Alt-PoW: An Alternative Proof-of-Work Mechanism

Thesis · May 2018

CITATIONS
0

READS
136

2 authors:

Sarah Sharkey
Trinity College Dublin
2 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Hitesh Tewari
Trinity College Dublin
33 PUBLICATIONS **435** CITATIONS
SEE PROFILE

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Alt-PoW: An Alternative Proof-of-Work Mechanism

Sarah Sharkey
B.A. (Mod.) Computer Science
Final Year Project May 2018
Supervisor: Dr. Hitesh Tewari

School of Computer Science and Statistics

O'Reilly Institute, Trinity College, Dublin 2, Ireland

Declaration

I hereby declare that this project is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

Sarah Sharkey

Date

Permission to lend

I agree that the Library and other agents of the College may lend or copy this report upon request.

Sarah Sharkey

Date

Abstract

In 2009, Satoshi Nakamoto launched the world's first decentralised digital currency, Bitcoin. The launch of such a system brought on a technology revolution of blockchain, the core component behind the currency. Since then, the popularity of Bitcoin has soared as well as many alternative cryptocurrencies, yet the underlying mechanism of Bitcoin is far from quintessential. It can be so costly that a transaction fee can be more than the value of the transaction itself, so inefficient that the energy consumed per year can be more than that of a country, and so slow that a transaction could take longer to confirm than the time for the moon to move its entire diameter. Bitcoin has indeed been revolutionary, but there is opportunity for a faster and more economical protocol to improve on its shortcomings.

In this report, a new distributed consensus mechanism is presented that gives participants in the network the opportunity to make informed decisions and incentivises them to be shrewd with their resources. As opposed to Bitcoin's single blockchain, multiple chains are introduced to parallelise and speed up transaction verification, and to improve the diversity of who is verifying transactions. These advancements are coupled with maintaining a secure and robust protocol.

Acknowledgements

First and foremost, I would like to thank my supervisor, Dr. Hitesh Tewari, for his continued guidance, knowledge and imagination throughout this project, for his commitment and for encouraging me to be creative whilst delving into the world of blockchain.

I would like to thank my family for the opportunities I have been given and for their boundless encouragement over the years.

I would also like to thank Mark for always being a source of help and unending support throughout the last four years.

I am thankful for all my peers and the faculty members of Trinity College for challenging me and for all they have taught me.

Contents

1 Introduction **1**

1.1 Report Structure 2

2 Background & Objectives **3**

2.1 Blockchain 3

2.1.1 Cryptocurrencies 4

2.2 Consensus Mechanisms 4

2.2.1 Proof-of-Work 4

2.2.2 Proof-of-Stake 5

2.2.3 Directed Acyclic Graphs: The Tangle 5

2.3 State of the Art: Bitcoin 6

2.3.1 Transactions, Blocks & The Chain 6

2.3.2 Mining 8

2.3.3 Block Rate as a Poisson Process 10

2.3.4 Security Considerations 11

2.3.5 Scalability 12

2.3.6 Energy Consumption 13

2.4 Objectives 13

2.4.1 Tactical Mining 14

2.4.2 Increased Block Rate 14

2.4.3 Reduced Energy Consumption per Block 14

2.4.4 Increased Mining Diversity 14
2.4.5 Maintaining Security 15

3 Alt-PoW: An Alternative Proof-of-Work Mechanism 16
3.1 The Concept 16
3.2 Protocol Details 17
3.2.1 Variables 17
3.2.2 Round Blocks 17
3.2.3 Multiple Chains 19
3.2.4 Difficulty of Blocks 20
3.2.5 Number of Rounds 22
3.2.6 Partitioning Transactions Across Chains 23
3.2.7 Transaction Dependencies 23
3.2.8 Tactical Mining 25

4 Evaluation 27
4.1 Forking 27
4.2 Target Block Time 29
4.3 Rate of Block Creation 31
4.3.1 Future Block Rate 38
4.4 Confirmation Time 39
4.5 Energy Consumption per Block 39
4.6 Mining Diversity 41
4.7 Inter-node Communication 41

5 Potential Attacks 42
5.1 Double Spending 42
5.2 Alternative History Attack 42
5.3 51% Attack 43
5.4 Concealing Round Blocks 43

6 Future Work **45**

6.1 Alt-PoW as a Markov Chain 45

6.2 Ideal Values for k, j and x 45

6.3 Typical Behaviour of Miners 46

6.4 Increased Peer-to-Peer Traffic 46

6.5 The Rate at which Mining Diversity Increased 47

6.6 Security 47

7 Conclusion **48**

A Source Code **53**

List of Figures

2.1 Bitcoin Transactions 6
2.2 Bitcoin Blockchain 7
2.3 Bitcoin Mining Distribution 9

3.1 One Round vs. Six Rounds 18
3.2 *K* Multiple Chains 19
3.3 *K* Chains With Dependencies 20
3.4 *K* Chains With Dependencies ($k = 4$) 21
3.5 Block Dependencies ($k = 10$) 24
3.6 Number of Block Dependencies ($k = 10$) 25
3.7 Mining Probability ($j = 6$) 26

4.1 Probability of Forking in Final Round of Alt-PoW Based on Remaining Hashing Power 30
4.2 Expected Arrival of Blocks ($k = 5$) 33
4.3 Expected Arrival of Blocks ($k = 8$) 33
4.4 Expected Arrival of Blocks ($k = 10$) 34
4.5 Expected Arrival of Blocks ($k = 5$) 35
4.6 Expected Arrival of Blocks ($k = 8$) 36
4.7 Expected Arrival of Blocks ($k = 10$) 37
4.8 Expected Arrival of Blocks vs. Number of Block Dependencies ($k = 10$) . 38

4.9 Energy Consumption of Alt-PoW as a Fraction of PoW's Energy Consumption ($x = 2.5$, $k = 10$) 40

Chapter 1

Introduction

> Well, if the rules of the game force a bad strategy,
> maybe we shouldn't try to change strategies.
> Maybe we should try to change the game.
>
> Brian Christian, Tom Griffiths

Bitcoin miners compete in a game for profit. Their arsenal is specialised hardware and their ammunition is hashing power as they contend to obtain coins of the world's most valuable cryptocurrency. Despite how well-prepared they may be for the race, the only strategy put before them is mindless computation without any opportunity for tactics or craft, due to the consensus mechanism in place. It is a winner-takes-all protocol, so the resources exhausted by non-successful participants have gone for no benefit. The protocol is slow, costly and has only a small number of dominant parties who frequently succeed.

The aim of this project is to create an alternative consensus mechanism that allows for strategic mining and increased mining diversity, resulting in a faster and more energy efficient protocol, whilst maintaining security of the network.

1.1 Report Structure

The report is structured as follows:

- **Chapter 2** aims to give the user the necessary background information needed in distributed ledger technology and consensus mechanisms, as these concepts will be used throughout the project. It also gives an overview of a current state of the art cryptocurrency, Bitcoin, highlighting its strengths and flaws, and from this the objectives of this project will be established.

- **Chapter 3** outlines the design of Alt-PoW based off the objectives described in chapter 2. It illustrates the evolution of the design process, how certain values are calculated and how the system overcomes numerous obstacles.

- **Chapter 4** evaluates the performance of Alt-PoW in comparison to Bitcoin's PoW, using different metrics such as speed, energy consumption and mining diversity.

- **Chapter 5** considers possible attacks on the system and how secure the system is in relation to these attacks.

- **Chapter 6** sketches out areas for future work based on the evaluation of Alt-PoW in chapter 4 and its security in chapter 5.

- **Chapter 7** summarises the motivation for the project and the objectives it achieved.

Chapter 2

Background & Objectives

2.1 Blockchain

A blockchain is a continuously growing ledger of an entire history of transactions. Transactions are collated into blocks; each block contains a body of transactions and a header of fields such as a timestamp and a hash of all the transactions in the block. Each block is cryptographically linked to the previous block and this cryptographic link is achieved by each block header containing the hash of the previous block's header [1].

A blockchain is immutable as any change to a field in the header or to a transaction in a block would alter the block's hash. The next block in the chain would now contain an invalid hash of the previous block, and so the cryptographic link is broken. Rewriting history in a blockchain would mean having to rewrite all blocks since that point in history, therefore, the security of a block is reassured as more blocks are added to the chain [1].

A blockchain is distributed; the network is made up of many nodes who continuously communicate with each other on a peer-to-peer basis. Nodes update each other with new transactions, new blocks and the current state of the chain [1].

2.1.1 Cryptocurrencies

One of many applications of blockchain technology is digital currencies. Cryptocurrencies provide digital monetary and payment systems that are an open and self-regulating alternative to central authorities that manage traditional currencies. Transactions represent the transfer of monetary value from one entity to another, and nodes in the network work to verify each other's transactions [2].

2.2 Consensus Mechanisms

Consensus mechanisms are essential in distributed computing for consistency and liveness in the network [3]. With every node containing a copy of the blockchain, there needs to be a mechanism in place so that each copy is consistent. Consensus mechanisms represent majority decision making in a network of many nodes, and provide an incentive for nodes to support the network and encourage nodes to be honest [1].

2.2.1 Proof-of-Work

Proof-of-Work (PoW) is a protocol where a prover must demonstrate to a verifier that they have exhausted a required amount of computational resources [4]. It has proved useful in deterring denial-of-service attacks as well as other security objectives [4, 5]. PoW as a consensus mechanism for a cryptocurrency involves participants in the network having to solve a moderately hard problem in order to be able to add a block of transactions to the chain, ergo, confirming the transactions [4]. The problem should be easily verifiable but difficult to solve [5]; the process of trying to solve the problem is called *mining* and the participants who take part are known as *miners* [4]. Miners who successfully add a block to the chain are rewarded in coins, which is the incentive for them to contribute to verifying transactions in the network [1]. However, due to the nature of this protocol, it is very reliant on energy consumption and therefore, introduces significant cost overhead [6].

2.2.2 Proof-of-Stake

Proof-of-stake (PoS) is another consensus mechanism, except the participant who is allowed to add a block to the chain is chosen based on their stake in the network as opposed to the amount of CPU effort they are willing to put in. A participant's stake in the system is dependent on how many coins they are willing to stake and/or how long they have held the coins. A participant is randomly chosen each block, and the more stake a party has the more likely they will be chosen. The elected party can issue a block and is rewarded for doing so. Contrary to PoW, no significant amount of energy is used [6, 7].

2.2.3 Directed Acyclic Graphs: The Tangle

An alternative approach to consensus is using a directed acyclic graph (DAG) as opposed to a blockchain. An example of such a system is the *tangle* [8]. The tangle is a DAG of transactions. When a new transaction arrives, it must approve k previous transactions, and these approvals are represented by directed edges. A directed path between transactions A and B implies A indirectly approves B. Therefore, the idea behind the tangle is that users who issue transactions are contributing to the network's security, and the more confirmations a transaction has the more it is accepted by the system with a higher level of confidence. Honest nodes will only be inclined to approve valid transactions; if node A approved an invalid transaction, that would deter honest nodes from approving node A's transaction as that would be indirectly approving an invalid transaction also. This mechanism also does not consume a significant amount of energy, it is quick and nodes can contribute to the security of the system without having to exhaust resources or prove their stake in the system.

Some other emerging DAG-based systems are Byteball [9], and IoT Chain [10] which can handle over 10,000 transactions per second.

2.3 State of the Art: Bitcoin

Bitcoin is the first decentralised cryptocurrency and the most valuable [11]. It was first introduced in 2008 by Satoshi Nakamoto [1], and has since then been considered a technical breakthrough as a completely decentralised payment system using blockchain. Bitcoin uses PoW as its consensus mechanism; its functionality is outlined below.

2.3.1 Transactions, Blocks & The Chain

Transactions

A transaction represents the transfer of bitcoins (BTC) between entities in the network, and the blockchain contains an entire public, immutable history of all transactions. Each transaction has a transaction ID (txid) which is the SHA-256 hash of the transaction. Transactions contain inputs which refer to one or more previous txids; these txids reference the transaction(s) where the coins to be sent are redeemed. Coins from a particular transaction can only be redeemed once[1] to avoid double spending and can only be redeemed by the entity who received them. The outputs of a transaction refer to the one or more entities that are to receive the coins; the sum of the values of the inputs must be greater than or equal to the sum of the outputs or else the transaction is considered invalid. When the value of the inputs exceeds the value of the outputs, the remaining coins are considered transaction fees and are collected by the miner of the block as an additional 'tip' [11].

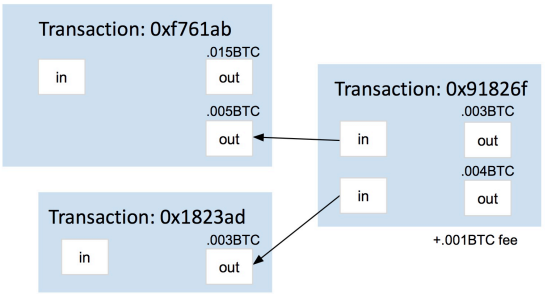

Figure 2.1: Bitcoin Transactions

[1] The specifics of how this is implemented are not important here.

Blocks

A block contains a header and a body of transactions; the hash of the block is the header hashed with SHA-256 [1]. The block header is 80 bytes and contains fields such as:

- Previous block hash
- Timestamp
- Merkle root: a hash of all transactions in the block
- Nonce: any integer value
- Difficulty: the required number of leading zero bits in this block's hash, e.g. 0x00000000000000000057fcc708cf0130d95e27c5819203e9f967ac56e4df598ee is an example target value

Bitcoin currently has a maximum block size of 1 MB [12], therefore, there is a limit on the number of transactions that can be included in a block.

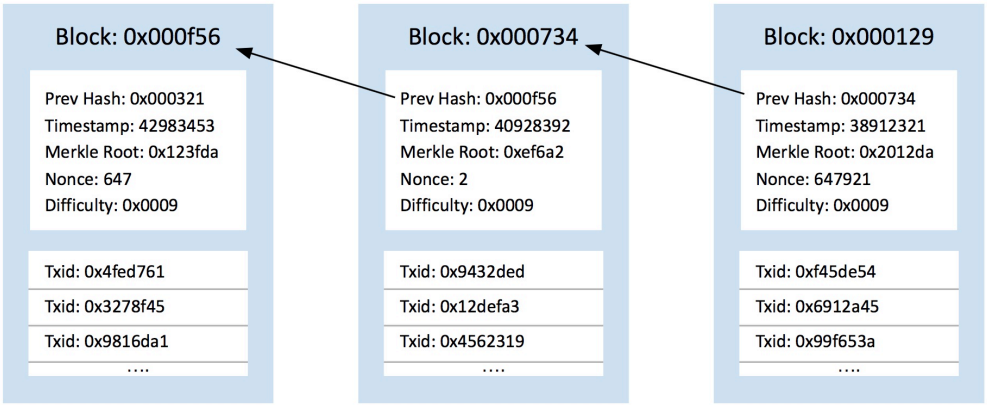

Figure 2.2: Bitcoin Blockchain

The first transaction in a block is called the *coinbase* transaction; it is a special transaction that transfers transaction fees and the block reward[2] to the miner of the block [1, 11].

[2]The block reward refers to additional coin rewards miners receive for mining a block. These are new bitcoins that are created to distribute more coins into the network.

2.3.2 Mining

Mining bitcoin is the process of collating unconfirmed transactions into a block, and finding a value for the nonce in the block header such that the block's hash has the required number of leading zeros so that the block can be added to the chain. Once a suitable hash is found, the miner submits the block to the network where the nodes verify if the block is valid; verification is a trivial operation compared to mining the solution. If the block is accepted by the network, it is added to the chain and the transactions are confirmed [1].

Mining Approach

The only approach available to miners is a brute-force search, testing different values for the nonce until they find a suitable hash. There is no such thing as being "1% towards solving a block"; a miner's chances of finding a solution are the same at any point as when they first started [13]. Therefore, a miner has no sense of progress when mining. They do not know if they are 30 seconds from finding a solution or 30 minutes, and they do not know how the rest of the network is faring either. Miners do not know whether it is in their interest to exhaust their CPU resources for a particular block, because they do not know, based on their current strategy of trying different nonces, what their chances are of mining the particular block.

Forks

Occasionally it can occur that two miners find and submit two different versions of the next block around the same time, this is causes a *fork* in the chain [14]. Nodes will always converge to the longest branch of the fork as the probability of the shorter chain catching with the longer chain becomes exponentially smaller the more it falls behind [1]. When both branches are of the same length, miners can work on either branch [14], however, they tend to work on the branch they received first but save the other branch in case it becomes longer [1]. Once a branch becomes longer, the other branch is considered orphaned and the fork is naturally resolved. Until the fork is resolved, blockchain replicas across the network

are in an inconsistent state.

Mining Distribution

As shown in Figure 2.3, the hashrate of the current Bitcoin network is unbalanced. The top three mining pools are based in China at the hand of cheaper electricity [15], and combined they have over 51% of the hashing power in the network. These three mining pools, therefore, have the potential to collude and influence mining in their favour (see section 2.3.4).

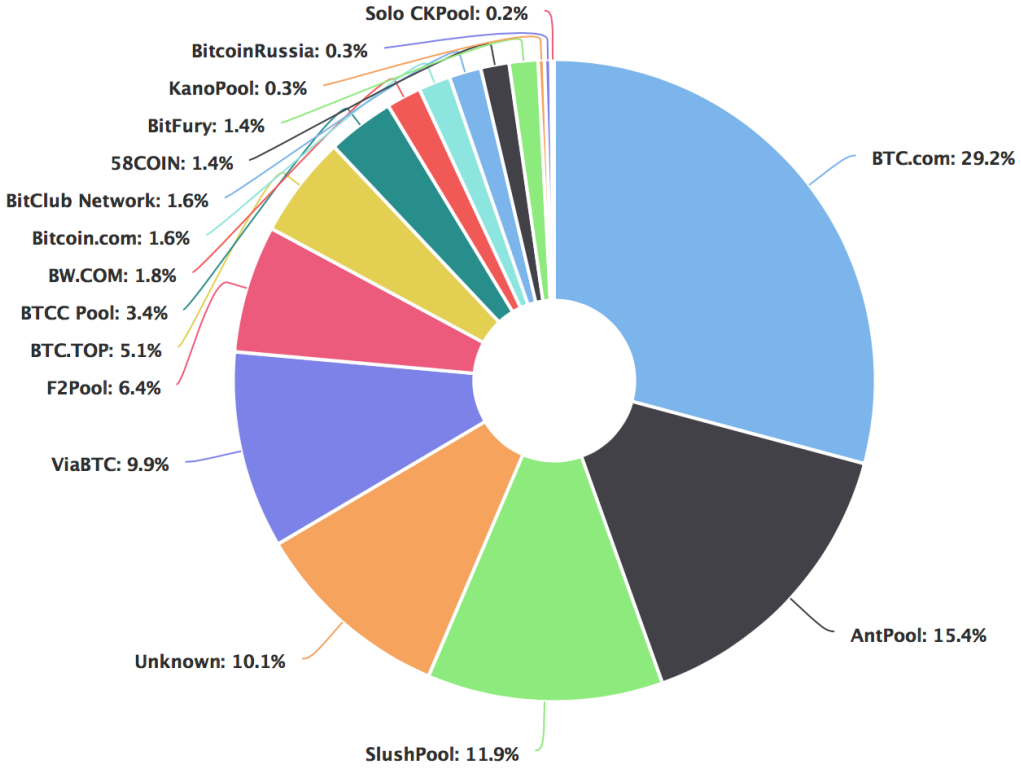

Figure 2.3: Bitcoin Mining Distribution[3]

[3]Blockchain. ”Hashrate Distribution.” 2018. Accessed April 23. https:// blockchain.info/pools.

Target Block Time

The difficulty of the PoW problem is adjusted every 2016[4] blocks so that the network maintains an average block rate of one block being added to the chain every 10 minutes [16]. 10 minutes was chosen as a compromise between first confirmation time and the amount of work wasted due to forks [13]. If the confirmation time were to be reduced, the risk of chain splits would increase as different nodes would receive different blocks first due to propagation time [16].

2.3.3 Block Rate as a Poisson Process

A Poisson process is a counting process that models occurrences of events that happen at an average rate, yet each occurrence is independent of the previous occurrences. The time between occurrences is described by an exponential distribution. As a result of periodically adjusting the difficulty, block finding is a Poisson process with a rate parameter $\lambda = \frac{1}{10}$; the time between block arrivals follows an exponential distribution with average rate of one block every 10 minutes [16].

A key property of an exponential distribution is that it is memoryless; regardless of how long miners have been mining a particular block, the expected time until a block is found is 10 minutes and the probability of finding a block remains constant [17].

Merging Poisson Processes

Independent Poisson processes of rates λ_1 and λ_2, respectively, can be merged into one Poisson process of rate $\lambda_1 + \lambda_2$ [18].

Splitting Poisson Processes

A Poisson process can be split into multiple Poisson processes using a deterministic function of the process. For example, the arrival of vehicles at an intersection is a Poisson

[4]There is no formal reason why 2016 blocks was chosen for Bitcoin, other than it provided a sufficient sampling size.

process of 100 cars per minute, 30% of the vehicles are cars and 70% of the vehicles are trucks. This Poisson process can be divided into two smaller Poisson processes: a Poisson process of car arrivals with an average rate of 30 cars per minute and a Poisson process of truck arrivals with an average rate of 70 trucks per minute [19].

2.3.4 Security Considerations

Alternative History Attack

An alternative history attack occurs when a miner attempts to generate an alternative chain faster than the honest chain to put the system in a different state, e.g. to revert a transaction the attacker previously published. The attacker must work against the rest of the network to beat the honest chain, therefore, would need extensive computational resources and also luck in their favour. The probability an attacker even catches up with the honest chain becomes exponentially smaller the further behind the alternative chain is [1].

$$q_z = \begin{cases} 1, & \text{if } p \leq q \\ (q/p)^z, & \text{if } p > q \end{cases}$$

where:

p = probability an honest node finds the next block
q = probability the attacker finds the next block
q_z = probability the attacker will ever catch up from *z* blocks behind

If unsuccessful, alternative history attacks are costly to the attacker due to the resources exhausted.

Confirmation Time

Due to the risk of a longer alternative chain being created, a block that was added to the chain is at risk of becoming an orphaned branch of a fork. Therefore, a block is considered securely confirmed after six blocks have been mined as opposed to one, which takes on

average after 60 minutes. This figure was chosen so that if an attacker were to obtain 10% of the network's hashing power, the probability of them catching up with the main chain from six blocks behind would be only 0.02428% [1].

51% Attack

The security of Bitcoin is based off the assumption that the majority of the hashing power in the network belongs to honest miners. The system is secure as long as one group of attackers do not collectively hold more than 50% of the network's hashing power, or else this group have enough CPU power to rewrite history with an alternative chain [1].

Sybil Attack

A Sybil attack is an attack where it is unknown to the network that a single entity is controlling multiple nodes in the network, spawning different virtual machines and IP addresses. The attacker pretends to be multiple distinct users and uses this to their advantage. However, Sybil attacks are hindered in Bitcoin by the use of PoW as its consensus mechanism. Attackers can only generate blocks at a rate proportional to their overall computational power which cannot be forged [20].

2.3.5 Scalability

Due to the 1 MB block size limit and a block being added to the chain on average every 10 minutes, there is a an upper bound on the number of transactions per second (TPS) the network can handle. This is estimated to be an average of 7 TPS, in comparison to PayPal's 500 TPS and VISA's 4000 TPS [12].

Transaction Fees

As Bitcoin becomes more popular and is adopted by more users, more transactions are being submitted to the network awaiting confirmation. The limitation on TPS results in a competitive market where only the transactions with the highest transaction fees will be

included in blocks by the miners. In [21] it can be seen that transaction fees are integral to the security of Bitcoin as in time they will be the only incentive for miners to continue to mine[5]. If transaction fees were to be reduced to a certain level, less miners would be incentivised to mine, thus increasing the probability that one of the remaining miners contains a large proportion of the outstanding network hashing power. However, with a fixed block size the competition can result in exceptionally high transaction fees that are not justified by an insufficient security level.

2.3.6 Energy Consumption

Due to the nature of the PoW mechanism, it demands a large energy consumption from the network for each block mined. The evolution of Bitcoin mining hardware from CPUs to ASICs (Application Specific Integrated Circuits) has resulted in more energy efficient hardware [15]. However, the energy consumption for a single Bitcoin transaction in December 2017 amounted to 259 KWh, which is more than one US household's weekly energy consumption [22].

2.4 Objectives

PoW has forced a single strategy on miners, a brute-force approach where they have no sense of progress and cannot tactically decide whether it is in their interest to exhaust their CPU resources for a particular block. As a result of this, the entire network's computational resources are squandered on a block where only one miner is rewarded for their work. With all miners in the network mining the same block, the target time for a block has been set to 10 minutes to reduce the risk of a fork occurring, as well as a one hour confirmation time, which in turn has led to a low transaction rate. Due to disparate electricity costs in different regions, there are a few very powerful mining pools that dominate the bitcoin network.

[5]The last bitcoin to be mined will be on May 7th 2140, leaving a finite 21 million bitcoins in the network [13]. From that point forwards, the only incentive for miners to mine will be transaction fees.

Bitcoin has revolutionised digital currencies using blockchain, however, the goal of this project is to improve on its fallbacks.

2.4.1 Tactical Mining

Miners are currently mindlessly mining as it is the only strategy that is available to them. If miners were able to observe their progress and analyse their chances in mining a particular block, they could be more tactical and savvy about their approach.

2.4.2 Increased Block Rate

If cryptocurrencies are going to be more universally adopted, an increased block rate is needed to handle the higher volume of transactions in the network and a shorter block confirmation time is essential. An increased block rate also reduces competition between transactions in the network to be included in a block which in turn will hinder unnecessarily high transaction fees.

2.4.3 Reduced Energy Consumption per Block

The energy consumption used to mine one block in Bitcoin is the entire network of miners exhausting their resources for on average 10 minutes; all miners but the one who successfully mined the block wasted their energy for no gain. A more economic energy consumption would be more sustainable for long-term adoption of a cryptocurrency.

2.4.4 Increased Mining Diversity

With the possibility of only three of the current top mining pools being able to collude and launch a 51% attack (section 2.3.4), increased mining diversity is essential to the security of the system.

2.4.5 Maintaining Security

Many of the drawbacks of Bitcoin's current PoW mechanism are a compromise for the system's security. An objective of this project is to maintain the security of the system whilst improving on these drawbacks.

Chapter 3

Alt-PoW: An Alternative Proof-of-Work Mechanism

3.1 The Concept

Alt-PoW introduces the idea of progress in mining. It replaces the one large problem that miners must, essentially blindly, solve for a block, with a sequence of smaller problems. This changes the game of mining from one round to multiple intermediate rounds to mine a block. If each miner knows what round they are in, and what rounds the rest of the network are in, they can calculate their likelihood of mining the block before the others. If, for example, there are 10 rounds, miner A is in round 1 and miner B is in round 8, then miner A can decide that it is not in their benefit to exhaust their computational resources on a block that they have a very small chance of succeeding with. Naturally, as the competition to mine a block proceeds, miners who are not faring well in the process will be savvy with their resources and start to drop out, reducing the energy consumption for this particular block.

This design begs the question, "Are miners who have dropped out just supposed to wait for the next block to start mining again?" Stemming from this, the design extends further to multiple chains where multiple blocks can be mined in parallel. There are various different

mining competitions that miners can enter into, and if they are failing to advance on one chain, they can drop out and decide to mine on a different chain where their chances are more prosperous.

With this design, miners can be more shrewd about how they use their resources and not have to wait around. They can make intelligent decisions about where to put their resources and know when it is time to stop mining a particular block or when to commit to it. The details of the protocol are outlined in the next section.

3.2 Protocol Details

3.2.1 Variables

Certain variables will be commonly used throughout the report when describing Alt-PoW:

$k =$ the number of sub-chains in the system
$j =$ the number of rounds per transaction block
$x =$ the target time (minutes) for a transaction block to be mined

3.2.2 Round Blocks

There must be complete transparency in the network as to which miner is in what round and progress cannot be forged. This can be implemented with blockchain; each miner has their own sub-chain that spawns off the last block in the main chain. Each block in a sub-chain represents a round the miner has completed; once a miner is in the last round, they can mine the block of transactions on top of their sub-chain. For example, if it is a 6-round system as in Figure 3.1b, a miner would have to mine 5 intermediate blocks before mining the actual block with transactions and rewards. If a miner decides to drop out of mining a particular block, their sub-chain is abandoned and other miners no longer need to keep a copy of this sub-chain.

Each block in the intermediate chain is called a *round block*, and blocks with transactions and rewards are called *transaction blocks*. Round blocks do not contain any real

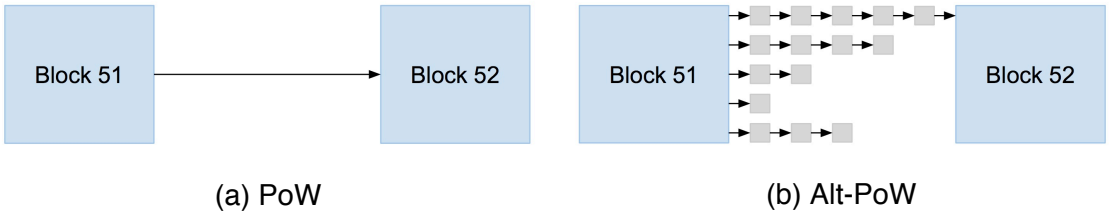

Figure 3.1: One Round vs. Six Rounds

transactions from the network and there is no reward for mining them, they are merely there as a representation of advancement in the process of mining the transaction block.

For convenience, when iterating through the chain, each transaction block contains the hash of the previous transaction block as well as the hash of the last round block.

0 BTC Coinbase Transaction

Round blocks contain one single transaction, a 0 BTC coinbase transaction. This transaction is simply sending 0 BTC to the miner who mined the round block. The purpose of this transaction is so the miner can identify that round block as their own. Any block mined on top of that round block must have a coinbase transaction sending to the same miner; if the next block is a round block then a 0 BTC coinbase transaction or a real coinbase transaction if the subsequent block is transaction block. This prevents a miner from mining on top of another miner's sub-chain as the coinbase transactions will differ.

Incentive to Publish Rounds

Miners are not obligated to publish their round blocks immediately, but it is in their interest to do so. Honest miners would be oblivious to the progression made by the miners withholding their round blocks, so the honest miners believe their chances of mining are better than they actually are, encouraging them continue mining longer than they typically would have. This only introduces more competition in mining the block, and more competition implies a reduced chance of succeeding for all parties, including the miners withholding

their round blocks. Therefore, miners are incentivised to declare their position in the rounds to deter more competitors.

3.2.3 Multiple Chains

To preserve a busy network, Alt-PoW has multiple chains spawning from the same genesis block. When a miner drops out of mining a block on one chain they have the opportunity to mine on another chain where their chances may be better.

Block 0 and the first block of each chain are all considered genesis blocks and are created when launching the system. From then on, block creation is left to the miners.

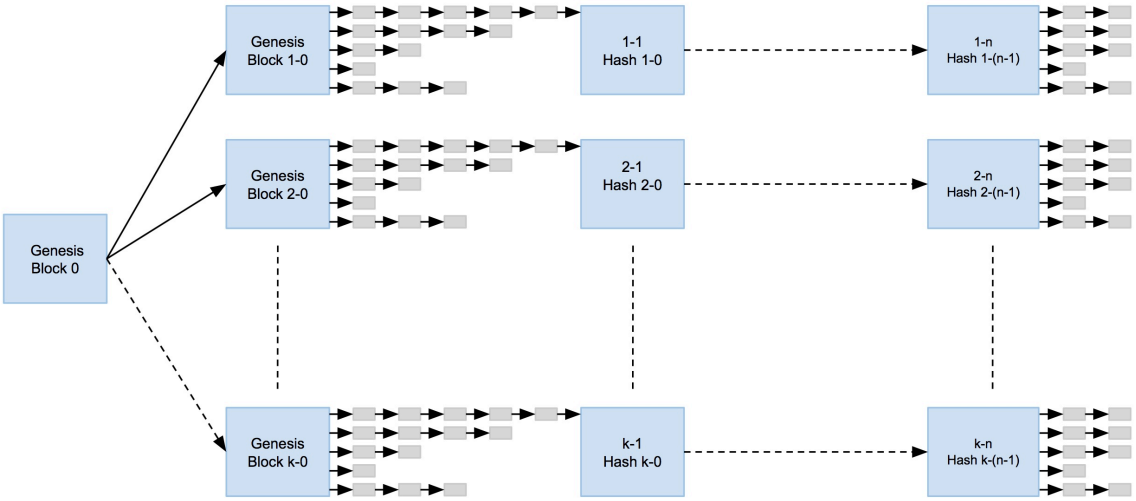

Figure 3.2: K[1] Multiple Chains

Inter-Chain Dependencies

There are a few concerns when it comes to dividing the network across multiple chains:

- As there would be less hashing power on a particular chain, it would be easier for a miner to dominate a chain and for a 51% attack to occur.

[1] Where k is the number of sub-chains.

- Nothing is stopping one chain from going much further ahead than the others, leaving the system uneven.
- If there are dependent transactions on different chains, it needs to be known that the dependent block came after the dependee. The only way to deduce which block came first would be by their timestamps, but using this method of determining the order of blocks would not be 100% accurate[2]

A solution to resolve the above concerns is inter-chain dependencies. A block not only contains the hash of the last block on its chain, but also a hash of a block on the next chain. As in Figure 3.3, block 1-1 contains a hash of block 1-0 and 2-0, block 2-1 contains a hash of 2-0 and 3-0 and so on, until the kth chain where block k-1 contains a hash of k-0 and 1-1.

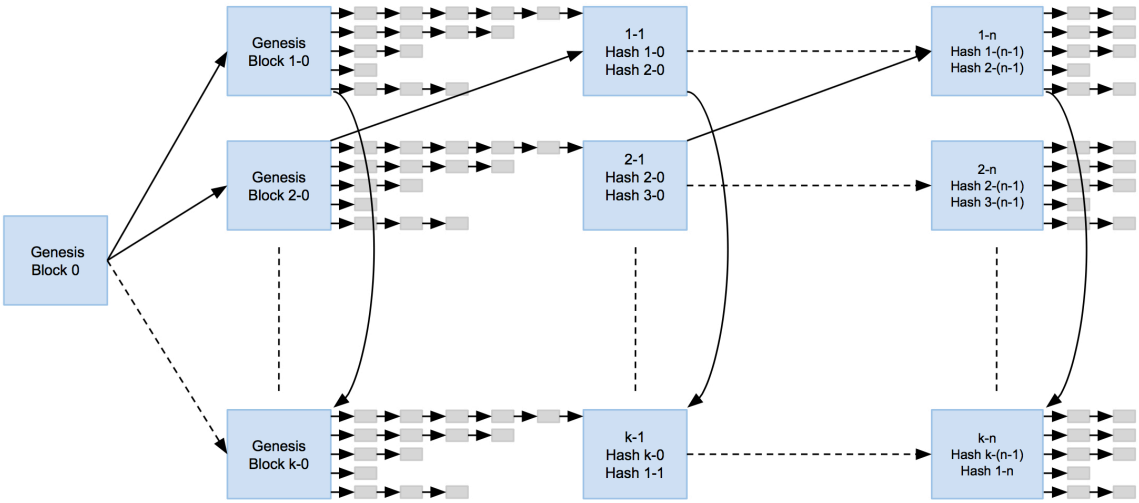

Figure 3.3: *K* Chains With Dependencies

3.2.4 Difficulty of Blocks

Block difficulty can be adjusted with the same approach as in PoW, which is adjusting the difficulty every 2016 blocks based on how long the last 2016 blocks took to mine (both

[2]Block timestamps are not exact and are only accurate to an hour or two [13].

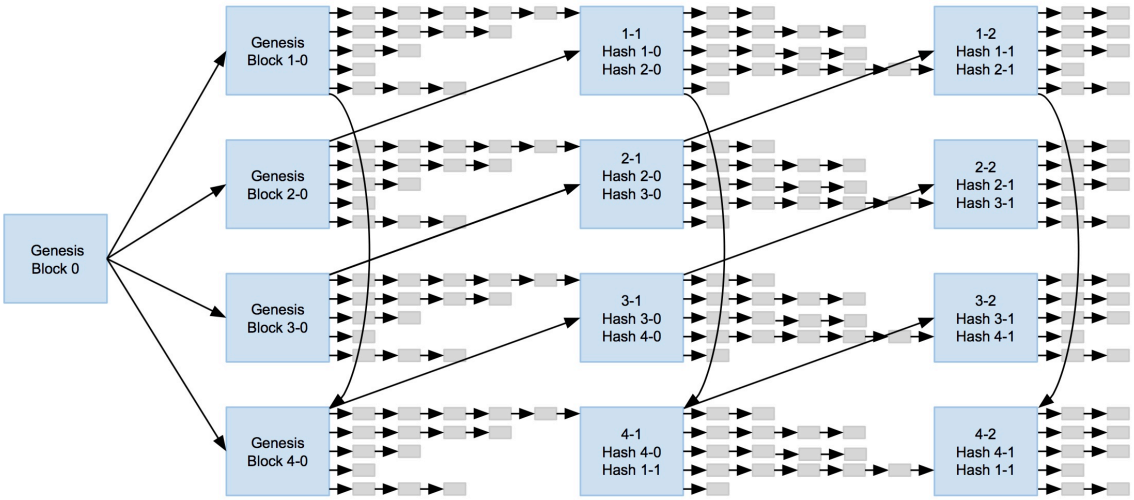

Figure 3.4: *K* Chains With Dependencies ($k = 4$)

round blocks and transaction blocks included). The exact state of the Alt-PoW system after 2016 blocks have elapsed cannot be predicted; some chains may contain more blocks than others and visa versa. Because of this, adjusting the difficulty of the system as a whole brings its complexities as it is not obvious what the last 2016 blocks were. Therefore, for simplicity, block difficulty is adjusted on each chain separately.

The difficulty of blocks on each chain can be adjusted to maintain an average block rate with the same method as in PoW. For example, if the system aims to have a new block of transactions mined on average every 10 minutes and there are 10 rounds, the difficulty of blocks would be periodically adjusted so that one block arrives on average every minute.

Given that there are dependencies between chains, the difficulty is not adjusted solely based on the timestamps of the blocks in one chain. Consider the time taken for the first round block to be mined. The time taken is dependent on when miners are able to start mining that block. To mine the block, miners need both the hash of the previous block in the same chain and the hash of the block in the next chain. If the hash of the block in the next chain is not available, as it has not been mined yet, then mining cannot begin on the block in this chain. Therefore, the wait for the block in the next chain to be mined is not to

be factored in our method of adjusting the difficulty of the blocks, as it may seem that the first round was very difficult for miners to mine when really the starting gun had simply not been fired yet.

So as a result of this, when calculating how long the last 2016 blocks took to mine on a chain, the time spent waiting for dependent blocks on other chains to arrive is not to be factored in. To calculate how long miners took to mine the block, the time is measured from when all of its dependencies had arrived. Most blocks only have one dependency which is the previous block on the same chain, however, the first round block also has a dependency on the next chain. The time taken for a block to be mined from the point in time all its dependencies had been mined is given in the following equation:

$$t_a = T_a - max(T_{a_1}, ..., T_{a_n}) \tag{3.1}$$

where:

t_a = time for block a to be mined
T_a = timestamp for block a
T_{a_x} = timestamp of block a's xth dependency

Due to the averaging process, block arrivals on a subchain follow on exponential distribution[3].

3.2.5 Number of Rounds

A goal of Alt-PoW is to keep the network nodes busy mining and tactical about where they mine. Consider a miner that tries but fails to mine on each sub-chain. If, by the time they get back to the first chain they tried, the block they dropped out of mining is still being mined, they are left waiting until the next block is mined so they can mine a new block. It would not be in their interest to join back in mining a block they already dropped out of because, as time goes on and other miners have advanced, their chances of mining are worse than when they decided to drop out. Therefore, it is desired that miners are kept

[3]From the point all block dependencies had been mined

busy. If a miner goes even as far as the first round in each chain, by the time they get back to the first chain they tried, the block they dropped out of should have been mined. Using this logic, the number of rounds should equal the number of sub-chains.

$$x = k * \frac{x}{j} \tag{3.2}$$

where:

$x =$ expected time for a transaction block to be mined

$k =$ number of sub-chains

$j =$ number of rounds

$\frac{x}{j} =$ expected time for a round block to be mined

3.2.6 Partitioning Transactions Across Chains

As mining is parallelised across multiple chains simultaneously, to prevent the same transactions being included in blocks on different chains the transactions must be partitioned. This is solved using modulo k on the transaction ID to partition the transactions into k subsets, so each chain has a distinct subset of transactions that are allowed to be included on that chain.

3.2.7 Transaction Dependencies

As mentioned in section 2.3.1, there are dependencies between transactions as a transaction references the outputs of other transactions. In Alt-PoW there are multiple chains so dependent transactions may end up on different chains. If transaction *A* depends on transaction *B*, it must be ensured that transaction *B* was in a block before transaction *A*. Therefore, transaction *A* must be included in a block that depends on the block that contains transaction *B*, and at the hand of inter-chain dependencies, these blocks do not necessarily have to be in the same chain. Figure 3.5 displays dependencies between blocks on different chains. In our example, if transaction *B* was in the green block then transaction *A* could be included

in any of the dependent purple blocks on the chain transaction A is allowed to be included in[4].

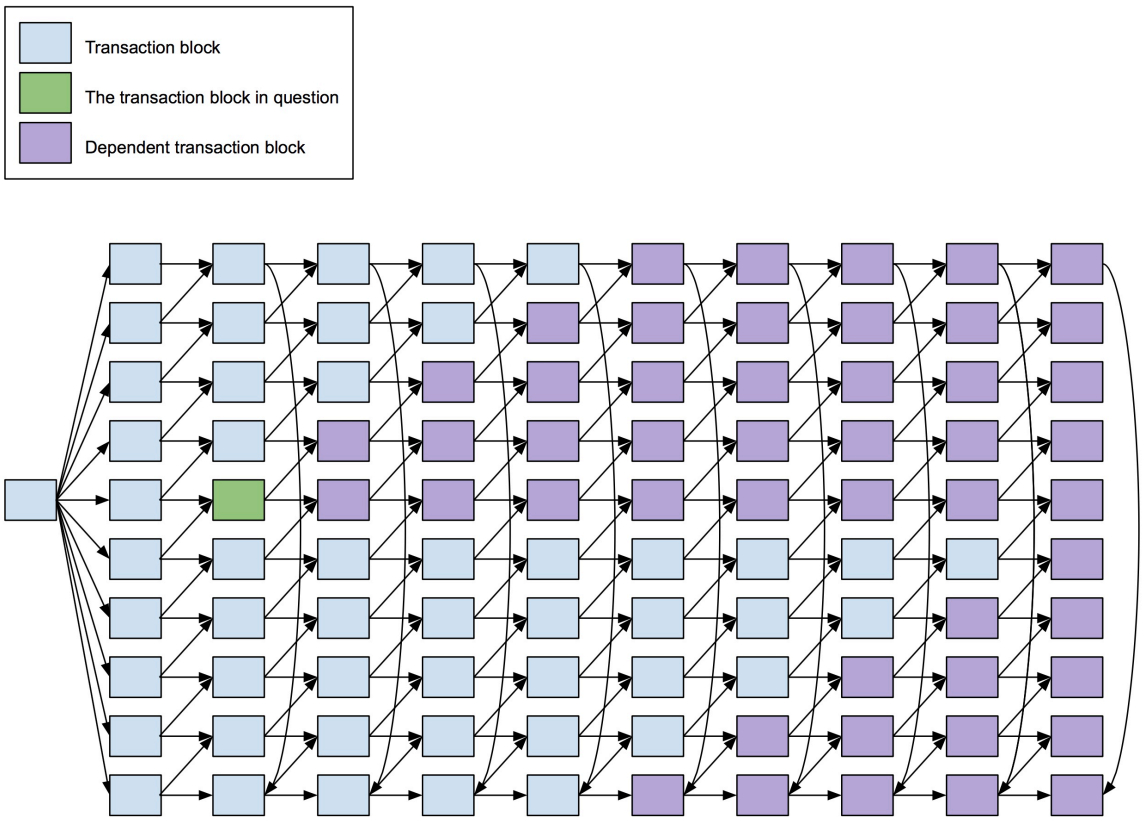

Figure 3.5: Block Dependencies (k = 10)[5]

Double Spending

The output of a transaction can only be spent once [11], however, the situation where Alt-PoW nodes have conflicting ledgers on where a transaction has been spent must be accounted for. If there are two different transactions on different chains, both using the same output of the same transaction, there must be a rule as to which transaction is accepted and which is not. The solution is that the transaction in the block with fewer dependencies

[4]Due to transaction partitioning.

[5]Round blocks have been omitted in this diagram.

overall is accepted. The reasoning behind this is that number of dependencies a block has correlates with the expected arrival of the block (Figure 4.8). Blocks with fewer dependencies are expected to arrive first, so the transactions in their block are accepted over the transactions in a block with more dependencies that is expected to arrive later. Figure 3.6 displays how many dependencies a block has. Towards the beginning of the system there are many blocks with the same number of dependencies. If there are two conflicting transactions in blocks of the same number of dependencies on different chains, the block in the chain with the lower chain index is accepted. However, the system quickly converges to a state where the number of dependencies blocks have goes up linearly by one so there would be no two blocks with the same number of dependencies.

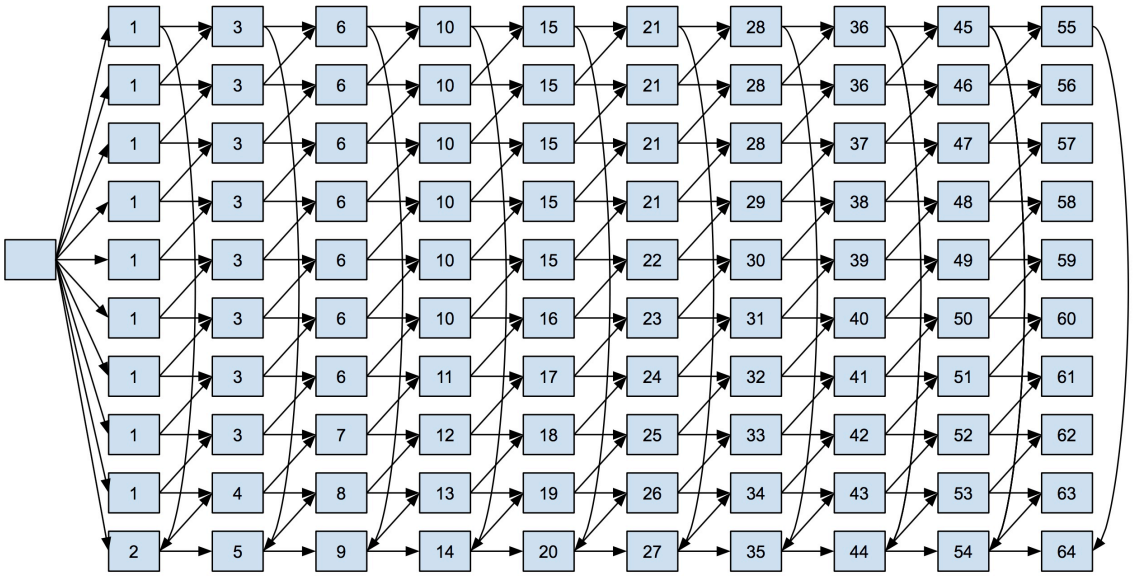

Figure 3.6: Number of Block Dependencies (k = 10)[6]

3.2.8 Tactical Mining

Given that block arrivals on each chain follow an exponential distribution due to the averaging process (subsection 2.3.3), miners can calculate their chances of successfully mining

[6]Round blocks have been omitted in this diagram.

the transaction block based on the current state of the rounds. Due to the memoryless property of exponential distributions, the amount of time a miner has been in a particular round does not factor into their overall chances of mining the transaction block. Disregarding heterogeneous hashing powers, the probability a miner mines a block before another miner is the 50%. The probability a miner mines a block before two other miners is 33.3%, and so on. If a miner calculates their probability of mining the transaction block, they must calculate the probability of them succeeding and every possible future scenario where that could happen. Using this approach, example statistics are shown in Figure 3.7.

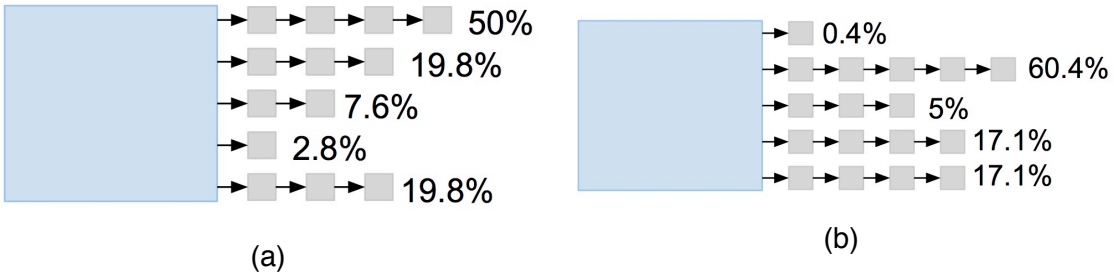

Figure 3.7: Mining Probability ($j = 6$)[7]

These statistics give miners an inclination about which chain is in their best interest to mine on. Mining is no longer a case of mindless computation, miners have the opportunity to make clever choices.

[7] Assuming no other miners join mining this block.

Chapter 4

Evaluation

4.1 Forking

One possible measure of the performance of Alt-PoW is the probability of forking compared to that of PoW.

Poisson processes can be deterministically split into multiple Poisson processes (section 2.3.3). An example of deterministically splitting the block finding process is having a block finding Poisson process per miner based on their proportion of hashing power in the network. For example, if a miner had $\frac{1}{10}^{\text{th}}$ of the hashing power in a network that had a block finding process of rate λ, the miner's block finding Poisson process has a rate of 0.1λ. If three miners with block finding rates of 0.1λ, 0.05λ and 0.15λ were to collude, by merging their individual Poisson processes (section 2.3.3) they would in turn have a combined block finding rate of 0.3λ.

The Alt-PoW block finding process can also be split based on round number; $\frac{1}{j}^{\text{th}}$ of the arrivals are the 1^{st} round blocks, $\frac{1}{j}^{\text{th}}$ are the 2^{nd} round blocks and so on. Therefore, the block finding Poisson process of rate $\frac{j}{x}$ can be split into j smaller Poisson processes of rates $\frac{1}{x}$, each representing the arrival of a particular round. Hence, the arrival of each transaction block is a Poisson process of rate $\frac{1}{x}$.

As discussed in subsection 3.2.4, Alt-PoW periodically adjusts the difficulty of block

problems to maintain block arrivals as a Poisson process with an average rate of $\frac{j}{x}$ blocks per minute based on the network's hashing power over the last 2016 blocks [1]. However, the hashing power put into the last 2016 blocks on a particular chain is not constant; hashing power for the first couple of rounds will start high and then start to deplete when miners drop out. When mining for a particular transaction block commences[1], miners enter with the same chances of mining as each other[2] as they are all starting from the same starting line. Miners who have dropped out on other chains due to falling behind in the rounds will join too. However, as the rounds progress, miners will begin to drop out when they are not faring well, leaving only a fraction of the miners left in the final round for the transaction block.

The probability of forking can be calculated by finding the probability that two transaction blocks occur within a certain period of time. The formula below [17] can be used to find the probability a blocks occur in a particular interval of time:

$$p(a|\lambda) = \frac{e^{-\lambda}\lambda^{a}}{a!} \tag{4.1}$$

where:

$a =$ number of blocks mined in a particular interval
$\lambda =$ number of blocks that is expected to be mined in a particular interval

Let's say a fork occurs if two blocks are submitted within two seconds of each other. The probability of a fork occurring is:

$$p(2|\lambda) = \frac{e^{-\lambda}\lambda^{2}}{2}$$

where:

$2 =$ number of blocks needed to cause a fork
$\lambda =$ number of blocks that is expected to be mined in two seconds

[1] Mining for a transaction block commences when both its dependent blocks are mined.
[2] Disregarding heterogeneous hashing powers.

λ is calculated based on the hashing power of the last 2016 blocks, however, the hashing power remaining in final round of transaction blocks is only a fraction of the overall average hashing power put into mining throughout the rounds. In the first round, the miners' hashing powers combined could result in a block rate of 1.75λ, however, in the final round the remaining miners' hashing powers combined could result in a block rate of 0.25λ.

Therefore, the probability of a fork occurring in the final round of a transaction block when half of the average hashing power is left is:

$$p(2|\frac{\lambda}{2}) \quad = \quad \frac{e^{-\frac{\lambda}{2}}(\frac{\lambda}{2})^2}{2} \quad = \quad \frac{e^{-\frac{\lambda}{2}}\frac{\lambda^2}{4}}{2} \quad = \quad \frac{e^{-\frac{\lambda}{2}}\lambda^2}{8}$$

The probability of a fork occurring in the final round of a transaction block when a quarter of the average hashing power is left is:

$$p(2|\frac{\lambda}{4}) \quad = \quad \frac{e^{-\frac{\lambda}{2}}(\frac{\lambda}{4})^2}{2} \quad = \quad \frac{e^{-\frac{\lambda}{2}}\frac{\lambda^2}{16}}{2} \quad = \quad \frac{e^{-\frac{\lambda}{2}}\lambda^2}{32}$$

Plotting these functions (Figure 4.1), it is clear that the probability of forking is reduced by only a fraction of the network's hashing power being left in the final round.

In Bitcoin, one block is expected to arrive every 10 minutes, so the amount of blocks we expect to arrive in a 2 second interval is $\frac{1}{300}$. Therefore, the probability of forking $p(2|\frac{1}{300}) = 5.54{*}10^{-6}$, however, in Alt-PoW, with half the average hashing power remaining in the final round $p(2|\frac{1}{600}) = 1.39 * 10^{-6}$ and with a quarter of the average hashing power remaining $p(2|\frac{1}{1200}) = 3.47 * 10^{-7}$. Therefore, it can be concluded that the probability of forking is reduced in Alt-PoW as compared to PoW.

4.2 Target Block Time

As discussed in section 2.3.2, the relatively long target time for a block in Bitcoin was chosen to be 10 minutes due to the risk of forking. However, the probability of forking

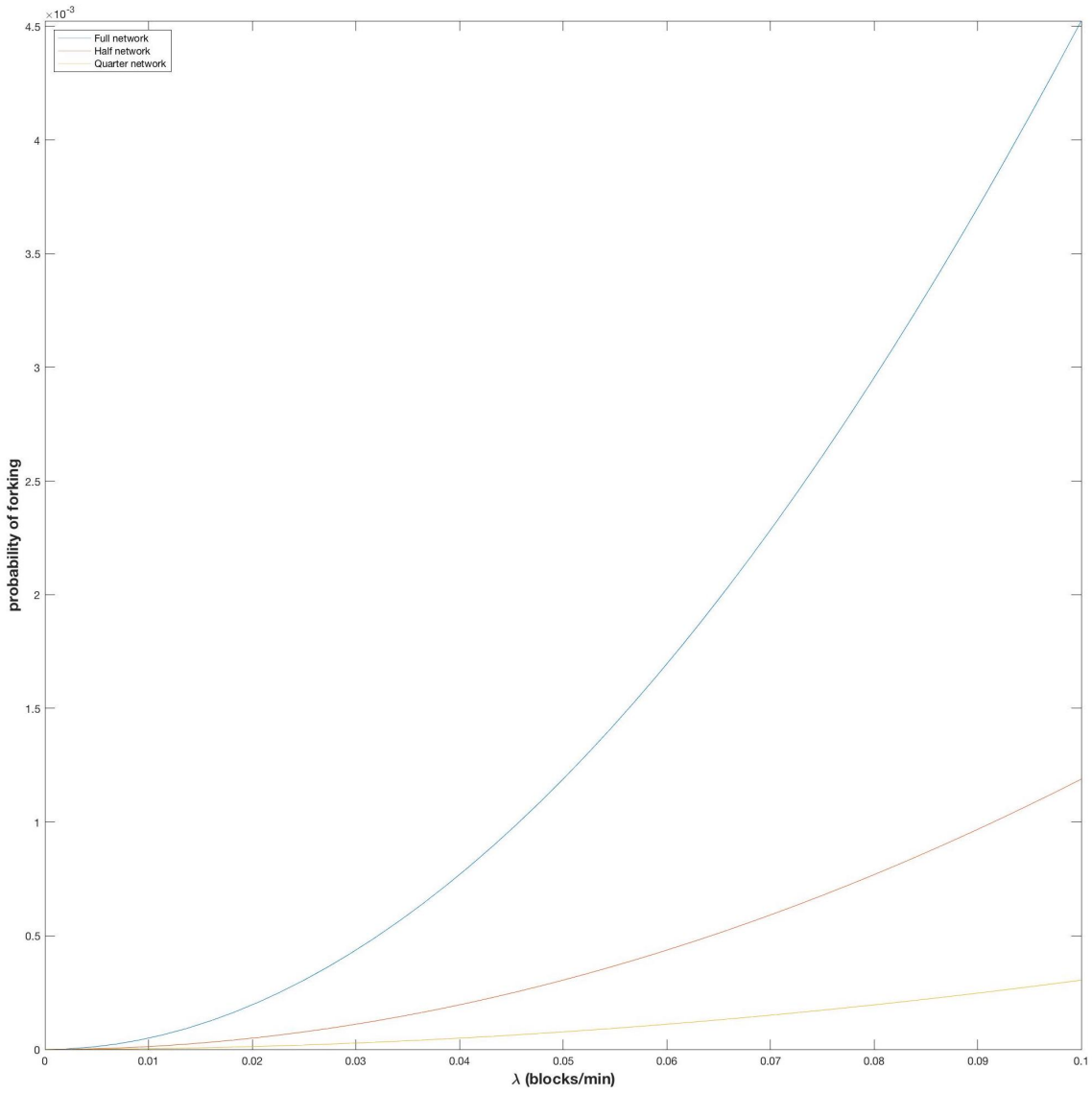

Figure 4.1: Probability of Forking in Final Round of Alt-PoW Based on Remaining Hashing Power

is reduced in Alt-PoW (section 4.1) so the target block time can also be reduced and yet maintain the same probability of forking. The factor at which the target time can be reduced is dependent on what fraction of the hashing power is left in the final round of mining a

transaction block. If there is half of the average hashing power on a chain left in the final round, the target time can be reduced to 5 minutes, maintaining a probability of forking of $5.54 * 10^{-6}$. If there is a quarter of the average hashing power left in the final round, the target time can be reduced to 2.5 minutes whilst maintaining the same probability of forking.

In the first few rounds of mining there will be more round blocks submitted than the average block rate, however, forking is not an issue here. Miners mine on their own round block chains so there are no forks involved. There is no incentive for two different miners to compete on the same round block chain. They must use the same coinbase transaction for their round blocks to be accepted so if one were to be successful, the miner receiving the block reward is already predetermined. If two miners wanted to collude and mine together it would be more in their benefit to unite their resources and mine together rather than competing on the same round block chain.

4.3 Rate of Block Creation

Given that the time between transaction blocks on a sub-chain follows an exponential distribution[3], the expected times of block arrivals can be calculated. Consider a network with $k = 10$ sub-chains; the initial genesis block and first block of each sub-chain are generated on creation. The expected time for blocks 1-1[4] (the first block mined on chain 1) through to 9-1 (the first block mined on chain 9) is x minutes as all of these blocks do not have any dependencies to be mined. However, block 10-1 depends on block 1-1 so its expected arrival is the expected time for 1-1 to be mined + the expected time for it to be mined, therefore, its expected arrival time is $2x$ minutes after creation of the system. Block 1-2 depends on block 1-1 and 2-1, its expected arrival is not as obvious to calculate:

[3]The time starting from when both block dependencies have been mined.

[4]Format: $chain\_index$ - $block\_number$. See Figure 3.3.

$$E[T_{2-1}] = E[t_{d_1}] + E[r_{d_2}] + E[t_{2-1}]$$

where:

$E[T_{2-1}]$ = expected arrival time of block 2-1
$E[t_{d_1}]$ = expected time taken for the first of two dependencies to be mined
$E[r_{d_2}]$ = expected remaining time taken for the second of two dependencies to be mined, given the first has already arrived
$E[t_{2-1}]$ = expected time taken for block 2-1 to be mined, given both its dependencies have been mined

When both block 1-1 and 2-1 are being mined, they are both independent exponential distributions, so combining them results in combining their rates (section 2.3.3) $\frac{1}{x} + \frac{1}{x} = \frac{2}{x}$. Therefore, the expected time for the first block to arrive is $\frac{x}{2}$ as opposed to x. Due to the memoryless property of exponential distributions [23], the expected remaining time for the second block to be mined is x minutes, regardless of how long the first block took to mine. When both the dependent blocks are mined, the expected time for block 2-1 to be mined is x. Therefore:

$$E[T_{2-1}] = \frac{x}{2} + x + x = 2.5x$$

Calculating arrival times for the succeeding blocks follow the same process, Figure 4.2, Figure 4.3 and Figure 4.4 display the expected arrival times of the first 25, 40 and 50 blocks in the network, respectively.

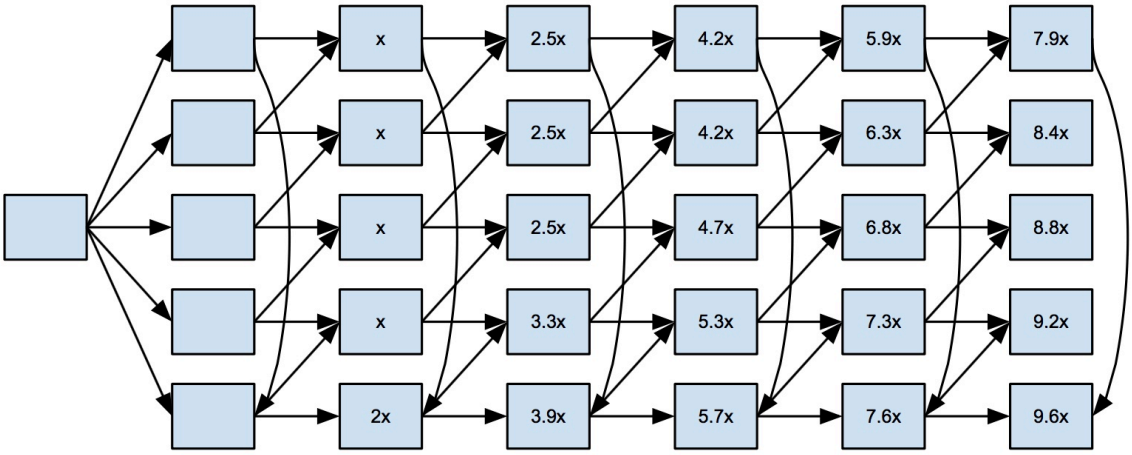

Figure 4.2: Expected Arrival of Blocks (k = 5)

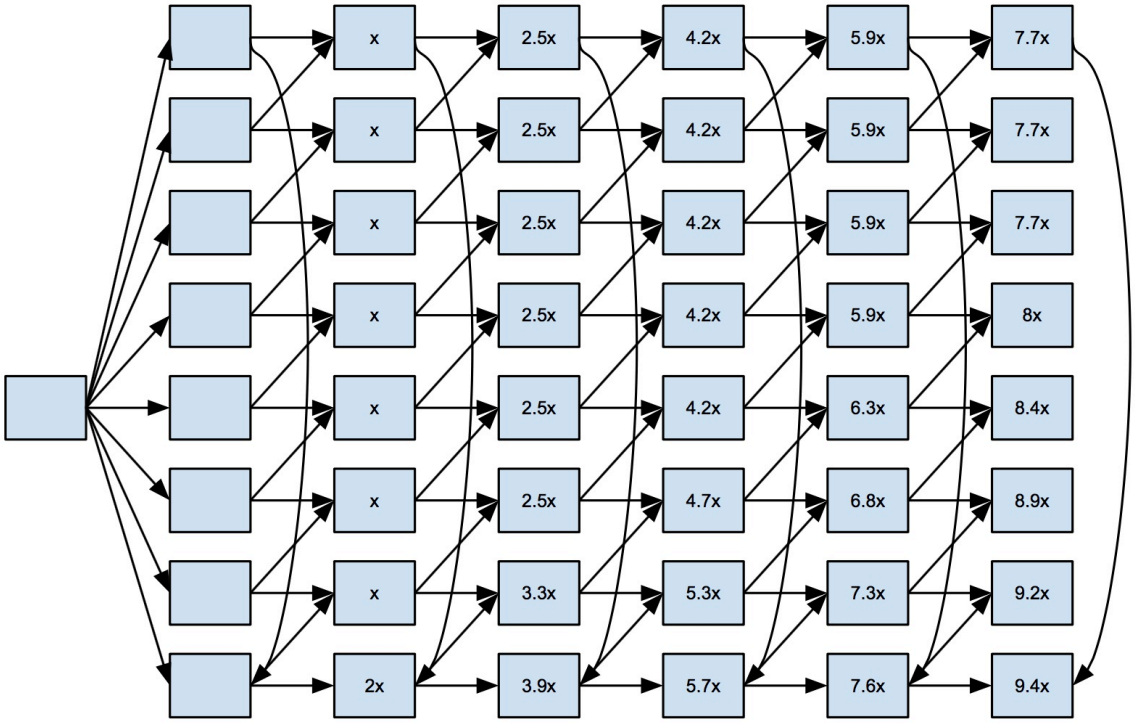

Figure 4.3: Expected Arrival of Blocks (k = 8)

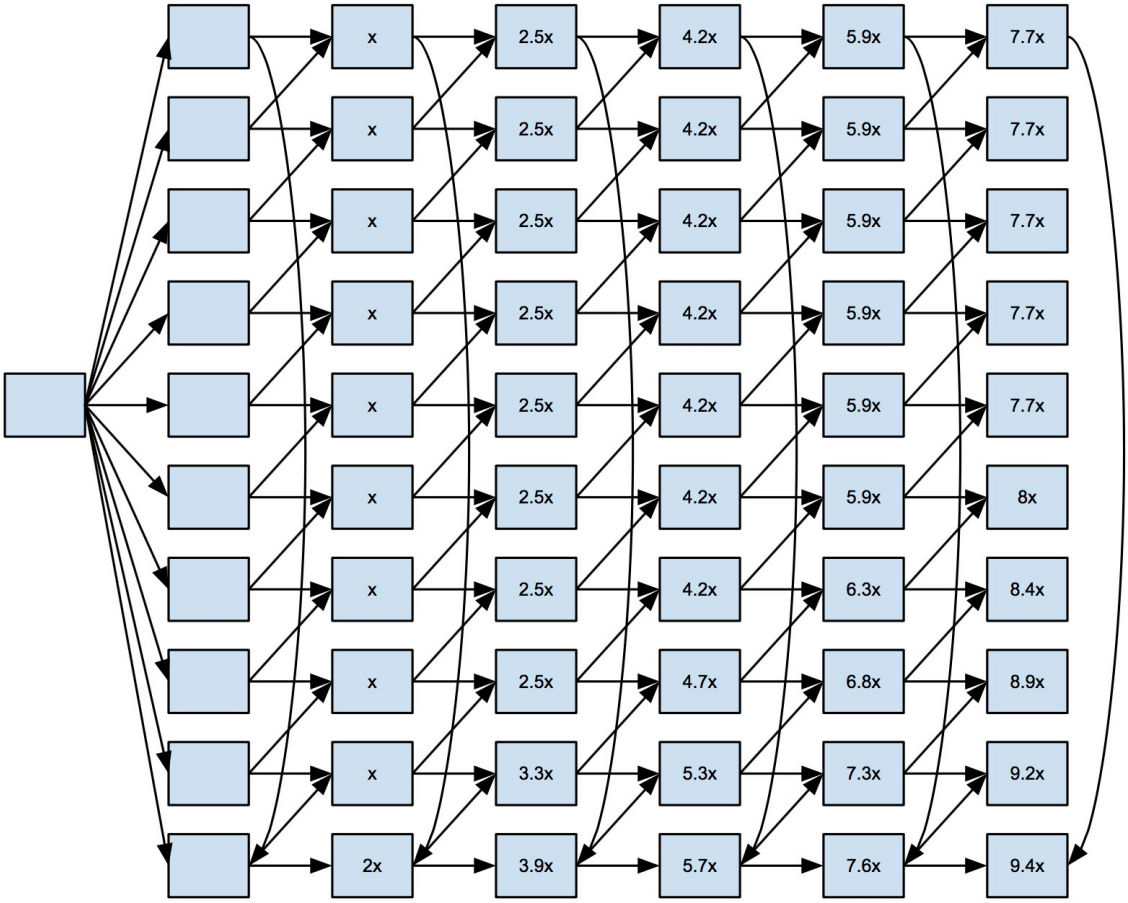

Figure 4.4: Expected Arrival of Blocks (k = 10)

The data is plotted with number of blocks mined vs. time taken (Figure 4.5, Figure 4.6 and Figure 4.7); as the target block time can be reduced (section 4.2), the data was plotted with x = 2.5 and 5, where x is the target time. Using the Matlab function `polyfit`, the data was fitted to a linear function to find the slope, which is the number of overall number of transaction blocks mined per minute. The results are displayed in Table 4.3.

For x = 2.5 and k = 10 the block rate was 2.1; compared to PoW's block rate of 0.1, Alt-PoW fared **21 times faster**.

As a result of the increased block rate, a higher volume of transactions are confirmed per second and so there is less competition between transactions to get into a block. This will in turn, reduce competitive transaction fees (see section 2.3.5).

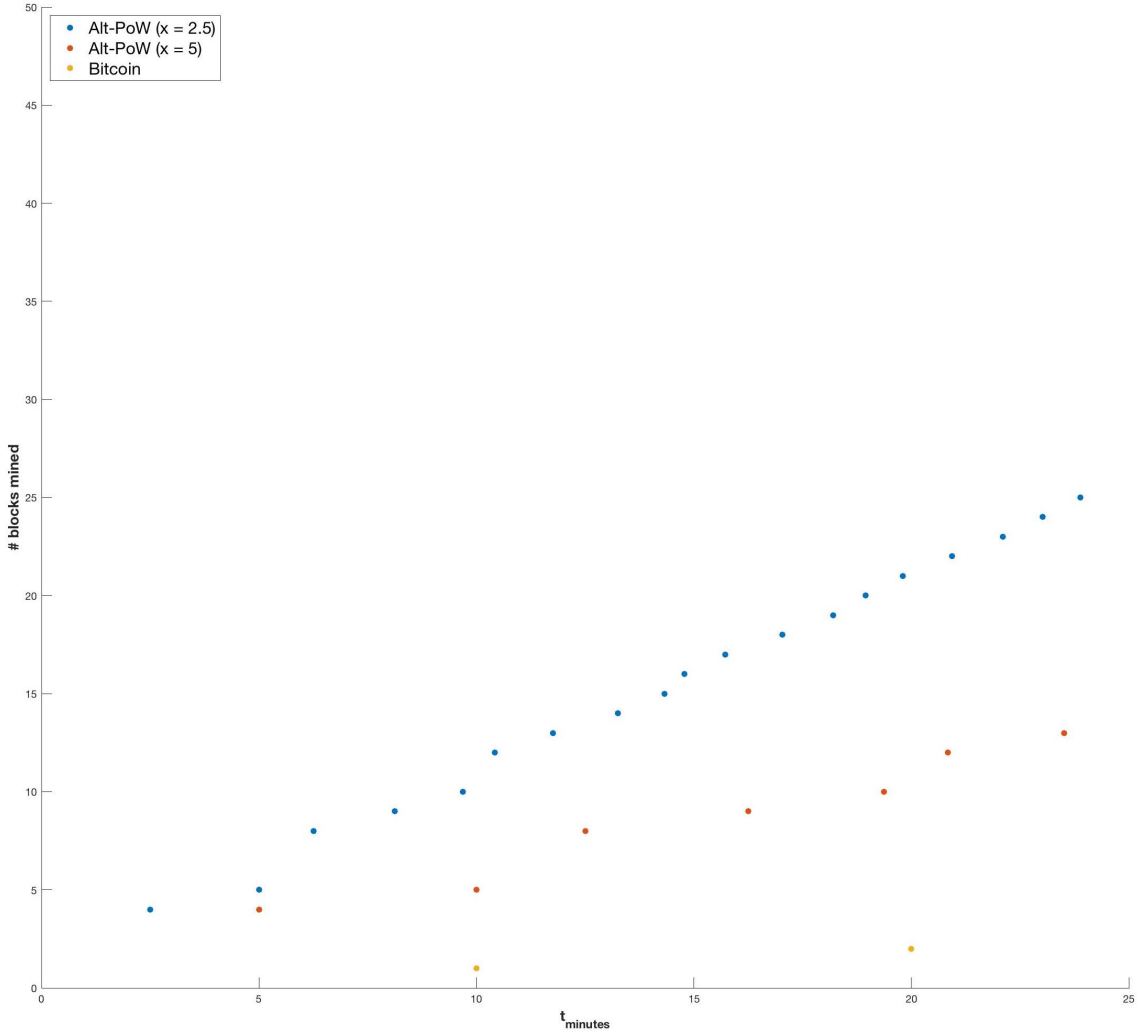

Figure 4.5: Expected Arrival of Blocks (k = 5)

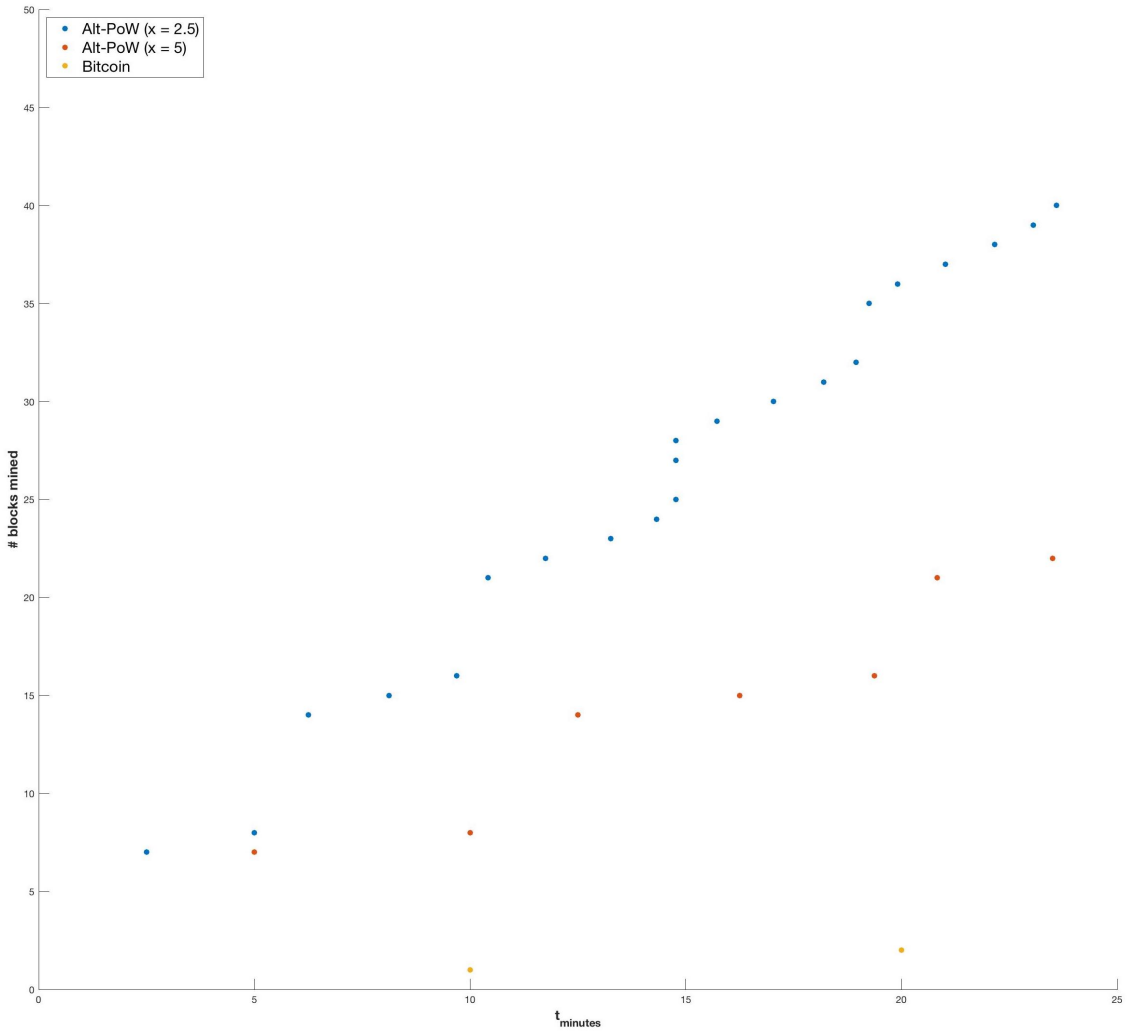

Figure 4.6: Expected Arrival of Blocks (k = 8)

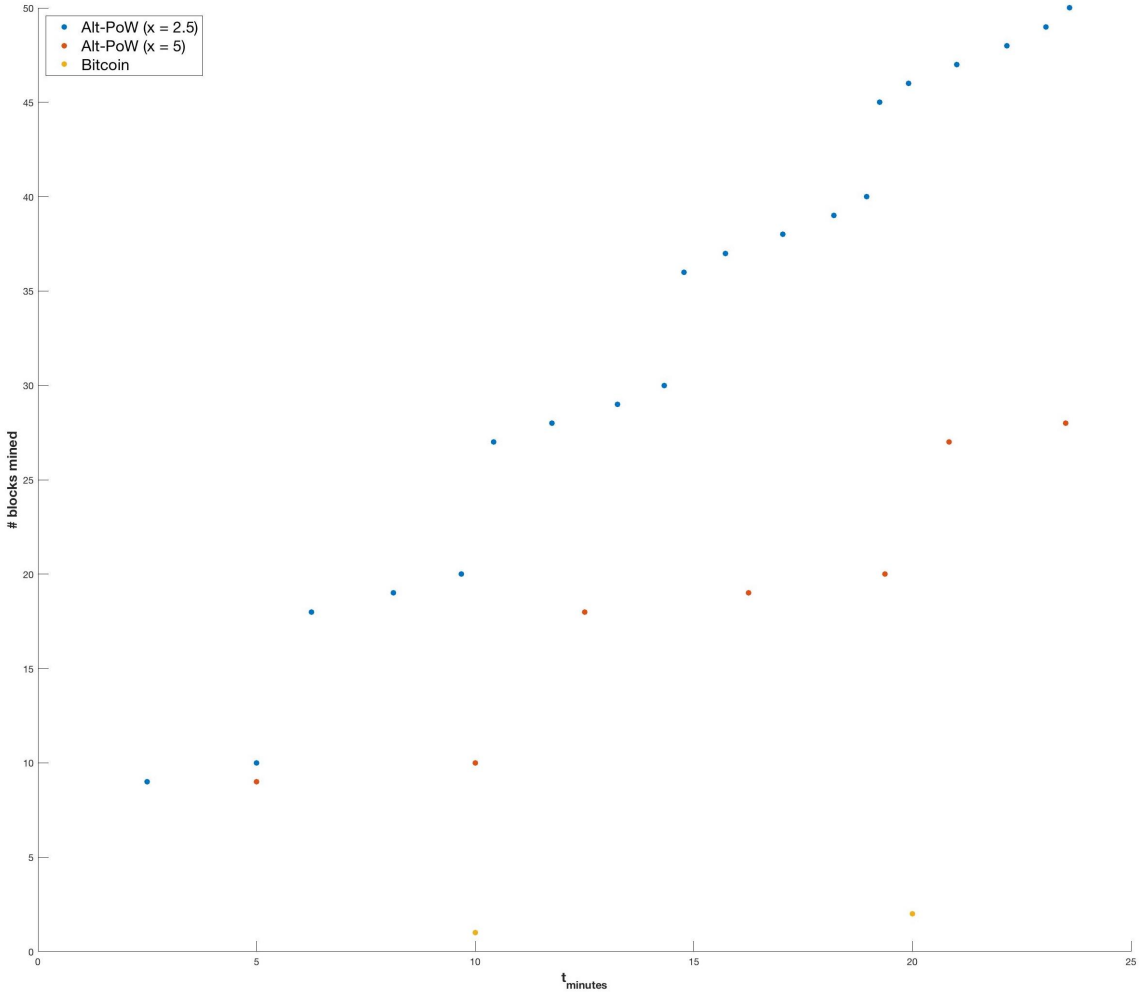

Figure 4.7: Expected Arrival of Blocks (k = 10)

Table 4.1: Transaction Block Rate

x \ k	**5**	**8**	**10**
2.5	1	1.63	2.05
5	0.5	0.82	1.03

Table 4.2: Total Block Rate

x \ k	**5**	**8**	**10**
2.5	5	13.04	20.5
5	2.5	6.56	10.3

Table 4.3: Alt-PoW Block Rates (blocks/min)

4.3.1 Future Block Rate

By plotting the time a block is expected to arrive vs. the number of dependencies the block has (Figure 4.8), a linear relationship between the two variables is seen. The number of dependencies each block has increases linearly over time (Figure 3.6), therefore, the expected arrivals of future blocks will also follow a linear trend.

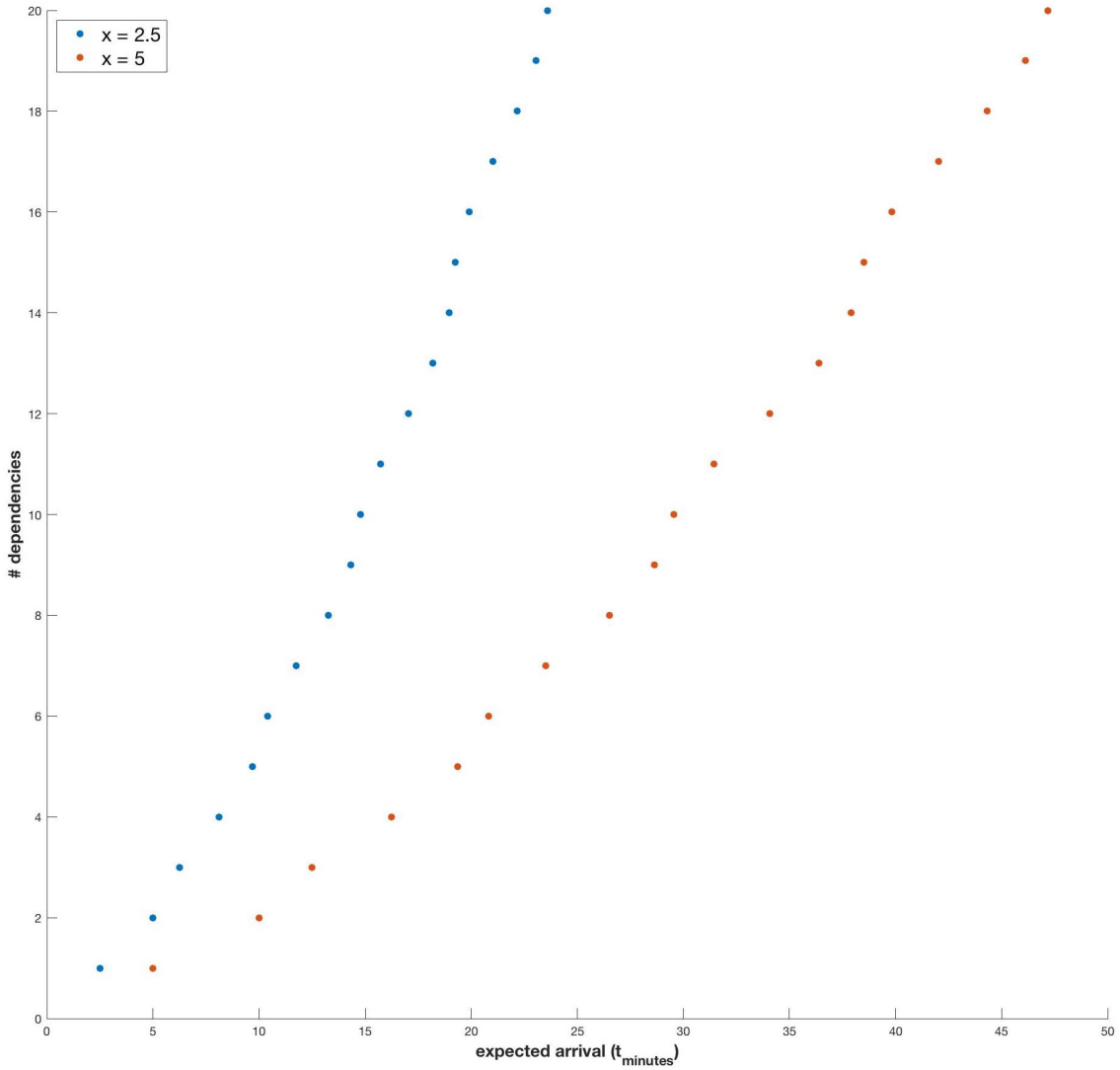

Figure 4.8: Expected Arrival of Blocks vs. Number of Block Dependencies (k = 10)

4.4 Confirmation Time

As the block rate in Alt-PoW is faster than in PoW (section 4.3), the confirmation rate has also increased by the same scale. This is due to a six block confirmation not having to be on the same chain as the block to confirm. Confirmation blocks need only be dependent on the block to confirm, so they could be on a different chain. See Table 4.3 for Alt-PoW's block rates.

4.5 Energy Consumption per Block

At the hand of distributing hashing power across multiple chains, reducing the target block time and miners dropping out of the mining race as the rounds go on, the energy consumption used per block has reduced. Assuming $\frac{1}{f}^{\text{th}}$ of the network's hashing power starts mining a particular block on one of the k sub-chains, and the hashing power is reduced by half every round from miners dropping out, the following equation is derived:

$$E_{AltPoW} = \frac{x_{AltPoW}}{x_{PoW}} * \frac{1}{f} * (\sum_{i=1}^{j} \frac{1}{i * j}) * E_{PoW} \tag{4.2}$$

where:

E_m = energy consumption per block in protocol m
x_m = expected time for a block to be mined in protocol m
$\frac{1}{f}$ = proportion of the network's hashing power initially mining on this chain
j = number of rounds

This formula was created to formalise the concept that energy consumption has been reduced in Alt-PoW, which is an important consideration for blockchain based networks. Plotting $\frac{x_{AltPoW}}{x_{PoW}} * \frac{1}{f} * (\sum_{i=1}^{j} \frac{1}{i*j})$ for x_{AltPoW} = 2.5, x_{PoW} = 10, j = 10, the energy consumption of Alt-PoW can be viewed as a fraction of the energy consumption of PoW (Fig. 4.9).

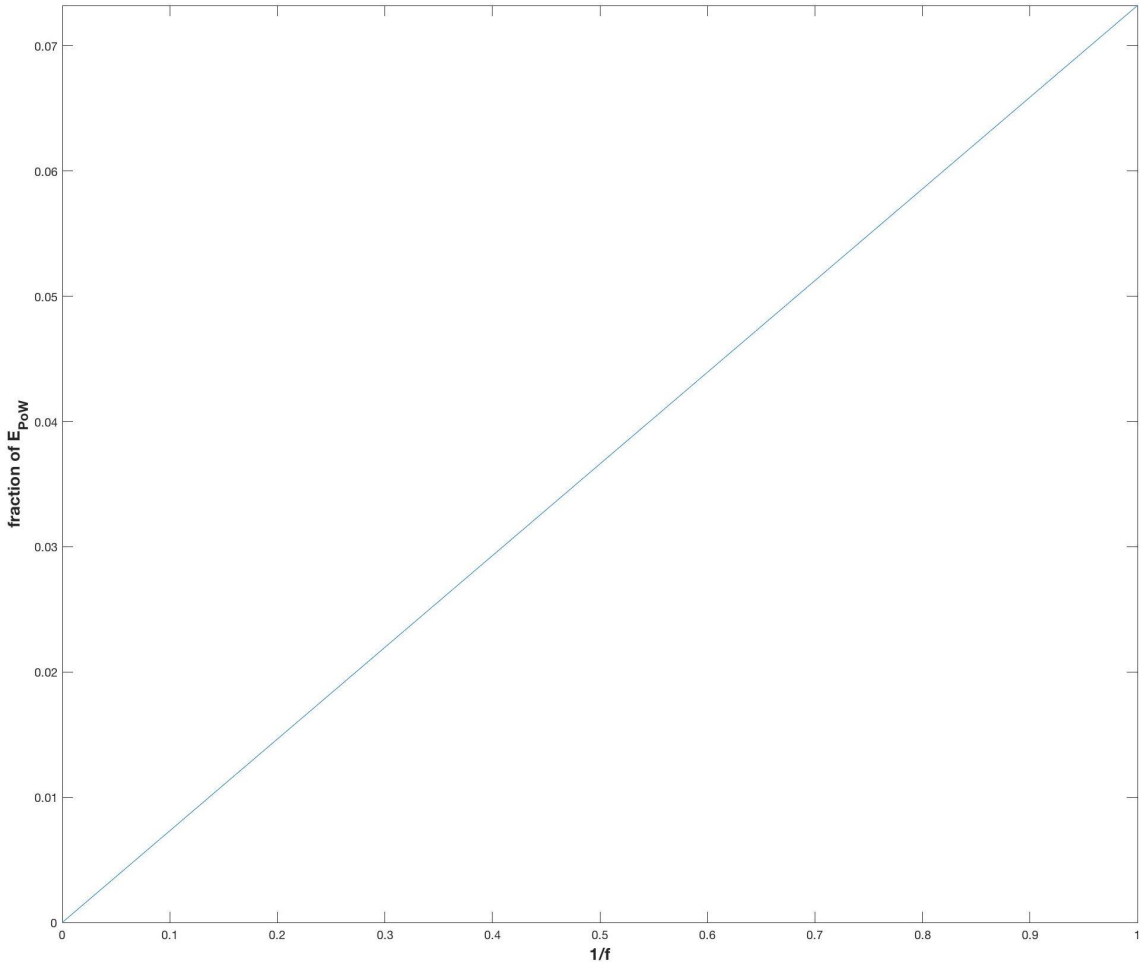

Figure 4.9: Energy Consumption of Alt-PoW as a Fraction of PoW's Energy Consumption (x^5 = 2.5, k = 10)

Considering a range of 0.1 to 0.9 of the network's hashing power starting on an individual block, the energy consumption of Alt-PoW is reduced by a factor of 0.007 to 0.066, which is **between 15 and 137 times less** than PoW's energy consumption per block.

[5] Where x is the target block time.

4.6 Mining Diversity

With mining distributed across k sub-chains, mining diversity is naturally increased. As seen in Figure 2.3, there are three large mining pools that dominate the network. In an Alt-PoW protocol, these mining pools would have to choose between using all of their hashing power on one chain, resulting in less competition on other chains, or distributing their hashing power across multiple chains. By distributing their hashing power these miners are weaker on an individual chain, so other miners on one of those chains face a better chance of mining than they would have being in competition against all of the dominant miner's hashing power.

4.7 Inter-node Communication

With the increased transaction block rate (section 4.3) there are more transaction blocks being submitted and propagated around the network per minute, increasing traffic on the peer-to-peer (P2P) links. In Table 4.3 the increased block rates are displayed. For x = 2.5, k = 10 and j = 10, there are 20.5 blocks submitted per minute, not including round blocks that were abandoned from miners dropping out. Efficient inter-node communication is essential in Alt-PoW for miners to be able to see a consistently updated view of the network to know what miners are in what round and to calculate which chain is deemed most prosperous to use their resources on. Therefore, there is a higher importance on the efficiency and reliability of the inter-node communication.

Alt-PoW performs very well under scrutiny by the various metrics. By cutting the target block time by a factor of 4 and introducing k = 10 sub-chains, it achieves a block rate and confirmation time **21 times faster** than that of Bitcoin without compromising on security, and between **15 and 137 times less energy consumption**. However, this is a compromise for increased traffic on P2P links. Mining diversity is increased which also strengthens the security of the protocol.

Chapter 5

Potential Attacks

5.1 Double Spending

A possible attack on the system is a miner creating two transactions that spend the output of the same transaction, and these transactions are mined simultaneously on separate chains. However, as discussed in section 3.2.7, a rule can be established so that the block with the greater number of dependencies containing a conflicting transaction will be considered invalid; honest miners will follow this rule. The network are incentivised not to mine on top of invalid blocks as, assuming the majority of the network nodes are honest, mining on an invalid block will have been a waste of hashing power as the majority of the network will not continue on this chain.

5.2 Alternative History Attack

As discussed in section 2.3.4, an alternative history attack requires resources and good fortune from the attacker. In Alt-PoW the hashing power of the network is divided across multiple chains, resulting in less hashing power on an individual chain. Less competition on each chain implies a better probability for each miner to mine, including attackers. Attackers have a better probability of mining a single block on a chain, however, their chances

of rewriting history after six confirmations is not as easy. Six confirmations do not have to be on the same chain, therefore, if an attacker wishes to rewrite history they must have enough hashing power to rewrite history on multiple chains. Figure 3.5 illustrates how the number of dependent blocks increases in a triangular number sequence until it reaches a point where all blocks on all chains depend on the block in question; this is juxtaposed with Bitcoin's block dependency rate of 1 block every ten minutes. With an increased block rate (section 4.3) and the rate of dependencies increasing in a triangular number sequence and then in multiples of k, it is more difficult for a miner to rewrite history in Alt-PoW as they would need the hashing power to mine multiple blocks on multiple chains to beat the main network.

5.3 51% Attack

Alt-PoW is grounded with the same underlying principle as in PoW: CPU power represents votes in a decision making process in the network. The more votes a party has the more contribution they have to network's conformation, and if a single party has the majority of the votes in the network then they have the opportunity to control the network in their favour. Therefore, Alt-PoW is still vulnerable to a 51% attack.

5.4 Concealing Round Blocks

Although miners are incentivised to publish their round blocks (section 3.2.2), there may be an opportunity for a strong miner to take advantage of this. If a miner had enough hashing power, they could successfully mine all round blocks on one chain without publishing their blocks, luring miners to keep mining on this chain. Whilst miners are lured onto this chain, the strong miner could mine on other chains. When other miners are getting close to finding a solution on the original chain, the strong miner submits their blocks and the lured miners' time and resources were wasted. The effectiveness of this attack is uncertain as it depends on how quick the strong miner finds all the round blocks ahead of the other miners on that

chain. If the strong miner only finds all the round blocks 15 seconds before the other miners are close to a solution, the effect of this attack is minimal. Also, as miners are divided across k sub-chains and miners naturally start to drop out of rounds as time goes on, only a small portion of miners will have been falsely lured to keep mining on the particular chain.

Chapter 6

Future Work

6.1 Alt-PoW as a Markov Chain

A Markov chain represents a mathematical system that transitions between states over time depending on certain probabilistic rules; each state is dependent only on the previous state [24]. One area of future work could be to look at the Alt-PoW system as a Markov chain. This would be incredibly useful for analysing the Alt-PoW mechanism as it would capture all possible network activity that would influence future evolution of the protocol, and could be used to predict Alt-PoW's behaviour long-term.

6.2 Ideal Values for k, j and x

In this report different values for k, j and x were analysed and their performance was measured, however, how these values cope in a real network situation is still uncertain. The speed of the network increases with the number of sub-chains, yet too many sub-chains could lead to a very small amount of hashing power per chain and ultimately reduce the security of the network. The value for x was chosen based on sustaining a probability of forking the same as in Bitcoin's PoW, however, there are many disparities between Alt-PoW and PoW so further investigation could show that Alt-PoW's value for x could be

reduced and still maintain an efficient network. A beneficial future undertaking would be to run simulations of real networks and how the different values for k, j, and x perform, to find the most optimal solution. The number of rounds was chosen to be equal to the number of sub-chains based on the idea that it would keep the network busy but not have a redundant amount of rounds, however, simulating a real network could also present a better alternative.

6.3 Typical Behaviour of Miners

The behaviour of miners in Alt-PoW is difficult to predict and replicate. The rate at which miners drop out and how long they tend to stay committed to a chain would depend on the state of other chains and what miners consider to be to 'the best' strategy that would have the best likelihood of them succeeding for the minimal amount of resources needed. Hashing power across the network is heterogeneous; some miners have much more computational resources than others. An interesting statistic to gather would be how a miner acts in a network dependent on how much hashing power they have, as it is reasonable to expect that miners with minimal hashing power would act different in the network than dominant miners. If the typical behaviour of a network of diverse miners was able to be replicated, calculations such as the average energy consumption per block would be more accurate to gauge.

6.4 Increased Peer-to-Peer Traffic

In section 4.7, it is discussed how traffic on P2P links will increase due to more round blocks and transaction blocks being submitted per second. Further examination on the possible strain this may put on the efficiency of the network's inter-node communication would be essential to characterise the network's overall performance.

6.5 The Rate at which Mining Diversity Increased

As discussed in section 4.6, the distribution of hashing power across multiple chains constrains dominant miners to choose whether to split their resources across multiple chains or to steer their resources toward one chain. This results in either the reduction of their dominance per chain or a large dominance on one chain, leaving the other chains for other miners to mine on. Either scenario implies a better opportunity for weaker miners to mine on a chain where they are not against such a dominant opponent, entailing increased mining diversity. However, further investigation into the actual rate of how mining diversity is increased would provide more concrete results.

6.6 Security

An essential objective of this project was to maintain the protocol's security; an insecure protocol is not a protocol that would attract and be adapted by honest participants. Simulations of attacks on the network when the network is in various states would replicate how the network could handle such attacks and how successful attackers may be. This investigation could provide results as to what are the most vulnerable states the network can be in and how to avoid the network entering one of these states. The use of a Markov chain here may also be worth exploration.

Chapter 7

Conclusion

The goal of this project was to bring strategy into mining, and Alt-PoW has done so. Miners want rewards, not to waste their resources, so once the opportunity to make intelligent decisions about where they put their resources is presented to them, this naturally results in a more efficient protocol.

The use of round blocks has introduced the idea of advancement in mining, a concept lost in Bitcoin's PoW. As round blocks are propagated around the network, miners observe each other's progress, giving them an inclination as to whether it is worth exhausting their resources over a block they have minuscule chances of mining or to continue pursuing. Miners dropping out of rounds dramatically reduces the energy consumption of a particular block.

Multiple inter-connected chains in Alt-PoW has allowed for parallelised mining, thus increasing the block rate, confirmation time and mining diversity, yet maintaining security due to the inter-chain dependencies. Not only can miners decide to drop out of mining a particular block when they are not faring well, but can also view the network as a whole and make smart and resourceful decisions as to which chain to mine on next. Energy consumption per block has reduced further by dividing the network's hashing power across multiple chains, and an increased block rate implies reduced transaction fees and delivers a less sluggish protocol.

Miners drop out of rounds when other miners progress, resulting in the probability of forking being reduced in the final round. PoW's conservative 10 minute target block time was chosen as a compromise between the risk of forking and confirmation time for a block, therefore, Alt-PoW's target block time is less than that of PoW yet with no increased probability of forking.

The security of Alt-PoW is proven to be no less than that of PoW, which is essential for its validity as a consensus mechanism. Satoshi Nakamoto introduced Bitcoin as the first completely decentralised payment system, removing the need for a trusted third party to carry out transactions. There had been attempts to create such a system before this, but it was Bitcoin that revolutionised digital currencies as its security was robust. By parallelising mining, Alt-PoW has increased mining diversity by giving weaker miners the opportunity to only compete against a subsection of the network as opposed to its entirety; mining diversity also strengthens the security of the protocol.

This project met the objectives it aimed to achieve, and there are still many possible areas of future exploration with Alt-PoW.

Bibliography

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

[2] A. S. Hayes, "Cryptocurrency value formation: An empirical study leading to a cost of production model for valuing bitcoin," *Telematics and Informatics*, vol. 34, no. 7, pp. 1308–1321, 2017.

[3] R. Pass and E. Shi, "Rethinking large-scale consensus." Cryptology ePrint Archive, Report 2018/302, 2018. `https://eprint.iacr.org/2018/302`.

[4] M. Jakobsson and A. Juels, "Proofs of work and bread pudding protocols," in *Secure Information Networks*, pp. 258–272, Springer, 1999.

[5] A. Back *et al.*, "Hashcash-a denial of service counter-measure," 2002.

[6] S. King and S. Nadal, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake," *self-published paper, August*, vol. 19, 2012.

[7] J. Siim, "Proof-of-stake,"

[8] S. Popov, "The tangle, version 1.4.2," *cit. on*, 2018.

[9] A. Churyumov, "Byteball: A decentralized system for storage and transfer of value."

[10] I. Foundation, "Iot chain white paper, a high-security lite iot os," 2018.

[11] N. Atzei, M. Bartoletti, S. Lande, and R. Zunino, "A formal model of bitcoin transactions." Cryptology ePrint Archive, Report 2017/1124, 2017. `https://eprint.iacr.org/2017/1124`.

[12] E. K. Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford, "Enhancing bitcoin security and performance with strong consistency via collective signing," in *25th USENIX Security Symposium (USENIX Security 16)*, pp. 279–296, 2016.

[13] "Bitcoin wiki." `https://en.bitcoin.it/wiki/`.

[14] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "Sok: Research perspectives and challenges for bitcoin and cryptocurrencies," in *Security and Privacy (SP), 2015 IEEE Symposium on*, pp. 104–121, IEEE, 2015.

[15] H. Vranken, "Sustainability of bitcoin and blockchains," *Current Opinion in Environmental Sustainability*, vol. 28, pp. 1–9, 2017.

[16] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*, pp. 1–10, IEEE, 2013.

[17] J. Marchini, "The poisson distribution," 2008.

[18] J. Pitman, "Poisson–dirichlet and gem invariant distributions for split-and-merge transformations of an interval partition," *Combinatorics, Probability and Computing*, vol. 11, no. 5, pp. 3–6, 2002.

[19] A. E. Holroyd, R. Lyons, and T. Soo, "Poisson splitting by factors," *Ann. Probab.*, vol. 39, pp. 1938–1982, 09 2011.

[20] J. R. Douceur, "The sybil attack," in *International workshop on peer-to-peer systems*, pp. 251–260, Springer, 2002.

[21] N. Houy, "The economics of bitcoin transaction fees," 2014.

[22] M. Thum *et al.*, "The economic cost of bitcoin mining," in *CESifo Forum*, vol. 19, pp. 43–45, Ifo Institute-Leibniz Institute for Economic Research at the University of Munich, 2018.

[23] K. Balakrishnan, *Exponential distribution: theory, methods and applications*. CRC press, 1996.

[24] W. R. Gilks, S. Richardson, and D. Spiegelhalter, *Markov chain Monte Carlo in practice*. CRC press, 1995.

Appendix A

Source Code

The source code for this project can be found on the CD attached.