is to compare the fees and the network performance using the usual block gas limit (two times the target) and a higher block gas limit (four times the target).

Methodology. The simulation run length remains the same as in the previous experiment, 6 486 blocks. The maximal gas limit of a block was changed to a value two times larger than in the previous experiment, meaning four times the target gas. Other parameters remain the same. An overview of the used parameters can be seen in Table 5.2.

Table 5.2: Parameters used in Experiment II.

Parameter	**Value**
Target gas	15 000 000
Maximal gas limit	60 000 000
Maximal base fee change denominator	8
Initial base fee (Wei)	$10 * (10^9)$
Expected transaction processing time (blocks)	5

Results. In Figure 5.4, the base fee trend remains similar to the previous experiment, including the fluctuations below the average, except for some of the **fluctuations** above the average, which are now limited. The general trend remains the same.

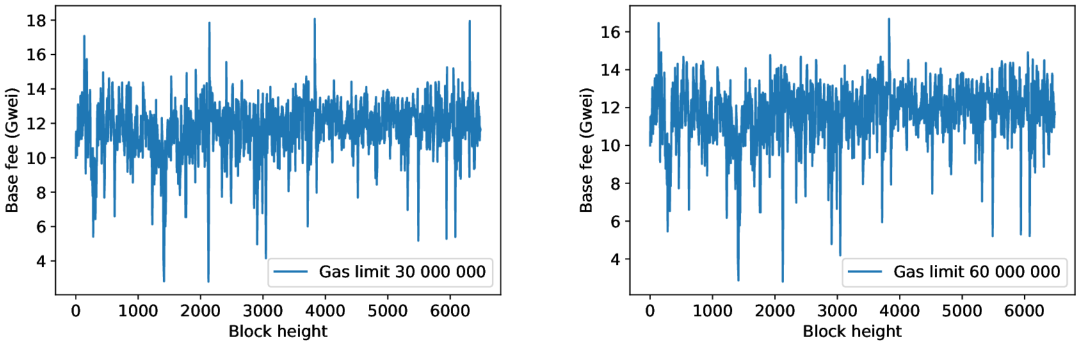

Figure 5.4: Comparison of the base fee with the maximal gas limit of 30 million (left) and the maximal gas limit of 60 million (right).

In Figure 5.5, the block utilization in both the usual and the higher follow the normal distribution. In the figure with the lower gas limit, there is a slight increase of **completely full blocks**. On the contrary, the blocks in the higher gas limit do not appear to be much fuller than usual and there are almost no completely full blocks.

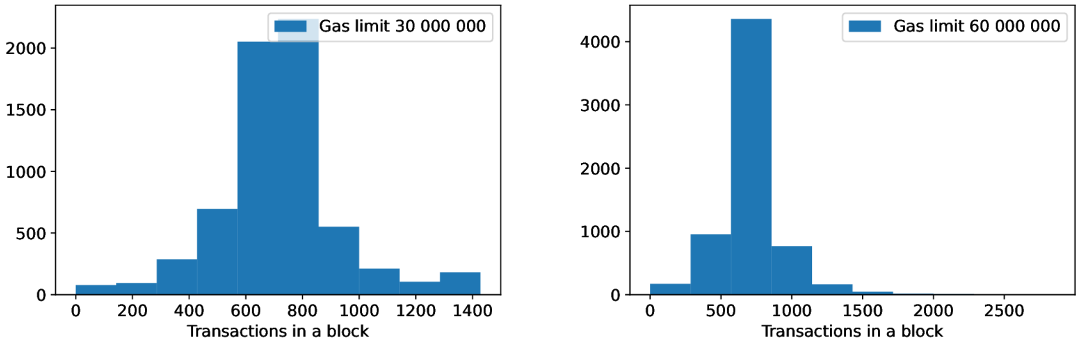

Figure 5.5: The number of transactions included in the blocks with the gas limit set to 30 million (left) and 60 million (right).

The mempool statistics, seen in Figure 5.6, show the higher gas limit mempool almost empty compared to the one with a lower gas limit, which experiences small spikes that are usual during the typical demand and are not detrimental to the network stability. In the context of the usual demand, such a change would be virtually pointless and may even make the network **slower** due to the less performant nodes, which may not be able to deal with larger blocks.

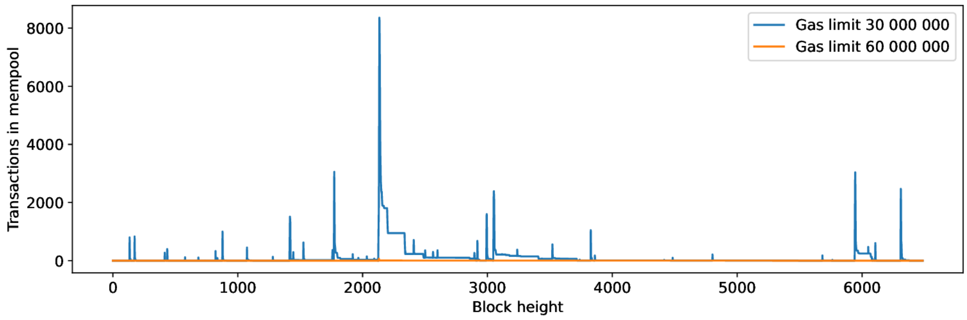

Figure 5.6: The number of transactions in the mempool, comparing the usual and the higher gas limit.

5.2 Corner cases

This section investigates the behaviour of the network and the new incentive mechanism under a high volume of transactions. The demand scenario was inspired by the demand during the Otherside NFT launch on 30 April 2022. The historical mempool statistics were too unstable to use in a simulation due to the large volume of failed transactions and the high volatility during the demand fluctuation, hence a mathematically computed demand scenario, inspired by the real historical values, was used. The peak seen in the historical data lasts around **100 blocks** and the highest value reaches about **50 000 new incoming transactions**. For performance purposes, the number of transactions, the gas limit and the gas target were all divided by 10. The demand scenario before and after the peak is taken over from the previous experiments, investigating the usual demand, and divided by 5, due to the demand being generally higher during the day of the mint. Therefore, all of the results are scaled-down and the changes have to be taken into account in the result analysis. The historical data and the computed demand scenario can be seen in Figure 5.7.

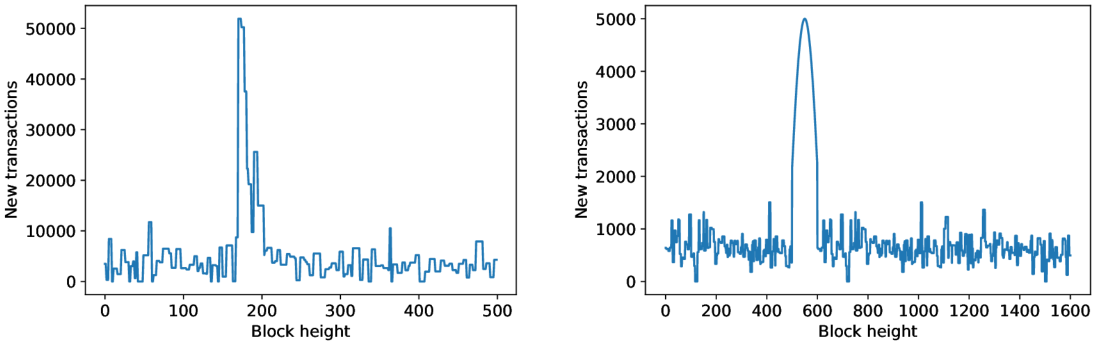

Figure 5.7: The historical demand according to the mempool statistics (left) and the mathematically computed demand scenario (right).

5.2.1 The Otherside NFT Mint

The Bored Ape Yacht Club is a collection of very popular NFTs, which can get sold for prices as high as several thousand ETH. With the total volume traded being almost 560 thousand ETH (May 2022), the collection is regularly featured in the top collections on the homepage of the NFT marketplace OpenSea.

Otherside is an upcoming metaverse game, created by the same author, Yuga Labs. The Bored Ape NFTs will be used as characters in the emerging ecosystem. On 30 April 2022, the authors launched the mint of virtual world land plots (sold also as NFTs) and the very high demand caused problems regarding the **network stability** and **gas prices**. These problems got resolved in about 3 hours.

During the 3 hours following the launch, the base fee rose to values as high as 6 000 gwei and the network got congested to the point of transactions being stuck for hours or even failing. Etherscan, an Ethereum block explorer, experienced outages due to a large amount of data. Because the fees paid for the gas cannot be retrieved, some users lost up to $4 500 per transaction. Users spent over $175 million on gas, and due to the new rules of the EIP-1559 incentive mechanism, a large amount of ETH also got burned, making ETH deflationary [4].

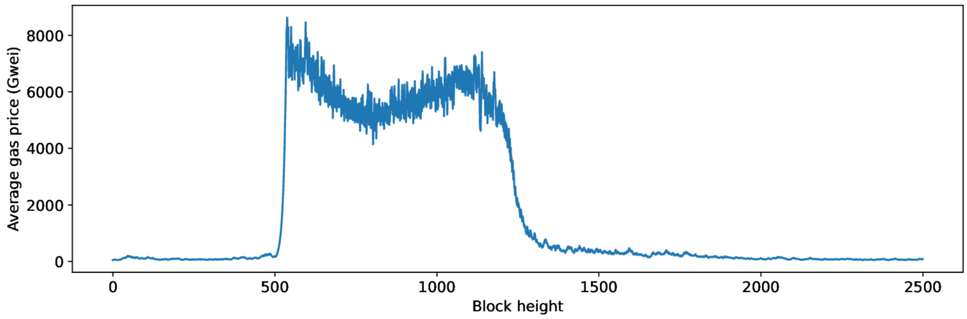

Figure 5.8: Gas price chart showing the peak during the Otherside NFT launch.

Table 5.3: ETH statistics on the day of the Otherside NFT launch and the day after. Source: https://etherscan.io/chart/ethersupplygrowth

	30 April 2022	**1 May 2022**
Total Ether supply	118 963 556.683	118 906 786.794
Daily block reward	+ 12 696	+ 12 762
Daily uncle incl. rewards	+ 25.125	+ 23.813
Daily uncle rewards	+ 688.75	+ 649
Daily Eth2 staking rewards	+ 1 504.817	+ 1 503.64
Daily burnt fees	- 4 282.812	- 71 718.341
Total	10 631.88	- 56 779.888

5.2.2 Experiment III

Goal. The goal of this experiment is to investigate the behaviour of the network under very high demand and to compare the simulation results with the real scenario. as well as with a first price auction simulation.

Methodology. The parameters used for the simulation match the currently used incentive mechanism but are scaled down for performance purposes. The target gas and the maximal limit are therefore divided by 10 to match the scale of the incoming transactions. The initial base fee, which was set to 10 gwei in Section 5.1, is now also divided by 10 and set to 1 gwei to match the other experiments. The expected transaction processing is now set to 100 blocks compared to the usual 5, because users pay very high fees and are expected to try and wait out the network congestion, hoping to successfully complete their transaction without significant money loss. The value of the NFTs, which inspired the experiments (as explained in Section 5.2.1), is also expected to be high, making the users even more determined to complete their transactions. The simulation run length is 1 600 blocks, consisting of 100 blocks of the peak demand and 1 500 blocks of usual, slightly higher, demand. An overview of the used parameters can be seen in Table 5.4.

Table 5.4: Parameters used in Experiment III.

Parameter	**Value**
Target gas	1 500 000
Maximal gas limit	3 000 000
Maximal gas limit (first price auction simulation)	1 500 000
Maximal basefee change denominator	8
Initial basefee (Wei)	$1 * (10^9)$
Expected transaction processing time (blocks)	100

Results. The base fee can be seen rising sharply, starting around block 600. This reaction is significantly delayed in comparison with the demand because the peak demand starts around block 500 and ends around block 600. The highest base fee value is over 20 gwei, which can be as much as **100x larger** than the base fee values right before the peak, matching the real scenario seen in Figure 5.8. The high volatility in base fee values during the decrease is likely caused by the simulated groups of transactions trying to get included in the block that matches their conditions, like the base fee or the expected payoff (calculated in the simulation environment). In reality, the users are not expected to act in groups, therefore the volatility can be omitted from the analysis, leaving the general trend. The gas fees cannot be compared with simulated prices during the first price auction, because the gas price would be arbitrarily chosen by the users, making the fee unpredictable. It can be assumed, that the volatility would be higher due to the lack of rules and regulations. The fees by themselves may be higher too, because the analysis does not take the arbitrary tips, send in addition to the base fee, into account.

The rising number of transaction requests in the mempool can be seen around block 500, right before the sharp base fee growth. The network is **fully occupied** processing a large number of transactions for about 450 blocks, before returning to the usual course. Assuming an average of 3 blocks being processed every minute, the processing of 450 blocks would take about 2.5 hours, roughly matching the time of the network congestion indicated in Section 5.2.1. The simulation of the block utilization corresponding with the first price auction mechanism indicates, that the transaction processing would be much slower, **congesting** the network for more hours, and possibly making it completely unusable.

Table 5.5: Example of specific values from the simulation of Experiment III.

Block	**Base fee**	**Incoming transactions**	**Decided transactions**	**Included transactions**	**Mempool length**
602	0.43889	634	53	142	21 835
603	0.49309	608	42	142	21 735
604	0.55399	582	55	142	21 648
605	0.62240	598	44	142	21 550
...	...	...	...	...	...
695	16.10881	1 294	3	142	13 920
696	18.09825	1 256	0	3	13 917
697	15.93098	1 314	4	142	13 779
698	17.89846	1 251	0	12	13 767

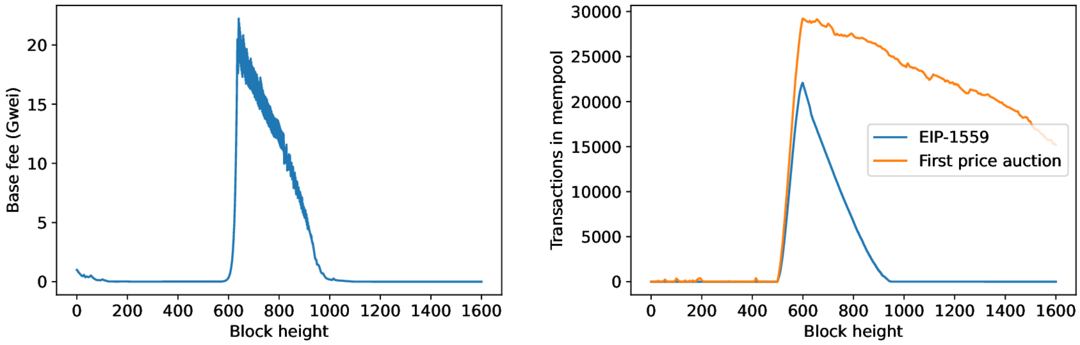

Figure 5.9: The base fee (left) and the number of transactions in the mempool (right) during the high demand scenario.

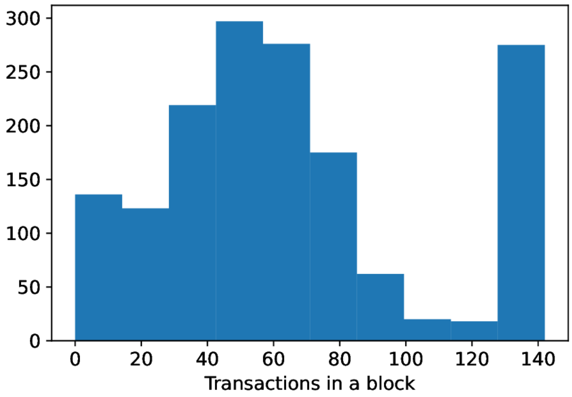

Figure 5.10: Histogram of the number of transactions per block (block utilization) during the high demand scenario.

The block utilization histogram, seen in Figure 5.10, matches the histogram of the block utilization during usual demand in Figure 5.5. The exception is the number of blocks containing the largest number of transactions possible, which is almost as high as the number of blocks matching the gas target. The almost empty blocks may be omitted due to inaccuracies in the simulation explained above.

5.2.3 Experiment IV

Goal. The goal of this experiment is to investigate the behaviour of the network under very high demand, comparing the usual block gas limit (two times the target) and a higher block gas limit (four times the target).

Methodology. The maximal gas limit of a block was set to a value two times higher than in the previous experiment, now being four times the target gas. The simulation run length, the demand scenario and other parameters remain the same. An overview of the used parameters can be seen in Table 5.6.

Table 5.6: Parameters used in Experiment IV.

Parameter	**Value**
Target gas	1 500 000
Maximal gas limit	6 000 000
Maximal base fee change denominator	8
Initial base fee (Wei)	$1 * (10^9)$
Expected transaction processing time (blocks)	100

Results. As seen in Figure 5.11, the base fee using the higher gas limit can reach even higher values, which may be considered an undesired behaviour. However, the high prices **fall** at a much higher rate, making the fee almost back to its typical values when the base fee with the lower gas limit reaches the highest value.

The number of transactions is very different in comparison with the lower gas limit. Most of the transactions get processed immediately after the incentive mechanism **adapts** to the large demand. Based on the result of the experiment, a higher gas limit may be a good solution to future significant peaks in the demand.

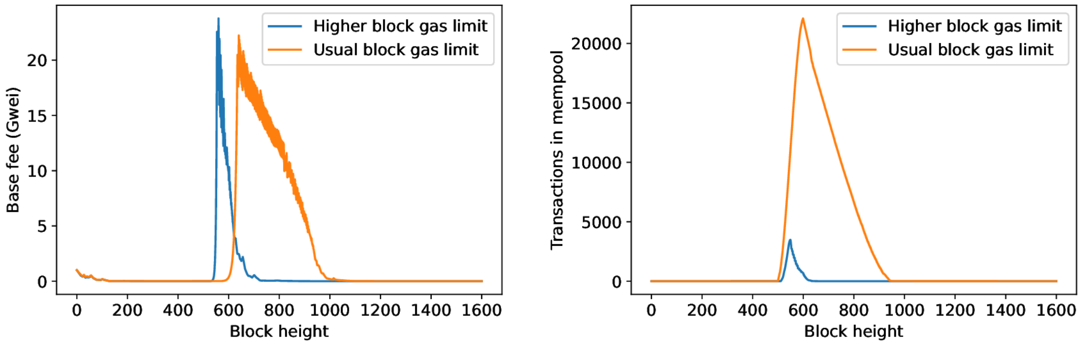

Figure 5.11: The base fee (left) and the number of transactions in the mempool (right) during the high demand scenario, comparing the results of different block gas limits.

The threat to network stability may lie in block utilization. As seen in Figure 5.12, the number of transactions in individual blocks is similar in comparison, except for several fully utilized blocks, containing four times more gas than the target. Significantly larger blocks would require more space and speed from the nodes. With less performant nodes unable to participate, the **number of nodes** may significantly decrease, making the network even slower than with a higher gas limit and low-performance nodes engaged in the network upkeep. This may be problematic during longer peaks, which could make the number of full blocks much higher in a short period of time. The effect of the change on the network should be thoroughly investigated before the implementation of this modification, particularly with an emphasis on the performance of nodes. The effects of this change are further discussed in Section 5.3.1.

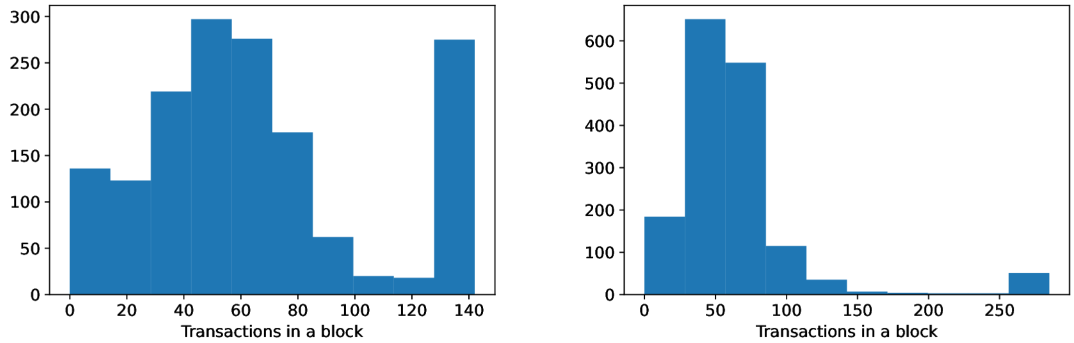

Figure 5.12: Histogram of the number of transactions per block (block utilization) during the high demand scenario, comparing the results of the usual (left) and the higher (right) gas limit.

5.2.4 Experiment V

Goal. The goal of this experiment is to investigate the influence of the base fee denominator on the network under very high demand.

Methodology. The maximal gas limit is set to the same value as in the previous experiment, meaning four times the target gas. The used denominator is now half of the initial value. The simulation run length, the demand scenario and other parameters remain the same. An overview of the used parameters can be seen in Table 5.7.

Table 5.7: Parameters used in Experiment V.

Parameter	**Value**
Target gas	1 500 000
Maximal gas limit	6 000 000
Maximal base fee change denominator	4
Initial base fee (Wei)	$1 * (10^9)$
Expected transaction processing time (blocks)	100

Results. In the comparison seen in Figure 5.13, the base fee reaches an even higher value before the price falls rapidly in the first half of the decrease. After that, the **fall** appears to slow down slightly before reaching the typical values. The base fee peak is over at almost the same time as the base fee using the higher denominator.

The difference seen in the mempool is slightly bigger, with the lower denominator mempool reaching much lower values before decreasing, following a trend similar to the base fee. It is necessary to say, that the faster increase in the base fee is supposed to discourage users from sending their transactions, making the number of transaction requests generally lower. While this may be the case when users send basic transactions, the fees may not discourage users motivated by the possibility of getting a valuable product. This behaviour **cannot be exactly predicted** and depends on the particular situation.

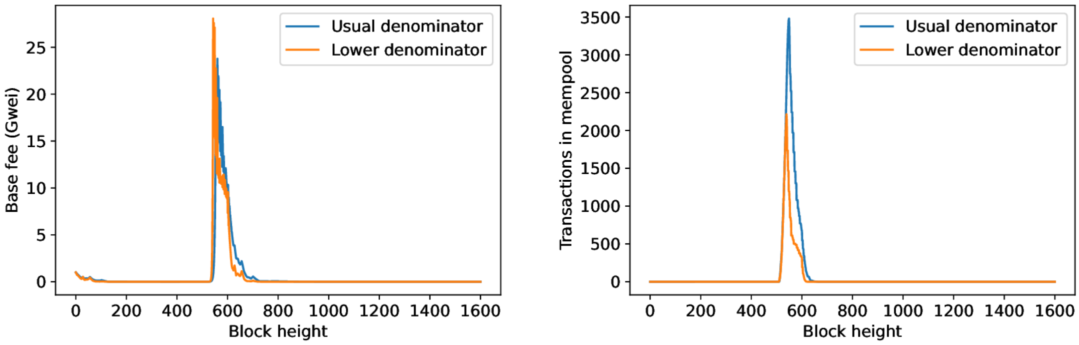

Figure 5.13: The base fee (left) and the number of transactions in the mempool (right) during the high demand scenario, comparing the results of different base fee denominator values.

As seen in Figure 5.14, the decrease in the completely full blocks is **insignificant**. Therefore, it can be assumed that a lower base fee denominator would not have a substantial positive effect on the network stability in the case of changing the gas limit to twice the value.

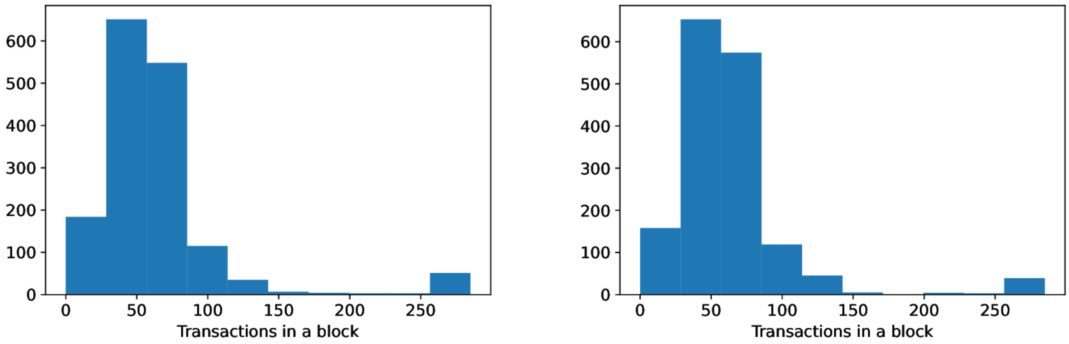

Figure 5.14: Histogram of the number of transactions per block (block utilization) during the high demand scenario, comparing the results of the usual (left) and the lower (right) base fee denominator.

5.3 Evaluation summary and discussion

Experiment I (Section 5.1.1) has **confirmed the expectations** regarding the gas price volatility and the transaction fee level, described in the economic analysis by Roughgarden (2020) [29]. The simulated results match the results of research by Liu et al. (2022) [20], who observed an easier fee estimation, but an unchanged fee level. The research is based on data collected from the mempool, the blockchain and exchanges. Experiment III (Section 5.2.2) tests the network during a high demand period, using parameters coincident with Experiment I (Section 5.1.1) and also with the currently used parameters. Based on the block utilization histogram, there is a significant number of blocks filled to the maximal gas limit possible. This phenomenon is also described in the empirical analysis by Liu et al. (2022) [20]. The same research analyses the possible effects of larger blocks on consensus security. According to them, larger blocks may lead to more forks and higher network load, leading to **possible security threats**. The results did not show significant effects using the block gas limit twice the size of the gas target.

The possibility of **increasing the block gas limit** was discussed in the EIP 1559 FAQ by Buterin [5]. According to him, the current limit is conservative and may be increased in the future, further utilizing the benefits of EIP-1559 regarding efficiency. The effects of a higher block gas limit were discussed in Experiment II (Section 5.1.2) and Experiment IV (Section 5.2.3). While the block utilization remained almost unchanged during a typical demand period. A faster https://github.com/ethereum/abm1559, but also an increase in the number of full blocks (four times the size of the gas target) was observed during the high demand period. As explained above, large blocks may negatively affect consensus security. The effect of even larger blocks on the network has not been thoroughly investigated by the papers used in this thesis and may be the subject of further research. The effect of the base fee determinant on the transaction processing has not been discussed either, but a lower denominator did not seem to have a large effect on the number of incoming transactions in Experiment V (Section 5.2.4). The possible improvements are summarized and further discussed in the following sections.

5.3.1 Higher block gas limit

According to the EIP 1559 FAQ [5], written by Vitalik Buterin, the change of the block gas limit to a higher value would increase the benefits of EIP-1559 from an efficiency perspective. He proposed to possibly modify the value after observing the effects of the new mechanism on the network. The experiments conducted in Section 5.2 indicate a positive impact of the higher gas limit on the transaction processing.

Before implementing these changes, the impact on the nodes needs to be thoroughly researched. Larger blocks may significantly increase the requirements on nodes. Based on the research on information propagation conducted by Decker and Wattenhofer (2013) [9], even slight changes in block size significantly increase the propagation delay. This change can then threaten the network stability and security and make it to occurrences like 51% attacks or selfish mining, based on the research by Göbel et al. (2015) [15]. The increase of the limit would therefore require extensive research regarding nodes and their performance, otherwise, the network security could be compromised.

5.3.2 Scaling

Most of the blockchains, including Ethereum, are monolithic and are lacking in the area of scalability. This phenomenon is called the blockchain trilemma and is further described in Figure 2.4. This weak point is the main cause of network congestion and high transaction fees. Designing future projects to utilize the scaling solutions may significantly improve the situation and the future adoption of scaling solutions is inevitable.

Currently, there are two known scaling solutions, sharding and rollups. **Rollups** are environments that process transactions, compress them, and add them to the main Ethereum blockchain. As a result, fees are cheaper and the network is used for more users and projects. There are two types of rollups at the moment, optimistic rollups and zero-knowledge rollups. **Sharding** is the process of splitting the main blockchain into several blockchains, called shards, to redistribute the security. This change should come after the transition of Ethereum to the proof-of-stake consensus mechanism. The validators are going to be divided into several groups, based on the number of shards, and each of the shards is then going to be secured by a sufficient number of validators [16].

Chapter 6

Conclusion

This thesis aims to investigate the behaviour of the new Ethereum incentive mechanism, implemented in EIP-1559 in August 2021, and propose any potential improvements in case of discovered flaws. Emphasis was placed on critical values regarding fees and block occupancy calculations.

The new mechanism was first demonstrated by simulating the network under usual demand, using historical data from mempool statistics. The results were compared to the previously used first price auction mechanism, showing a significant decrease in gas price volatility. The results were coincident with existing research. In the following experiments, the parameters were changed to see the impact of the changes in comparison with the usual, currently used parameters. It was discovered that the changes would not strongly impact the network under average demand, and the mechanism works satisfactorily with the current values.

The second part of the experiments tested the mechanism using a demand scenario with a significant peak. The specific values were inspired by the network state during the Otherside NFT mint, which caused the Ethereum network to be congested for several hours. The processing of such a demand lasted for 450 blocks in the simulation. Setting a higher gas limit showed significant improvement in the network stability during peak demand. The problem may lie in the full blocks, which are now two times larger and would increase node performance and space requirements. This change could result in fewer nodes and lower decentralization, as well as delays in block propagation, which could lead to inconsistencies or even security threats. A lower base fee denominator only insignificantly lowered the number of full blocks. Although the base fee has an essential role in stabilizing the system, a higher base fee would probably not reduce the number of transactions in this case, as the peaks are often caused by users who send time-sensitive transactions, which exceed the price of the fees if successful.

The experiments and the recent events indicated possible weaknesses in the new incentive mechanism. Based on increasing the block gas limit, a possible solution was proposed. This change would require further research before implementation, as both decentralization and security can be affected. The continuation of this thesis should consist of experiments, tests, and surveys regarding the technical node requirements.

Bibliography

[1] *Ethereum Accounts*. January 2022 [cit. 2022-01-17]. Available at: https://ethereum.org/en/developers/docs/accounts/.

[2] Andreas M. Antonopoulos, G. W. *Mastering Ethereum*. 2nd ed. O'Reilly Media, Inc., 2019. ISBN 978-1-491-97194-9.

[3] Antonopoulos, A. M. *Mastering bitcoin: programming the open blockchain*. Second editionth ed. Beijing: O'Reilly, 2017. ISBN 978-1-491-95438-6.

[4] Boom, D. V. *How Bored Ape Yacht Club Broke Ethereum*. 2022. Available at: https://www.cnet.com/personal-finance/crypto/how-bored-ape-yacht-club-broke-ethereum/.

[5] Buterin, V. *EIP 1559 FAQ*. 2021. Available at: https://notes.ethereum.org/@vbuterin/eip-1559-faq.

[6] *Consensus Mechanisms*. October 2021 [cit. 2021-11-07]. Available at: https://ethereum.org/en/developers/docs/consensus-mechanisms/.

[7] Coutts, V. *Ethereum Tokens Explained.: A Beginner's Guide to Token Standards: Everything You Need to Know*. Available at: https://medium.com/linum-labs/ethereum-tokens-explained-ffe9df918008.

[8] Dannen, C. *Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners*. Berkeley, CA: Apress, 2017. ISBN 978-1-4842-2535-6.

[9] Decker, C. and Wattenhofer, R. Information propagation in the Bitcoin network. *IEEE P2P 2013 Proceedings*. IEEE. 2013, p. 1–10. DOI: 10.1109/P2P.2013.6688704. Available at: http://ieeexplore.ieee.org/document/6688704/.

[10] *Decentralized finance (DeFi)*. December 2021 [cit. 2021-12-01]. Available at: https://ethereum.org/en/defi/.

[11] *What is Ether (ETH)?* December 2021 [cit. 2021-12-01]. Available at: https://ethereum.org/en/eth/.

[12] *What is Ethereum?* December 2021 [cit. 2021-12-01]. Available at: https://ethereum.org/en/what-is-ethereum/.

[13] *Ethereum Virtual Machine (EVM)*. December 2021 [cit. 2022-01-16]. Available at: https://ethereum.org/en/developers/docs/evm/.

[14] *Gas and Fees*. March 2022 [cit. 2022-01-16]. Available at: https://ethereum.org/en/developers/docs/gas/.

[15] Göbel, J., Keeler, H., Krzesinski, A. and Taylor, P. Bitcoin blockchain dynamics: The selfish-mine strategy in the presence of propagation delay. *Performance Evaluation*. 2016, vol. 104, p. 23–41. DOI: 10.1016/j.peva.2016.07.001. ISSN 01665316. Available at: https://linkinghub.elsevier.com/retrieve/pii/S016653161630089X.

[16] Hoffman, D. *Ultra Scalable Ethereum: Why modular blockchains are the best scaling solution for crypto*. Bankless, LLC, 2022. Available at: https://newsletter.banklesshq.com/p/ultra-scalable-ethereum.

[17] Homoliak, I., Venugopalan, S., Hum, Q., Reijsbergen, D., Schumi, R. et al. *The Security Reference Architecture for Blockchains: Towards a Standardized Model for Studying Vulnerabilities, Threats, and Defenses*. October 2019.

[18] Iredale, G. *6 Key Blockchain Features You Need To Know Now*. November 2021 [cit. 2021-11-07]. Available at: https://101blockchains.com/introduction-to-blockchain-features/.

[19] Khan, D., Jung, L. T. and Hashmani, M. A. Systematic Literature Review of Challenges in Blockchain Scalability. *Applied Sciences*. 2021, vol. 11, no. 20. DOI: 10.3390/app11209372. ISSN 2076-3417. Available at: https://www.mdpi.com/2076-3417/11/20/9372.

[20] Liu, Y., Lu, Y., Nayak, K., Zhang, F., Zhang, L. et al. Empirical Analysis of EIP-1559: Transaction Fees, Waiting Time, and Consensus Security. *ArXiv preprint arXiv:2201.05574*. 2022.

[21] Mao, W. *Modern Cryptography: Theory and Practice*. 5th edth ed. Upper Saddle River: Prentice Hall, 2004. ISBN 0-13-066943-1.

[22] Nakamoto, S. *Bitcoin: A Peer-to-Peer Electronic Cash System*. Bitcoin.org, 2008.

[23] *Decentralised peer to peer: Bitcoin's most important feature*. April 2021 [cit. 2021-11-07]. Available at: https://zerocap.com/decentralised-peer-to-peer-bitcoin/.

[24] Peringer, P. *Modelování a simulace: Studijní opora*. 2021.

[25] Polynya. *The lay of the modular blockchain land*. Available at: https://polynya.medium.com/the-lay-of-the-modular-blockchain-land-d937f7df4884.

[26] *Proof-of-stake (PoS)*. January 2022 [cit. 2022-01-17]. Available at: https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/.

[27] *ProofOfStake.com*. 2018. Available at: https://proofofstake.com/.

[28] *Proof-of-work (PoW)*. October 2021 [cit. 2021-11-07]. Available at: https://ethereum.org/en/developers/docs/consensus-mechanisms/pow/.

[29] Roughgarden, T. Transaction fee mechanism design for the Ethereum blockchain: An economic analysis of EIP-1559. *ArXiv preprint arXiv:2012.00854*. 2020.

[30] Sayeed, S., Marco Gisbert, H. and Caira, T. Smart Contract: Attacks and Protections. *IEEE Access*. 2020, vol. 8, p. 24416–24427. DOI: 10.1109/ACCESS.2020.2970495. ISSN 2169-3536. Available at: https://ieeexplore.ieee.org/document/8976179/.

[31] Standards, N. I. of and Technology. *Blockchain Technology Overview*. Internal Report 8202. Washington, D.C.: U.S. Department of Commerce, 2018.

[32] Tanenbaum, A. S. and Wetherall, D. J. *Computer Networks*. 5th ed.th ed. Upper Saddle River: Prentice Hall, 2011. ISBN 978-0-13-212695-3.

[33] *What Was The DAO?* New York: Gemini Trust Company, LLC, 2021. Available at: https://www.gemini.com/cryptopedia/the-dao-hack-makerdao.

[34] Vacca, J. R. *Computer and information security handbook*. Burlington: Morgan Kaufmann, c2009. ISBN 978-0-12-374354-1.

[35] Wang, W., Hoang, D. T., Hu, P., Xiong, Z., Niyato, D. et al. A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks. *IEEE Access*. 2019, vol. 7, p. 22328–22370. DOI: 10.1109/ACCESS.2019.2896108. ISSN 2169-3536. Available at: https://ieeexplore.ieee.org/document/8629877/.

[36] *Ethereum Whitepaper*. 2022 [cit. 2022-01-12]. Available at: https://ethereum.org/en/whitepaper/.

[37] Wood, G. Ethereum: A secure decentralised generalised transaction ledger. Berlin versionth ed. 2021.

[38] Zhao, Y. and Nayak, K. *EIP-1559 In Retrospect*. 2022. Available at: https://decentralizedthoughts.github.io/2022-03-10-eip1559/.

[39] Zou, W., Lo, D., Kochhar, P. S., Le, X.-B. D., Xia, X. et al. Smart Contract Development: Challenges and Opportunities. *IEEE Transactions on Software Engineering*. 2021-10-1, vol. 47, no. 10, p. 2084–2106. DOI: 10.1109/TSE.2019.2942301. ISSN 0098-5589. Available at: https://ieeexplore.ieee.org/document/8847638/.

Appendix A

Storage medium

```
root
├─ implementation
│  ├─ abm1559
│  ├─ data
│  │  └─ commands.txt
│  ├─ notebooks
│  │  ├─ graphs.ipynb
│  │  └─ simulation.ipynb
│  ├─ results
│  │  ├─ experiments
│  │  └─ simulation
│  ├─ scripts
│  │  ├─ mempoolstats.py
│  │  └─ txnsfees.py
│  └─ requirements.txt
├─ latex
└─ bp-xburia28.pdf
```

The folder **implementation** contains the following data:

- **abm1559** containing the simulation environment
- **data** containing all data sets used in the experiments and commands to download them

- **notebooks** containing the Jupyter Notebooks for the simulation and the graph plots
- **results** containing all of the outputs from scripts
- **experiments** contains the plots used in this thesis
- **simulation** contains the simulation run outputs
- **scripts** contains the scripts used to process the data sets
- **requirement.txt** contains all of the requirements for the abm1559 environment

The folder **latex** contains the LaTeXsource codes of the thesis. The file **bp-xburia28.pdf** is the thesis in PDF format.

The Python version used during the implementation of the thesis is **Python 3.8.5**. To install all requirements for the abm1559 environment, run the following command in the **implementation** folder:

pip install -r requirements.txt

To get data from the Ethereum blockchain, use the following command to install the Ethereum ETL tool:

pip3 install ethereum-etl

Then use the commands written in **commands.txt** in the data folder to download the data in CSV format.