Erlend Lid Helland

On Graph Based Cryptocurrency Systems

Master's thesis in Communication Technology
Supervisor: Colin Alexander Boyd, Chris Carr
June 2019

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Information Security and Communication Technology

NTNU
Norwegian University of
Science and Technology

Erlend Lid Helland

On Graph Based Cryptocurrency Systems

Master's thesis in Communication Technology
Supervisor: Colin Alexander Boyd, Chris Carr
June 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Information Security and Communication Technology

Title: On Graph Based Cryptocurrency Systems

Student: Erlend Lid Helland

Problem description:

In recent years doubt has started to rise around Bitcoin's capability to deliver on being a global digital currency for everyday use, due to its limitations. In the group of challengers to this position, a subgroup has formed of systems that swap out the blockchain as a distributed data ledger in favour of Directed Acyclic Graphs (DAGs). This results in cryptocurrencies that instead of storing their data in a blockchain, store it in a graph structure where the data is located in the vertices of the graph. The change directly affects different properties of the system such as throughput, consensus and latency.

This thesis aims to explore how the change in data structure affects the properties of cryptocurrency systems, both through a literature study and experiments on different live systems. It will look at how different real world cryptocurrencies deal with the switch of data structure and the implication it has for the overall system. To help decide if this can solve the current challenges of blockchain based systems, a discussion of the advantages and disadvantages introduced will be conducted. The thesis will also consider how the findings compare to the promises and expectations raised by the proponents of this shift from blockchain to graphs.

Responsible professor: Colin Alexander Boyd, IIK

Supervisor: Chris Carr, IIK

Abstract

This thesis investigates and compares metrics of cryptocurrencies based on blockchains to those based on Directed Acyclic Graphs (DAG). Blockchain-based cryptocurrencies are facing multiple challenges in domains such as scalability, decentralization and resource utilization. This thesis investigates whether cryptocurrencies utilizing a DAG as their data structure can offer solutions to these issues. To answer this question, both a theoretical study and experiments involving different cryptocurrencies are carried out. The theoretical study is aimed at gathering data on different systems for comparison of the different solutions and implementations. The experiments are aimed at the DAG-based systems since these are much younger and have minimal research looking into them.

The result of the thesis shows that in terms of scalability, DAGs do offer better solutions in most regards. The thesis also points out multiple challenges faced by DAG based cryptocurrencies, such as a denial of service attack and the lack of incentives in the system. The most significant advantage to blockchain solutions is identified as being their proven security and longevity in this space. This thesis emphasizes the trade-offs made when moving from a blockchain to a DAG as the data structure in a cryptocurrency system. In summary, DAG systems offer better scalability but do not provide the same trust assumptions as the previous blockchain solutions.

Sammendrag

Denne oppgaven undersøker og sammenligner trekk ved kryptovaluta basert på blokkjeder med kryptovaluta basert på Rettede Asykliske Grafer (RAG). Blokkjedebasert kryptovaluta møter utfordringer på felt som skalering, desentralisering og ressursutnyttelse. Denne oppgaven undersøker om kryptovaluta som bruker en RAG som sin datastruktur kan løse noen av disse utfordringene. For å besvare dette spørsmålet ble både en teoretisk studie og forskjellige eksperimenter på ulike kryptovalutaer gjennomført. Den teoretiske studien ble gjennomført for å skaffe data på de forskjellige systemene til bruk i sammenligninger mellom dem. Eksperimentene fokuserte på RAG-systemene, siden disse er mye nyere og har mindre forskning enn systemene basert på en blokkjede.

Resultatene i denne oppgaven viser at RAG systemer generelt sett har bedre løsninger for skalerbarhet. Oppgaven viser også til flere utfordringer disse RAG baserte systemene har, som for eksempel distribuert tjenestenektangrep og en mangel på insentiver i systemene. Oppgaven peker på demonstrert sikkerhet og lang levetid som de største fordelene blokkjedesystemene har over RAG-systemene. Oppgaven legger vekt på valgene som er gjort når man går fra blokkjede til RAG som datalager i en kryptovaluta. Det konkluderes med at RAG-systemer tilbyr bedre skalerbarhet men de har ikke samme tillit blant brukere som blokkjedesystemene.

Preface

This thesis is submitted at the Department of Information Security and Communication Technology at Norwegian University of Science and Technology (NTNU). This work constitutes the masters thesis for the MSc program in Communication Technology with specialization in Information Security. The duration of the thesis was 20 weeks, performed in the Spring of 2019.

I would like to thank my supervisor Chris Carr and responsible professor Colin Boyd for our meetings and their constructive feedback throughout the process of writing this thesis.

Trondheim, 6th of June 2019. Erlend Lid Helland,

Contents

List of Figures

List of Tables

Chapter 1
Introduction

Since Satoshi Nakamoto published his paper on Bitcoin in 2008, the narrative of Bitcoin being the future of online payments has existed. Through cryptographic operations, Bitcoin is able to facilitate financial transactions without the support of a central bank or governmental body. Using a trustless system to bypass the traditional banks and intermediaries who siphon off value is an alluring idea to many industries and actors. During the early years, this seemed within the grasp of Bitcoin, which transferred value with few problems while the pool of users was moderately small. However, as the user base grew, it became clear that the number of transactions processed by the system could not scale alongside it. Ensuring that all transactions get processed in a time frame that facilitates use as a currency is considered vital for Bitcoin to reach mass adoption. The usability of Bitcoin is greatly diminished if users have to wait for hours or even days before a transaction is processed. This limited throughput is directly tied to the underlying protocol, data structure and choices made in implementation.

The data structure used in Bitcoin and most other cryptocurrency systems is what is referred to as a blockchain. A blockchain uses a sequence of blocks to store arbitrary data, where the sequence stems from all blocks referencing their immediate predecessor through a pointer to the previous block. The sequential nature of this means that to remain a chain, only one block at a time can be added. This single point of entry is in contrast to graphs, where new nodes can be added at multiple points in the existing structure. This is one of the limitations that is causing the throughput issues we are observing in Bitcoin during high load periods. The system cannot handle multiple block inputs from different parties at the same time, without causing forks and thus wasting resources. In order to prevent forks, blockchains have to limit the rate at which data is added to a blockchain. This practice also limits the number of users the network can serve at any given time.

As a solution to this scalability problem, some of the newer cryptocurrency systems have tried to introduce a graph as the data structure for their solutions. This

move aims to solve the aforementioned problem of data throughput in the network being limited due to the weaknesses of a blockchain. The switch in the data structure is mainly targeted at addressing the throughput issues, while other protocol changes aim for improvement in other aspects of the cryptocurrency.

This thesis takes a closer look at this change in the data structure, but also looks at other choices that help facilitate this change. The scope extends to examine the rationale behind this change, the trade-offs made to implement it and the consequences it has on the operation of the networks. The thesis will also comment on the validity of the claims of these systems and if they achieve their stated goals in terms of performance. One of the main results will be a direct comparison between some prominent blockchain systems and DAG systems, as well as a discussion about the reasons for any discrepancies.

1.1 Motivation

It is becoming increasingly clear that no current blockchain implementation can handle the number of transactions needed to become a viable, real-world competitor to existing payment solutions. Many areas are being researched to solve this, with different actors looking into different solutions. Bitcoin has the lightning network, Ethereum has sharding, and many other systems are experimenting with changing the parameters of a blockchain to facilitate better scaling. One of the solutions currently being researched and developed to deal with this challenge is to develop cryptocurrency systems based on Directed Acyclic Graphs (*DAG*).

Bitcoin is by its design limited to blocks of 1MB and inter-block time of 10 minutes on average, which equates to roughly 5-7 transactions per second (TPS). DAG-based systems like IOTA and Nano boast of numbers in the thousands of transactions per second (for comparison purposes, Visa operates on average at 3,000 TPS and has a peak capacity of 56k TPS [26]). However, since these systems are new and less tested than a blockchain systems, there is still much doubt around their security, stability and consensus algorithms.

The issue of scalability is one of the most pressing challenges faced by any blockchain system aiming for mass adoption. There are also other hurdles on the road to adoption, as laid out by Meiklejohn [14], but scalability is the one most tightly connected to the underlying technology.

One of the main selling points of Bitcoin is its decentralization. Gervais, Karame, S. Capkun and V. Capkun [7] however, argue that Bitcoin is, in fact, more centralized than it seems on the surface. The research points to large mining pools, powerful exchanges and wallets as major centralization spots in the supposedly decentralized

currency. There is hope in the field that DAG-based systems can deliver genuinely decentralized transactions, as proposed by multiple projects mentioned in this report.

With the increasing worry about the environment in recent years, its also valuable to note that O'Dwyer and Malone estimated that the power consumption of Bitcoin mining already in 2014 was on par with that of the country of Ireland [18]. This consumption is in stark contrast to the claims of extremely low power consumption from some DAG based projects, due to their lack of mining.

Some of these challenges are intrinsic to a blockchain, while others are more coincidental. It does, however, appear that circumventing these obstacles is more achievable and practical in a DAG system.

1.2 Objective

The two main objectives of this thesis will be first, to answer the two research questions posed below, and second, to present a comprehensive comparison of scalability measures in blockchain and graph-based systems.

To effectively comment on the advantages of DAGs compared to a blockchain, it is essential to obtain a good understanding of the challenges and limitations of blockchains. Thus, one of the earlier objectives of the work will be to look at some metrics of the scalability issues faced by blockchains. This knowledge, combined with an understanding of DAG-based architecture, will aid in providing an answer to the first research question as formulated below.

Research question 1: Can DAGs be used to solve the scaling issues faced by blockchain based systems?

To answer this question, both the theoretical feasibility of the protocols and the actual implementation will be considered and discussed. The research, experiments and discussion aim to make informed comments on how the scalability of DAG systems compare to blockchain systems and the differences inherent between them.

As mentioned above, extra attention will be directed towards how the graph based systems differ in protocol and operation on a system level. As with every technological solution, an improvement in some regards will generally come from some trade-off in some other quality. The second research question will, therefore, try to investigate what trade-offs are made when switching the data structure from blockchain to a graph. This includes weaknesses in design, implementation, security or usability. These challenges are addressed through the research question as posed below.

Research question 2: What new attacks and challenges do DAG systems introduce?

In the current cryptocurrency ecosystem, the overwhelming majority of both networks and research are focused on the blockchain approach to distributed ledgers. This thesis aims at comparing the leading blockchain player in this space, Bitcoin, and the research around it to that of DAG-based systems. Comparing various approaches and how they differ, will help bring some understanding as to if and how cryptocurrencies can be enhanced by deploying a DAG as their data structure.

It is the intention for this thesis to give a comprehensive summary of what the problems of current blockchain systems are, which of these DAG could and could not solve, and what new challenges DAG-systems brings with it into this space. The thesis will also comment on which areas of DAG-based systems are most wanting of further research.

1.3 Scope and Limitations

This thesis will look at the most prevalent DAG based commercial solutions, as well as some proposed academic solutions. A natural limiting factor in this regard is that there are not many systems currently operating. For experiments and implementation discussions, the examples will mainly be limited to those from the Nano and IOTA systems. This is due to the academic solutions not being operated at the moment, even though this might change in the future with the DAGlabs initiative [38]. Byteball will not be considered for experiments unless Nano and IOTA are found lacking for this purpose. This follows from the pre-project where it was found that Byteball has a rather small and inactive community compared to Nano and IOTA.

For comparison purposes with blockchain systems, Bitcoin will mainly be used as a counterweight to these newer and less tested protocols. Bitcoin is chosen due to its name recognition, long history (relative to this space) and its market dominance. When comparing protocols in the abstract Bitcoin will be used as a representative of a blockchain systems. When making the direct comparison to blockchain systems, Ethereum will also be considered, since it is often regarded as the most advanced and scalable blockchain of the prevalent live networks. These two are chosen for comparison since they are widely known and have useful resources and documentation of operations going back multiple years. This makes a comparison of data points more reliable to carry out.

The work will also be limited in what aspects of the new protocols that will be in focus. Of particular interest is the technical aspects of the protocol which facilitates

the increased scalability being touted by the DAG systems. The DAG systems will be more thoroughly evaluated than the blockchain systems, due to their inexperience and the notion of new technology being more in need of testing than established technology. It is also worth noting that this thesis is restricted to considering systems that are proposed and known by the community. This seems self-evident but is vital to consider that the systems analyzed might not be the best solutions to the problems considered, especially given the short amount of time the DAG ideas have been around.

1.4 Thesis Outline

The main goal of this thesis is to compare and discuss different approaches to implementation of cryptocurrencies and in particular, the difference between a DAG-based and a blockchain based system. This includes a literature review, experiments on different networks and a comparison of the results gathered. The literature review will serve as an investigation into relevant systems, both in blockchain and DAGs, as well as the underlying technology for both. It will make it a point to not only look at whitepapers from the creators of the systems but also to supplement with published academic papers where possible. This source selection will generally be a challenge, due to the young age of this research field and the lack of published literature, especially on DAG systems.

After this initial task, the resulting knowledge will serve as a foundation for determining the metrics and data points most fitting for the comparison of systems. Then experiments to collect this data will be conducted, and results will be presented to serve as a basis for the discussion about trade-offs, pros and cons of the different approaches. Finally, a conclusion will be reached for the research questions and some areas for further research will be identified.

Chapter 2
Background

As the most widely deployed and used instances of cryptocurrencies, blockchain is still being regarded by many as the leading invention of this new technology space. The decentralized and secure nature of sending value between untrusted users has inspired a new shift in the online economy. In this new space of currencies based on cryptographic principles and functions, the most time tested and thus trusted players are generally those built upon the blockchain and in particular the Bitcoin system.

Being published in 2008 by the pseudonym Satoshi Nakamoto, Bitcoin has had the *first mover* advantage and is thus what all new players are compared to [17]. This chapter starts with a brief introduction to graph theory, which will be relevant for the DAG systems to be discussed. Following this, the basic concepts of blockchain in general and Bitcoin, in particular, will be introduced. The technologies and their challenges will be discussed, and key areas of problems will be identified. Building on the discussion of the technology and protocols, metrics for scalability to be used for later comparison between systems will be identified and justified. Data is gathered for these metrics from the Bitcoin and Ethereum systems, such that these can be compared to the DAG systems.

2.1 Graph Theory

Graphs in the context of this thesis refer to the graphs that are used in computer science as opposed to mathematical graph plots. Graphs have long been used as mediums of storage since their nature of relating different objects to each other has proven useful. The simplest graph G envisioned consists of two disjoint sets of edges E and vertices V. Edges are relations between two vertices, such that an edge x, y is said to join the vertices x and y. In a undirected graph, the edge x, y and y, x is exactly the same [4].

However, the use of graphs this thesis considers will only be directed graphs. This means that the edges x, y and y, x are *not* the same. One denotes a path from x to

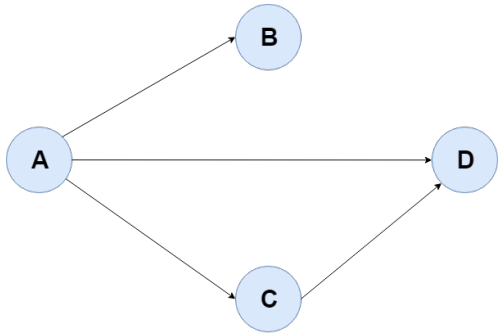


Figure 2.1: *Simple example of a Directed Acyclic Graph. It can be observed that there is no way of moving from right to left in this graph. Arrows on the edges indicated that they are directed, and can only be traversed in the indicated direction*

y, while the other denotes a path from y to x. This is especially important when one considers the second property graphs in cryptocurrency almost always posses, the fact that they do not contain any circuits. A circuit or cycle in a graph refers to the existence of paths in the graph that start and stop in the same vertex. It is easy to show that all undirected graphs have to contain circuits since one could walk back and forth through the same edge to end up where the path started. Combining these properties, direction and no circuits, results in a *Directed Acyclic Graph* or DAG for short. A simple example of a DAG can be seen in Figure 2.1 and a definition including the attributes discussed are laid out in Definition 2.1.

Definition 2.1. A graph $G = (V, E)$ consists of two sets: a finite set V of elements called vertices and a finite set E of elements called edges. Each edge is identified with a pair of vertices. If the edges of a graph G are identified with ordered pairs of vertices, then G is called a directed graph. A directed graph is acyclic if it has no directed circuits [25].

2.2 Blockchain Basics

The foundation for the blockchain is what is commonly referred to as *Distributed Ledger Technology* (DLT). DLT facilitates a consensus validating mechanisms across a network of computers or nodes that can be used to conduct transactions between peers without the need for an intermediary or central authority [15]. Transactions are validated by the network and added, along with a group of other valid transactions, to an existing chain of earlier transactions. These groups of transactions are referred to as *blocks* and the resulting chain as the *blockchain*. When a transaction is added to the chain, it can generally not be altered or removed. This is because all nodes in the network now have added the transaction to their copy of the chain. It is common

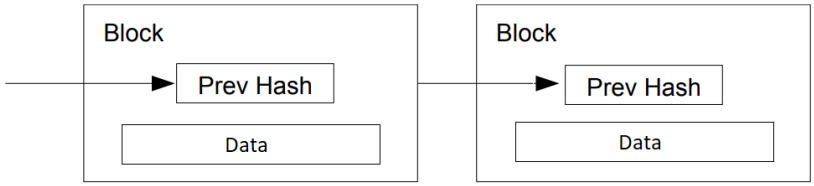


Figure 2.2: *An illustration of a general blockchain, where each block contains some data and a reference to its predecessor. This figure is a modified and generalized version from [17].*

to differentiate between two main types of chains: *Permissioned Blockchains* and *Permissionless Blockchains*.

- **Permissioned Blockchains** refers to proprietary networks that that specified clients or entities can access to perform transactions. These clients or entities are known, identifiable and given explicit permission to alter the state of the blockchain. This setup of blockchain is mostly applicable to business relationships, where participants require some means of identification, and when different users do not necessarily trust each other [27].
- **Permissionless Blockchains** refers to networks and their respective blockchains where anyone, regardless of identity and admittance, can participate in the network and thus change the underlying data structure that is the blockchain. This is the most common version of the blockchain and the one used for all current cryptocurrency implementations of some scale.

What is here referred to as *adding a block to the chain*, is in reality just adding the hash of the most recently added block of the chain (commonly referred to as the *tip* of the chain) into the new block that is to be appended. This makes it so that each refers to the block proceeding it, thus making a long chain from the most recent block back to the first block at the start of the chain. The first block is commonly referred to as the *genesis block*.

The blockchain is regarded as the most important innovation to come out of the cryptographic currency movement thus far. It is essentially a decentralized database spread across a network of cooperating computers that all take place in maintaining it. The cooperating entities do not have to trust each user to behave correctly, they only have to trust that more than half of all contributing users act morally. Due to its decentralized nature, it is almost impossible to alter fraudulently, since any malicious

agent would have to alter more than half of all local copies of the chain stored by the participants in the network. In a big enough network, this task is practically impossible. The blockchain as a *data structure* for a new system of decentralized, trustless transaction and storage is the key innovation. This leads to actors claiming that the blockchain will become a new layer to run protocols and applications on, like the internet. Blockchain can both be used to add a digital payment to the internet or to do public asset management of any sort [24].

2.3 Introduction to Bitcoin

Bitcoin, as described by Satoshi Nakamoto, was the first system to use a blockchain as the data structure in which every transaction are stored. Even though the blockchain stems from this work, it has been generalized and is today utilized in most of the cryptocurrencies that exist. This chapter will provide a quick overview of how the blockchain specifically is implemented into the Bitcoin system. This thesis will follow the convention in computer science where capital-B Bitcoin refers to the system, and lower -b bitcoin refers to the unit of account inside this system.

2.3.1 The Economics of Bitcoin

Grinberg describes the Bitcoin system as such: "Bitcoin is a digital, decentralized, partially anonymous currency, not backed by any government or other legal entity, and not redeemable for gold or other commodities. It relies on peer-to-peer networking and cryptography to maintain its integrity" [3]. Others have described Bitcoin as a communication protocol that facilitates the use of virtual currency [24]. While the latter description is maybe more correct, the earlier better captures Bitcoins primary usage at the time of writing and will, therefore, be the definition explored here to give a basic primer on some of the main features of bitcoin.

While on the subject on Bitcoin as currency, it is useful to note its similarities and differences to traditional currencies. Like the US dollar, a bitcoin is not by design redeemable for any other currency or commodity such as a given amount of gold. For non-digital currencies, this has been the norm for decades. These paper currencies that only rely on the public belief that governments and banks will prevent rampant inflation are commonly referred to as *fiat* currency. Before fiat currencies became the standard, public trust in a monetary system was obtained through assurances that a note of value could always be exchanged for some commodity like gold. This direct connection between currency and gold created confidence in the value of the currency since it was not trivial to conjure up more money and such decrease the value of all notes in circulation[83].

Bitcoin tries to circumvent the obstacles of both fiat and gold-backed currencies, by functioning as defined by known algorithms and supply and growth defined by rigorous mathematics. Instead of being governed by central banks or politicians, Bitcoin is managed through cryptographic rules that are enforced through computer code that is available to all. Individuals who own bitcoin do so by either running a Bitcoin client on their computer or via an account on a website that runs a client for all its users [8].

The economy internal to the Bitcoin network is based mostly on two mechanisms in the protocol which governs the interactions between clients. To incentivize the creation of new blocks by miners, a reward system is embedded in the protocol. The first transaction in a bitcoin block has a recipient address that the miner can change to whatever he likes, most likely the miners own address. This transaction will pay out some predetermined amount of bitcoins when the block is accepted into the main blockchain of the system. At the time of writing, this reward amount is set at 12.5 bitcoin, and this process of creating new coins out of nothing is referred to as *bitcoin minting*.

The reward in bitcoin is set to decrease by 50% for every 210 000 blocks created. Given only this incentive for miners, it is clear to see that the game theory optimal solution is for miners to only create empty blocks without transactions to minimize the amount of work they have to carry out in order to receive freshly minted bitcoins. To prevent this situation where no transactions are included, rendering the system useless, there is also another incentive on the transaction level of the system. Almost all transactions contain some funds allocated towards a *mining fee*, claimable by whoever includes the transaction in a valid block. The technicalities of both the fee and the mining process are discussed in the upcoming chapter.

2.3.2 The Technology of Bitcoin

Bitcoin, as mentioned earlier, utilizes a chain of blocks appropriately named as the *blockchain* as its distributed data structure. It is inside these blocks that all of the transactions of the system is stored since it first started operating in early 2009. The Bitcoin blockchain follows the general structure illustrated in Figure 2.2, where the data stored are the transactions contained in each block. These transactions, the rules governing how blocks of transactions are constructed, verified and broadcasted together with the rules concerning the creation of new bitcoins, are the most relevant parts of the bitcoin system in the context of this thesis. The creation of new bitcoins and the economic incentives for participating in the system were discussed in the previous chapter and will this will therefore not be rehashed here.

Transactions are one of the essential components of any digital currency system. They enable users to transfer value between each other, without which a monetary

system is rather useless. A typical transaction in the Bitcoin network contains one input of bitcoins and one or more outputs of bitcoins. One important technical aspect to note here is that the whole input of the transaction has to be spent. Whatever amount of currency is left after the outputs are subtracted from the input, is considered by the protocol as a fee left for the miners who first include the transaction in a valid block and add it to the chain. To avoid having all the remaining currency being spent on fees, the sender can include its address as a recipient, sending any remaining funds back to its own account.

A vital component closely related to transactions is the blocks forming the actual blockchain, as mentioned in the introduction to this chapter. In the original Bitcoin protocol, these blocks are limited in size never to exceed 1 megabyte in total size. It is interesting to note that this was not originally the case, and this limit is thus never mentioned in the white paper. Satoshi introduced it later in code changes, but made no mention of it in his commit messages which hint at him not wanting to draw attention to this detail [20] [21]. This block size is the main factor limiting the *transactions per second*(TPS) that the Bitcoin network can process. TPS for Bitcoin can be calculated by taking the block size (1MB or 1 000 000 bytes) and dividing by the average transaction size we expect. This gives the number of transactions that can fit into one Bitcoin block. Then one has to adjust for time since Bitcoin on average produces a block every 10 minutes, and this thesis is interested in transactions by the second. To account for this, divide the number of transactions in a block by the time it takes to produce the block. This operation is demonstrated in Equation 2.1.

$$\frac{1\ 000\ 000 \text{ bytes}}{\text{Transaction Size} * 10 \text{ min} * 60 \text{ sec}} \tag{2.1}$$

As observed from the blockchain, the smallest transactions to be recorded in the network are 62 bytes in size [80]. Applying the equation above, one can then find about 27 TPS as the absolute highest the Bitcoin network can currently deliver. This number is necessarily not realistic since most transactions are far larger than this. Using a more realistic transaction size of 250 bytes per transaction, it is shown that Bitcoin can deliver up to about 7 TPS.

Construction of new blocks is also a key point of contention both in the Bitcoin protocol and the discussion of blockchain problems. In Bitcoin, new blocks are generated by members of the network referred to as *miners*. Miners compete to solve a puzzle to prove that they did some work, in the original white paper this is called *Proof-of-Work* [17]. In Bitcoin, the puzzle to be solved is to construct a block such that the hash of the block starts with some given number of zeroes. This can be achieved by the miner tweaking the contents of the block, hashing to check if it solves the puzzle, and tweaking again if it is not a valid solution. The process is repeated

for all miners until someone finds a valid block, and is referred to as *Bitcoin mining*. In later discussions about the energy consumption and computing power required to maintain the Bitcoin network, this process of mining will be the critical point of consumption. After miners have proved that they have done this work, they can attach a valid block to the end of the chain. Miners are at full discretion to include whichever transactions they choose in this block.
When a miner finds a new valid block, it will broadcast it to all active nodes in the Bitcoin network. It is essential that this block propagates to all nodes well before the expected creation of a new block, 10 minutes later. This is because if another node creates a block without knowing of the latest block produced, a fork is created in the chain. This ultimately leads to one of the blocks being orphaned and thus useless, being a waste of both time and resources for the network.

2.4 Challenges Faced by Blockchain

Bitcoin have in later times come under heavy fire from critics of the system, who claim that the network cannot deliver on key aspects of becoming a real-world alternative to neither fiat currencies nor payment solutions like Visa and Paypal. This chapter will discuss some challenges and limitations to Bitcoin, where some are related to blockchains in general while others are specific to the Bitcoin system. Most of these challenges can be viewed in the light of what is called the *blockchain trilemma*, which essentially claims that a blockchain cannot be both sufficiently **scalable**, **secure** and **decentralized** [32]. Both Bitcoin and most other blockchain based systems seem to sacrifice scalability to gain security and centralization(even though this is also becoming questioned lately due to the rise in prominence of mining pools).

2.4.1 Scalability

The biggest problem standing between Bitcoin and mainstream adoption as a means of payment is the issue of scaling to the task at hand. If Bitcoin wants to capitalize on its position in the cryptocurrency market and become the go-to digital currency, it has to deal with its huge issue in regards to scalability. Mentioned previously was the fact that Bitcoin can at most achieve about 7 TPS in its current design, as can be devised from Equation 2.1. For comparison purposes, Visa operates on average at 3000 TPS and has a peak capacity of 65 000 TPS [77] [26]. This huge disparity has to be lowered, if not completely shrunk, for Bitcoin to have a piece of the global retail market.

The scalability concerns are shared by many, as noted by Sarah Meiklejohn, a prominent actor in this field, who included it twice in her list of top ten blockchain obstacles article [14]. She discusses how scalability can be applied both in the sense of transaction throughput and in the sense of storage as to not place an unbearable

Scalability Metric	Bitcoin	Ethereum
Max Theoretical Throughput	7 TPS	27 TPS
Average Measured Throughput	4.04 TPS	6.16 TPS
Maximum Measured Throughput	4.92 TPS	15.62 TPS
Minimum Theoretical Delay	300 seconds	7 seconds
Median Measured Delay	507 seconds	26 seconds
Time to Stability	60 minutes	3 minutes
Node Storage	200GB	150GB
Initial Download	200GB	2,200GB

Table 2.1: *Scalability metrics with values from Bitcoin and Ethereum. Explanation of metrics and values follows at the end of Section 2.4.1.*

burden on the users of the system. Meiklejohn argues that the biggest hurdle of scaling blockchains is the insistence that every node in the network needs to agree on the full state of the ledger. This means that the system does not scale if more computing power joins, neither in terms of processing transactions(throughput) or the time users must wait until their transaction is included(latency).

Storage is, as mentioned earlier, increasingly becoming an issue due to the growth in data stored on the blockchain. Currently, the size of the blockchain in Bitcoin is about 200GB, and growing with a projected 144MB every day. The size of the storage requirement is at odds with the idea of all participants in the network storing their copy of the distributed ledger to facilitate integrity. This integrity measure requires that no entries can be deleted from the blockchain ledger.

In table Table 2.1 some metrics for cryptocurrency scalability are laid out. Here follows a quick summary of what the different metrics represent and where the numbers are gathered from.

Max Theoretical Throughput is used as the maximum mathematical transactions that can pass through the network per second. For Bitcoin, this is calculated through Equation 2.1 explained earlier in this chapter. For Ethereum, Gas has to be accounted for since it is the limiting factors in Ethereum blocks. The block gas limit is about 8 million, a transaction costs about 21,000 gas, leaving the result at about 380 transactions per block [44]. This is a maximum since everything other than simple transactions costs more than 21,000 gas, thus leaving far less room for

transactions in a block. 380 transactions at an average block time of 14 seconds come out to roughly 27 transactions per second [45]. This metric is somewhat misleading for Ethereum since it is not a strict monetary currency like Bitcoin. Where Bitcoin only wants to be a store of value, Ethereum aims to be so much more, with a focus on smart contracts. In practicality, this can be seen through the metrics where Ethereum will generally perform worse against the theoretical limits for simple transactions since logic and contracts are far more resource expensive to carry out.

Average Measured Throughput. To measure this metric, two weeks preceding the writing of this was chosen for both systems, from the 11th to the 25th of February. The total amount of transactions confirmed into these networks were then divided by the time passed to find the transactions per second. For Bitcoin, this comes out to roughly 4.04 transactions per second and is half of the theoretical maximum [30]. For Ethereum, there were about 6.16 transactions per second in the chosen two-week span [46].

Maximum Measured Throughput. To give a sense of what the network is capable of in real-world conditions as opposed to the calculations, it is useful to look at the highest rate of transactions the networks have been shown to handle. It was chosen to look at 24 hours both to ensure that the traffic was sustained and not some unexplained anomaly and due to the availability of resources. Since this is a metric that extends back in time, historical data from block explorers were used. On the 4th of January 2018, Bitcoin had 425 008 transactions in the 24 hour day, resulting in an average of 4.92 TPS over the time period. This is the highest throughput recorded in the Bitcoin network on any single day [35]. On the same day, the pool of transactions waiting to be included was about 85 000 and thus not the limiting factor of the throughput [70]. This could hint towards 5 TPS being a more realistic upper bound for Bitcoin throughput. The same day, the 4th of January 2018 Ethereum had its peak of recorded throughput also. This amounted to 1 349 890 transactions in a 24-hour span, resulting in 15.62 TPS sustained average throughout the day [36].

Minimum Theoretical Delay gives the minimum average time from transmitting a transaction to the network until it appears in the data structure of the system. In Bitcoin and Ethereum this would be on the blockchain. If one considers the fastest case, where the sender of the transaction chooses to pay a high fee and the traffic in the network is low, this can be assumed to be half of the time between blocks in the system. Assuming transactions arrive randomly in the 10 minutes between blocks, half will arrive before 5min, and half arrive after, averaging out to the middle point of the possible interval. This results in an average theoretical delay of 5 minutes. The real world case of this is far more complex, having to account for both the choice of fee from the sender that the amount of transactions waiting to be included in a block (This pool of waiting is what commonly referred to as the *mempool*). Five

minutes is thus a lower bound for the average delay in the system. The tale is the same for Ethereum, where blocks on average are created every 14 seconds, leading to a theoretical delay of 7 seconds [45].

Median Measured Delay Where the above mentioned metric gives the average minimum theoretical delay, this metric aims the actual delay in the live systems. This is assessed through the median measured delay from a transaction is posted to the system until it is included in a block. Even though, as stated above, this is dependant on many different factors, it is still useful to look at data from systems that are operating. In Bitcoin, this median delay is 507 seconds, for the last two weeks of transactions. This is not too far off the lower limit in terms of seconds, but relatively it is still 65% over this minimum estimate. This value was calculated from the two-week median preceding the 1st of April 2019 [62]. In Ethereum, the median delay for transactions was estimated to be about 26 seconds, in the last 1500 blocks preceding the block 7481836 on the 1st of April. This amounts to about 11 days worth of blocks [43]. The relative difference is quite big at about 370% over the minimum calculated delay.

Time to Stability measures the time until a transaction submitted to the network is considered immutable and thus trusted by the community. This is a useful metric since transactions can be deemed invalid after they are transmitted due to conflicts like double spends. This is what one would consider as security in other systems, but as explained later, nothing is ever completely secure from being overturned in these distributed systems. Merchants need to be able to trust that a transaction is final before they will release goods that were bought with the transaction. This time interval is not something that is set in stone in any of the systems, but more general guidelines based on best practice utilized by current actors. The default for the classic Bitcoin client is six blocks deep in the chain, while exchanges and merchants often require more, especially for more valuable transactions. With a 10 minute delay between blocks, this results in 60 minutes of waiting as a rule of thumb. In a blog post on block times and security, Ethereum founder Vitalik Buterin suggests that a three-minute stability requirement on the 14 second block time blockchain is equal to the Bitcoin 60 minute stability on a 10-minute blockchain [47].

Stability is a less rigorous metric than others since there is no way to 100% guarantee that a transaction is secure. Given an attacker with sufficiently large mining power, no transaction is safe since the whole history of the chain could be rewritten. Stability is still a useful metric since it can say something about the probabilities of a transaction getting reversed depending on the power of the attacker and the depth of the transaction in the chain. This, in turn, means that different actors in the ecosystem can choose different levels of risk they are comfortable with.

The tradeoff here is then between the confirmation time users have to wait and the security level that the merchant deems appropriate. In Bitcoin, most smaller actors will accept about three block confirmations, while more prominent entities like exchanges will generally require six blocks of confirmation.

Node storage is a metric that shows how much permanent storage space is currently required to run a full node in the network. In many networks, there are options for running more lightweight nodes, but full nodes are chosen since they contribute the most to and are essential to the health the network. Almost all of this space is used to store the data structure containing the transactions of the system. For Bitcoin, this means that the whole blockchain has to be stored, amounting to roughly 200GB of data [28]. Pruning of the Bitcoin blockchain is possible, but due to limiting protocol messages, running a node in pruning mode will result in it not relaying transactions. This is due to there not existing a way for the node to communicate that it only possesses the last two days of blocks [75]. Due to this, the full blockchain storage size is chosen. For permanent storage, Ethereum only needs to store the most recent account state of all accounts in the network. This means that even though its blockchain is far larger than that of Bitcoin, a node only needs to store about 150GB permanently [41].

Initial Download describes the size of the download needed to initialize and start a node in the network. This download has to be done in order to obtain and verify the whole ledger of earlier transactions in the system. This is crucial since one of the selling points of cryptocurrency is its trustless natures, thus requiring us to check all earlier transactions for ourselves. In Bitcoin, this requirement in self-explanatory since it is the same as the storage requirement. In Ethereum on the other hand, the discrepancy is very noticeable. This is because pruning the Ethereum blockchain to keep its storage requirements down does not affect the download size at all. If one were to download a pruned version of the blockchain, it is impossible for the node to know if all earlier and now forgotten transactions were valid and legitimate. Thus the node needs to download the full Ethereum blockchain totalling about 2,200GB of data [42]. The node can prune this down while it downloads such that it will never use the full 2,200GB size on disk.

2.4.2 Decentralization

Decentralization is commonly touted as one of the greatest advantages of Bitcoin when compared to other monetary systems. The thought of a single point of failure or a single governing body is something many actors with libertarian tendencies have questioned for a long time. This stems from a plethora of ideas, like the distrust of a central bank regulating the minting of a new currency, the thought of rampant inflation in times of political instability or a governments power to freeze funds if they

see it fit. Decentralization in cryptocurrencies can be framed a few different ways, but this thesis will generally limit itself to consider mainly two forms of decentralization: governance- and computational-decentralization.

Decentralization of governance refers to how the system is governed, meaning who can introduce new rules into the protocol and change the existing rules the network operates under. Meiklejohn points out that, in a decentralized system, "who makes the rules matters at least as much as who enforces them" [14]. The behaviour observed in a multitude of blockchain system seems to indicate that even decentralized ones tend to become centralized when governance is concerned. In Bitcoin, for example, significant parts of the community are very hesitant to change anything originating from the founder, Satoshi Nakamoto. This became clear when the debate about block size was ongoing for years and is still not settled at the time of writing. Ethereum could be considered even more centralized, as almost all governance is carried out by a core set of developers and the Ethereum foundation.

The second mode of decentralization of interest to this thesis is what is referred to as computational decentralization. In Bitcoin, this is manifested through a few different mechanisms. The decentralized ledger and each users ability to validate transactions based on the history observed is undoubtedly a core element of the system. The more concerning property of computational decentralization found in Bitcoin today is the ability for any one user to contribute to the network meaningfully. Contributing to the operation of the Bitcoin network generally means to create blocks and verify transactions. Early on in Bitcoin, when the difficulty of producing blocks was much lower, a single user could easily create multiple blocks on their own and broadcast it to the network. Recently, with the exponential increase in computing power(commonly referred to as *hash rate*) both available to the network and needed by it, this is no longer the case [29].

To have any meaningful chance of solving a block and reaping some reward, users have to join big, centralized mining pools. Mining pools function by letting nodes join the pool and then split up the searching space for the next block into smaller segments and letting different nodes search in these sub-spaces. If a node finds the correct answer, the reward is shared between all nodes in the mining pool. This effectively reduces the *variance* of the expected return of mining bitcoins. As of the time of writing, the four largest mining pools control roughly 53% of the total computing power in the Bitcoin network. The centralization of mining has earlier been found to increase over time, since the beginning of mining in 2009 [2].

This centralization in computing power is viewed by many as an increasing worry for the health of the network [2] [16]. Both due to Bitcoins reputation as a decentralized network of equal peers, and the constant threat of 51% attacks to alter

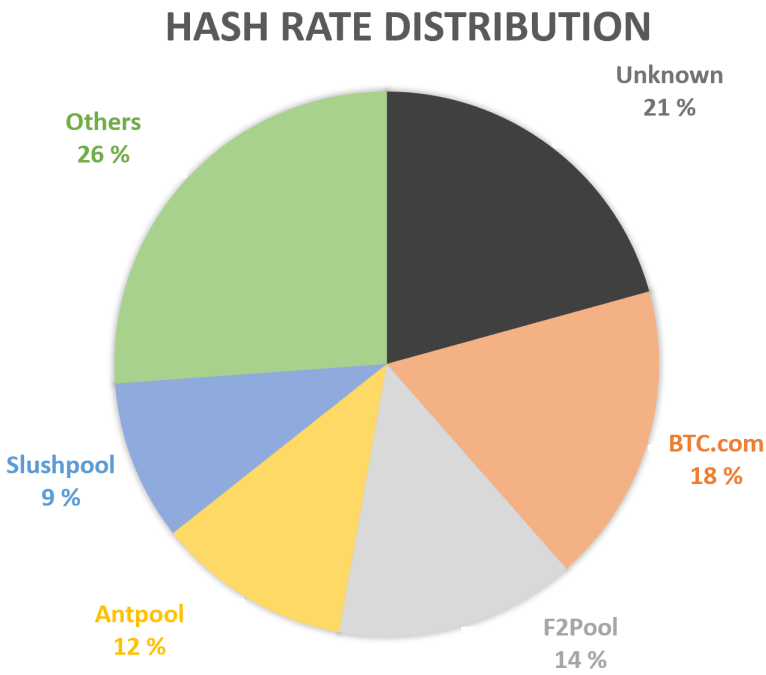


Figure 2.3: *Distribution of hash rate in Bitcoin from the four most recent days as of the 6th of February 2019. It is observed that the four largest mining pools control 53% of the total hash rate of the network. Data from Blockchain.com [50].*

the distributed ledger. It is beyond doubt that the organization of miners into these pools consolidates an inappropriate amount of power to the coordinators of said pools, giving them considerable influence over the system. This somewhat relates to the problem of governance discussed earlier in this chapter.

2.4.3 Resource Utilization

Critics have labelled Bitcoin and other proof-of-work based protocols as *environmental disasters* due to their huge power consumption and people routinely question whether using this amount of resources on running a network for payment is worth it [18]. This becomes especially true when one considers how few real-world transactions pass through the Bitcoin network. In 2014, when the network was smaller than it is today, it was estimated that Bitcoin consumes as much power as a medium-sized nation like Ireland [18]. While many of the critiques of this aspect of Bitcoin probably are greatly exaggerated, it is still true that it would be beneficial if this goal of decentralized payments could be achieved in a cheaper manner [14].

Better resource utilization can be framed in multiple ways, ranging from fees in user transactions to the cost of running the system both in terms of individual nodes, and the system as a whole.Taking the micro view, as is applicable when nodes are concerned, both the processing power needed to meaningfully contribute to the network and the storage capacities required to store the data ledger needs to be considered as significant barriers to entry if they become to resource intensive. One can also make a case for bandwidth requirements and hardware cost as meaningful barriers, but at this time, they seem less of a concern. Proponents of Bitcoin could argue that the revenue from mining offsets all of these costs, but this does not hold for all cases and misses the point of having a low barrier to entry for users wishing to participate in the network. A low entry barrier is essential if it is desired that the network is decentralized. High entry cost is a step towards only big players being able to extend their influence over the network.

In a more macro view of the system, one is less concerned about the load placed on individuals but focuses on the total sum of costs for operating the system, wherever they are located. This is also a vital view since one could imagine of a network where the load on each node is less, but sums to the same network-wide costs due to a higher number of participants.

Chapter 3
Methodology

As described in the introduction, this thesis aims to make a comparison of different scalability metrics of blockchain and DAG systems, while also commenting on their key differences and individual weaknesses and challenges. To do this, it is first essential to gain an understanding of the systems that are to be discussed and then utilize this knowledge to point to relevant metrics for a fair comparison. This includes a preliminary introduction to the system which will be considered and the underlying technologies that are most relevant to this thesis. When this has been introduced, an investigation into available tools will be conducted. For this project, the tools mostly consist of node software for the different networks, being run on virtual servers rented from hosting providers. These tools will enable experiments on the networks, which facilitates the gathering of data points relevant to the metrics identified.

3.1 Trustworthiness and Reliability of Sources

Due to the nature of the cryptocurrency ecosystem, where many actors with different motives and a great deal of money are involved, there are bound to be people trying to take advantage of others. With the increase in new currencies launched in 2017 and 2018, several outright scam projects were released. The best example of this was the Bitconnect project, which in reality was a Ponzi scheme where the founders siphoned off the money and suddenly closed down the project in January of 2018 [64]. Examples like this, along with many others of proponents of different systems who hype up their project and lie to increase its valuation, make it clear that many sources in the cryptocurrency space are not to be trusted without corroborating evidence.

This section aims to introduce some level of scepticism to the trend of self-published white papers that are seen in the cryptocurrency space. While there are some advantages to this method, mainly swift releases and broad access, they may be outweighed by the negative aspects.

The biggest problem with this method of publishing is that there is no structured peer review process. Since the paper is public, one could argue that everyone can read and critique it, but this is countered by the fact that it is easy for legitimate criticism to get lost in all the noise. A structured peer review by a legitimate publication with qualified peers never take place, and thus, the conclusions and claims in the paper are never scrutinized adequately. This, coupled with the fact that the team producing the white paper is almost always financially involved in the system, leads to a culture where scams, plagiarism and half-truths are a part of the norm.

This lack of peer-reviewed publications in the field is certainly a weakness, but there are some mitigating factors in play as well. The biggest one of these is the fact that almost all systems in this space are open source, and thus, the reader of the papers can easily go and investigate to see if everything is as presented. The Coordinator in IOTA is the exception to this rule, as it is closed source, and no one outside the project knows how it works. This enables every user to look at the code, download and test it, and discuss it with others. After all the attention this space has garnered over the last year, more people are critical of new projects and claims, and scams are quite commonly called out. One example of this kind of 'public' peer review is when Heilman et al. [10] published an article criticizing the choice of IOTA to use their own hash function, known as *curl*.

The absence of published literature, especially on DAG-based systems, also speaks a lot to the fact that this technology is still very new and emerging. While Bitcoin has matured and has been studied for about ten years now, the first DAG based projects were not even proposed until late 2015. This will hopefully be remedied over time, but at the moment, most of the information about these systems comes from studies of the actual protocols and information directly from the founders and developers working on the system.

Since the creators of the currencies generally control different publishing platforms for their product, it could make it harder to get through with legitimate criticism. This was especially true in the early days of the cryptocurrency space when everything was announced on the Bitcointalk forums (This is still true to some extent, although a lesser one). If one were discriminated against or silenced on this central forum, the outreach and thus the exposure of the project was severely limited.

3.2 Nano Mainnet

To get a better understanding of the network, and enable experiments on them, it is useful to run a node participating in the execution of the protocols on the network. As earlier described, Nano nodes in this network mainly only participate by recording and rebroadcasting most transactions or blocks. Representative nodes have a higher

```
root@NanoNode:~/Nano# stat data.ldb
  File: data.ldb
  Size: 14909755392     Blocks: 29120624   IO Block: 4096   regular file
Device: fc01h/64513d    Inode: 270569      Links: 1
Access: (0600/-rw-------)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2019-03-07 11:47:22.785715046 +0000
Modify: 2019-03-11 14:39:55.785263170 +0000
Change: 2019-03-11 14:39:55.785263170 +0000
 Birth: -
```

Figure 3.1: *Size of Nano ledger on disk of the node after it caught up on blocks from the main network.*

workload, due to also having to participate in consensus voting. The Nano main network and its corresponding reference node implementation will be the main tools used for data gathering and measurements in the Nano network. This data will later be presented, in Chapter 5, and be used for a scalability comparison between DAG and Blockchain systems.

To run a node that can both propagate transactions and vote on consensus, a digital server was rented on the cloud hosting service DigitalOcean [39]. Initially, the option with the lowest computing power was chosen, although this can easily be upgraded if necessary at a later point. The chosen hardware configuration contains 1 GB of RAM, 1 CPU core, 25GB of SSD storage and 1TB of bandwidth use each month [40]. Initially, this seems reasonable, given that the entire Nano ledger at this point is about 14.9GB and bandwidth is the most demanded resource in the system.

To set up a Nano node, a tutorial outlining the process was followed [51]. The setup is based on a Docker container with a Nano node running inside it. This setup is provided officially by the Nano foundation and is open source like the rest of the project. The setup process was swift and easy to conduct. After the initial configuration is complete, the node needs some time to download and check all blocks in the ledger. It is now possible to interact with the node through an RPC interface which the developers describe commands at [66]. Communication with the interface is via HTTP with a payload of application/JSON data that contains a JSON object with the commands issued. For this purpose the utility *curl* is used [37].

To make the node able to vote in the network, a wallet and an account for our node have to be set up. First, to create a wallet, one can issue the *wallet_create* command. This returns the randomly selected wallet ID: 'E5F953B1A4E8219AEB512B08CD47989569ACF68162C028177D1AFBF17AA38145'. When nodes are concerned, the wallet can almost be viewed as a person. This is because like in traditional banking, a person can have more than one banking account, a wallet can contain several Nano accounts. To make an account tied to a wallet, the *wallet_create* command was used:

```
{
"action": "account_create",
"wallet": "E5F953B1A4E8219AEB512B08CD47989569ACF68162C028177D1AFBF17AA38145"
}
```


with the response:


```
{
"account" : "xrb_3k1fi8ghmupjxsw9but7a596f1x6jykf1ya9o8c5tgnnoaaf1sjmmj8hz85t"
}
```


To verify that the wallet and address is working as intended, we send 1 Nano from an online wallet account to the newly created account. The online wallet account is hosted at *https://nanowallet.io/*, and the transaction can be viewed at [67]. To make the node a representative a *change* transaction, as explained in Section 4.1.1 is needed. Issued from the online wallet, this would make the node able to vote with the weight of the online wallet as if it controlled this currency. The node however does not have spending power of the currency stored in the online wallet.

To change representative the *account_representative_set* for the RPC interface is used as displayed below.


```
{
"action": "account_representative_set",
"wallet": "E5F953B1A4E8219AEB512B08CD47989569ACF68162C028177D1AFBF17AA38145",
"account": "xrb_1og6z68drckx3nituwgd5e6h9ufqx6bcxpkok15u58pu35wgjh7ie7ga31dc",
"representative" : "xrb_3k1fi8ghmupjxsw9but7a596f1x6jykf1ya9o8c5tgnnoaaf1sjmmj8hz85t"
}
```


When these transactions are broadcast and accepted by the network, the result is that the node can vote with a weight of 397 Nano, even though it only possesses a single Nano. It is now ready to both participate in spreading transactions as a node and help achieve consensus as a voting representative.

3.3 IOTA IRI Main network

In order to set up an IOTA node, a community quick start guide was followed [53]. This guide explains the setup process of a full IRI node in the IOTA main network. IRI means that it follows the IOTA Reference Implementation outlined by the IOTA foundation [55]. The server used for this setup was of the 4GB RAM, 2 CPU cores, 80GB SSD Storage and 4TB of network traffic. The higher specifications of this setup are due to the IOTA node refusing to run on anything lower than 4GB of RAM. After installing the IRI full node client, the next step is to add neighbour nodes that

are part of the IOTA network. This step is necessary to connect the node to the main network such that it can receive and broadcast transactions.

To find neighbours to link to, the guide recommended posting a request in the official IOTA Discord channel. Many cryptocurrencies use Discord channels, which functions as a collection of chat rooms, for community interaction. In the IOTA channel, there are dedicated rooms for many features, one of which is dedicated to node sharing. After posting a request for neighbours running the IRI version of the main network node, it took about 24 hours until the recommended threshold of 4 active neighbours was met. Neighbours are important for a node since the only way to learn of new transactions in the network is through communication with them. When the connections are established, one must wait for the node to synchronize with the network state.

The backlog of transactions in the Tangle is quite large, and the node needs to discover these through its neighbours. This proved to be a more tedious task than first anticipated. 72 Hours after the initial node setup the node has still not caught up to the network. This is evident through the *LSMI (Latest Subtangle Milestone Index)* counter displayed in Figure 3.2. The LSMI counts the most recent *Milestone* issued by the Coordinator that the node has seen in the Tangle. A caught up node should not be far behind the official newest milestone index, which in the image is represented by the *LMI(Latest Milestone Index)* counter.

Roughly one week after startup of the node it became clear that the LSMI counter would never catch up to the LMI counter at the current pace. Over 48 hours (following the 72-hour initial period) the LSMI counter has increased by one and was still trailing by about 90 000 milestones. To try to remedy this situation, the hardware of the node was upgraded to try to facilitate better handling of transactions. The LSMI counter will only increase if the node has observed all transactions that the milestone in question refers too. The stagnation in this counter could therefore possibly stem from bandwidth limitations which would throttle the number of transactions the node receives from its neighbours.

After about three more days of waiting, the local milestone counter (LSMI) had still not moved. Asking around in the official IOTA discord and on the Reddit page of the project, several users suggested adding TCP connections to new neighbours instead of only using UDP as had been the practice earlier. After opening the firewall of the server for TCP connections, two new TCP neighbours have been added, bringing the neighbour connection count up to a total of eight.

One more week of waiting went by without the node gaining much on the network as a whole. After more research into this issue, users on the official IOTA discord pointed to the possibility of downloading a copy of someone other nodes database,

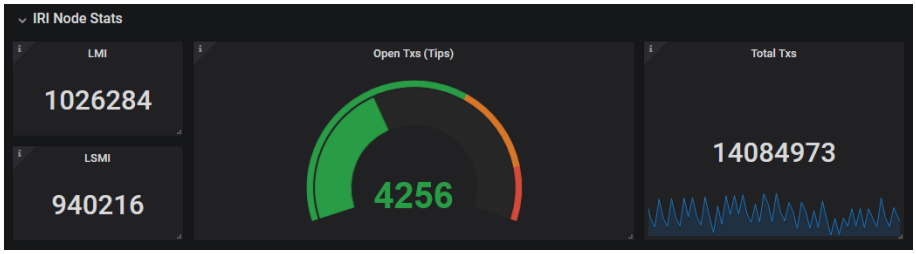


Figure 3.2: *Image from IOTA Node statistics after 72 hours of uptime. Graphs made with Grafana interface for nodes.*

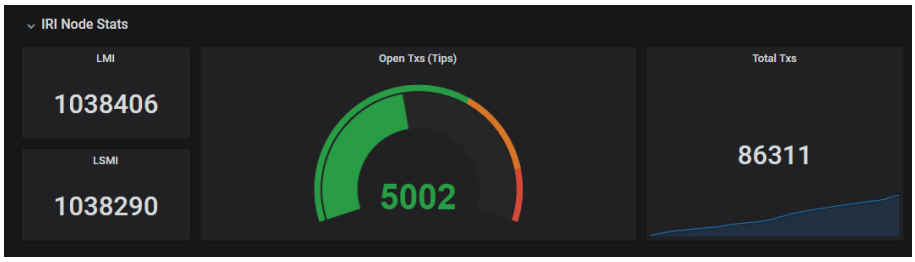


Figure 3.3: *Image from IOTA Node statistics after downloading external database snapshot and restarting the node. Graphs made with Grafana interface for nodes. Image captured on the 8th of April.*

that was closer to the network state than the local one used by the node used in this experiment. The reference implementation running on the test node has native support for downloading a recently synced database. Using this option, the LSMI counter went from being about 100 000 behind to only being about 100 milestones behind the latest released one. Again the node is left to its own devices to sync with the network fully. This state can be seen in Figure 3.3.

After this, the node was allowed to run for a period exceeding two weeks to catch up to the network and operate freely. This, slightly excessive, extended period was due to other tasks being worked on and Easter occurring. During this time, the node caught up to the network rather quickly and started operating in sequence with the network rather than always lagging behind and trying to catch up. It was in this caught-up state that the measurements described in the result chapter were obtained.

3.4 Experiments

In order to make a meaningful comparison between the DAG and blockchain systems, some experiments will be carried out on the DAG networks described in Section 3.2.

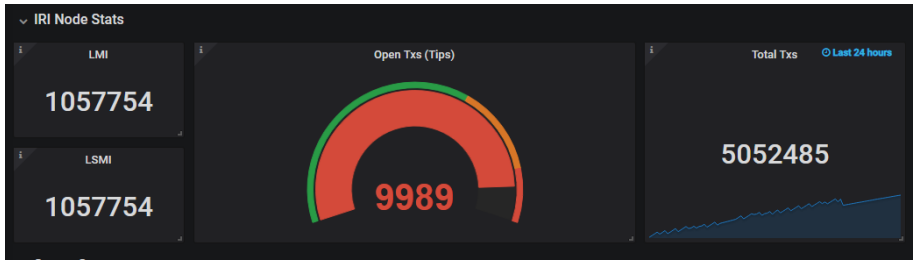


Figure 3.4: *Central node statistics after the node has caught up to the rest of the network. This shown by the LMI and LSMI counters being equal. Image captured in the 22nd of April.*

Experiments will be designed to gather information about the metrics described in Chapter 2 and will be ordered according to what subgroup of challenges they aim to address.

The experiments and studies of different live networks that are to be undertaken, align with what Wieringa describes as *single-case mechanism experiments* in his book *Design Science Methodology* [81]. Wieringa outlines this type of experiments as: 'A single-case mechanism experiment is a test of a single case in which the researcher applies stimuli to the case and explains the responses in terms of mechanisms internal to the case'. This kind of experiment can be done both in laboratory conditions or live in the field, depending on the purpose of the research. Wieringa goes on to claim that this kind of experiment is best suited for validating new technologies, evaluating implementations and investigating problems. This methodology applies to the questions that this thesis poses.

Other key traits of a single-case mechanism experiment are that a single object is studied, the architecture of the object is known, and the research goal is to describe and explain cause-effect behaviours of the object. In the case of this thesis, the object would be the different DAG systems and their architecture would be known to everyone through the open nature of these projects and the literary study. The outcome of this experiment is then a better understanding of its cause and effect behaviour under different circumstances.

This experiment is treated like any other scientific experiment, with research context, research problem and design of the experiment documented. It is vital for the integrity of the results that the experiment is carried out as initially described, and any anomalies or discrepancies are noted.

Due to limits in available community resources and general community activity, this work will only focus on the Nano and IOTA networks for experimentation.

Byteball will not be considered, mainly due to its lacking community support when it comes to operating in the network. Both IOTA and Nano sport thriving communities of both developers and volunteers producing content and answering questions about the technology. This can be seen both on their respective subreddits and discord servers. Byteball does not have such a large following and thus is harder to approach with questions.

3.4.1 Scalability Metrics in DAG systems

Experiments to asses the scalability of the DAG systems focus on taking measurements that can be used for comparison with the blockchain properties listed in Table 2.1. Here follows a brief discussion and summary of how to best assess and gather data for the different metrics. The result of the measurements in DAGs will be discussed later, in Chapter 5 of this thesis, which deals with results.

The metrics at hand compare both theoretical limitations and differences in measured values to shed light on different aspects of the systems. Theoretical values are most valuable when investigating the intrinsic limitations of systems, stemming from design decisions. This differs from limitations caused by outside factors and technologies. Measurements tell a compelling story of how design and theory behave in a real-world implementation and is therefore essential to include for meaningful comparisons.

Max Theoretical Throughput

Max throughput would be extremely hard and expensive to test in these systems. Given that they both advertise throughput in the thousands of transactions per second, the experiment would have to produce a massive amount of traffic. This would, in turn, require a computational load from the small PoW in the systems, which in aggregate would be a task too computationally expensive for almost any academic endeavour. Due to these circumstances, the max throughput is calculated instead of measured, much like what is done for Bitcoin and Ethereum.

Average Measured Throughput

Where the nodes are working as intended, this metric can easily be measured by noting the transaction counter at different times and thus calculating the observed TPS. The time slot being observed Third party explorers are used to verify the accuracy of the measurements. It is useful to note that this would only give the transactions that the node set up for the experiment see in the network, but this assumption is reasonable given that in a functioning decentralized network, all nodes should see every transaction.

Maximum Measured Throughput

These measurements are based on historical data. Some of the DAG systems have rather poor data gathering in place to do such measurements, which limits the scope that can be assessed through this metric. Compromises have to be made based on the availability of data, such as only considering the last X number of weeks. This can limit the usability of the measurements in a meaningful way since the intention is to investigate the discrepancy between claims and proven strength. If the measurements of the peak strength displayed by the networks cannot be accessed, this could negatively affect their comparability. Stress tests which show shorter bursts of higher throughput will also be considered.

Nano has historical data on TPS going back to January of 2018 with granularity down to individual days on their official online explorer [31]. From this, it can gather the maximum amount of blocks that were added to the ledger in a single day and thus calculate the TPS from this data. It is undoubtedly a point open for discussion whether TPS or individual actions on a network is the fairest basis for comparisons. For Nano, this is especially pressing, since it is the only network considered here where a transaction requires two distinct actions to be finalized. This could be argued both ways, especially when considering the fact that after a *send* block has been generated, the transaction is completed from the viewpoint of the sender. He or she has no way to ever reclaim this transaction, even if the receiver never accepts it. This could lead one to conclude that only a single action is required from the sender to carry out a transaction. On the other hand, the receiver must construct a *receive* block to claim the transaction and be able to use the funds at a later date. Since this is what is generally referred to as a transaction in other systems, it is deemed most fair to treat the latter as a fully executed transaction.

IOTA has a worse historical record of their transactions, going back in time. This is probably due to their technique of *snapshots*, where nodes agree on a network state and then treat this state as a new starting point for the network history. All database entries older than the snapshot is thus deleted, such that the ledgers of the nodes do not grow out of proportions. The unfortunate downside of this is that it is hard to gather historical data going back for an extended amount of time. The best source that could be found in research for this thesis was a third party explorer that kept data going back three months [52].

Minimum Theoretical Delay

When dealing with DAGs and this metric it is important to both look at the systems now, and where they intend to be in the future. IOTA, for example, currently relies on their Coordinator for confirming transactions, but will at a later time solely rely on new transactions confirming older ones. Both modes of operation will be

considered in this work. This is one of the most meaningful and interesting metrics since it shows a clear picture of how the design of these cryptocurrencies impact critical traits of their usability.

Median Measured Delay

Delay is a slightly less tangible metric in DAG-based systems, due to the asynchronous nature of their operations. The definition of when to start and stop measuring becomes more unclear in asynchronous environments, as discussed in the section on maximum measured throughput. Median delay measurements give insight into how much a system differs from its promises on a critical trait of digital currencies, the measurement of how quickly a transaction is known to the network after it has been sent.

For measuring the median delay in Nano, the node set up in the network can be used. The RPC interface for this node can give the time elapsed for the last 2048 blocks voted over by the network [66]. To increase the chance of getting a representative result, it is advised to measure more than once, especially since the measurement only goes back 2048 blocks. Given a block creation of about 0.2 blocks per second, it would take about 10 240 seconds or about 3 hours for all blocks to be replaced. Based in this it would be advisable to do measurements at least 3 hours apart to ensure elections of blocks are not counted twice. To account for these parameters, at least five measurements were conducted with at least 5 hours separating them.

The IOTA node has better support for finding historical data on confirmation times since it lets the user define the time window to be considered when returning results. It is worth noting that this metric can change drastically going forward, since after the implementation of the open source coordinator called *Compass* on April 10th 2019. This has happened and passed, without much change in confirmation parameters the time of writing. In the future, this could certainly change, as the IOTA foundation has reported that it probably will. Considering the network is now relatively stable after this change, looking at averages going back about a week should suffice to get a reliable picture of confirmation times.

Time to Stability

Stability in DAG systems can mean different things from what is normal in blockchain systems. In blockchains with PoW, a potent attacker can always come along and construct its own longer chain that would, in turn, invalidate transactions submitted to the old original chain. In DAG systems this metric comes down to a much more system specific metric since stability and confirmation are done via different

schemes. Whereas in blockchain systems, this will generally always come down to if a transaction is added to the chain, and how many blocks deep it is from the tip.

Node Storage

In order to measure how much data storage is needed to run a node, nodes are setup and initialized to participate in their respective networks. The nodes were then allowed to run for about a week, to ensure that they were sufficiently caught up to the current state of the network.

Initial Download

To track the bandwidth usage of starting a new node, the tool vnStat was installed before starting the node initializing sequence [78]. This tool lets the user track all network traffic in the kernel, providing a lightweight option for bandwidth statistics. This metric is not exact as it is not entirely clear when nodes start participating in network activity and thus what traffic is downloaded for the startup of a node and what traffic stems from participating in the network.

Chapter 4
DAG based systems

This chapter gives a brief introduction to how DAG systems differ from blockchain ones, with a focus on which of these differences stem from the change to DAG as a distributed ledger over a blockchain. These differences between the systems as well as their causes and effects will be the main takeaway from this chapter. The information concerning these systems is mainly gathered trough the official white papers released by the respective organizations behind the cryptocurrencies. These white papers are highly commercial and produced by a team with a financial interest in the product, so they should not be trusted blindly. This is addressed through a section of source critique in Section 3.1. At the end of the chapter a brief summary of different security risks can be found. This serves to mention and discuss different possible security risks in the systems. This chapter is adapted from the pre-project submitted in December of 2018 at Department of Information Security and Communication Technology at NTNU.

4.1 Commercial Implementations

The following systems are live today, with main networks operating in a commercial setting. All of these have tokens that can be bought and offer some utility to current users. All of these live implementations are *Transaction DAGs*, meaning the graph contains only single transactions as vertices, with edges referring to earlier transactions [82].

4.1.1 Nano

Nano (previously known as Raiblocks) is a cryptocurrency first announced on the Bitcointalk forums in February of 2016 [12]. Nano claims to be a feeless, fast, scalable, simple and environmentally friendly cryptocurrency for everyday monetary use. Nano issues every account with their personal account-blockchain. When viewing all these account-chains in relation to time and each other, they form the Directed Acyclic

Graph that Nano terms as their *Block-Lattice*. The Block-Lattice is illustrated in Figure 4.1 [13].

Nano operates by letting each account owner have full control over their individual chain. This control manifests itself through that only the holder of a chains private key can add to the chain. Implied in this is that both the sender and receiver of a transaction must alter their own chains to reflect the value being transferred. The account notifies their representative node, which in turn alerts the rest of the network. Since the network is asynchronous, these two alterations need not happen at the same time.

To achieve consensus on what transactions are valid, Nano utilizes a distributed proof of stake system, where users can choose representatives to vote on their behalf. Voting power is assigned based on how many tokens a wallet is holding at any time. If any node discovers two conflicting transactions, they will alert the network and accept votes on which transaction to keep. The most 'popular' transaction stays, while the loser gets discarded by the network [13].

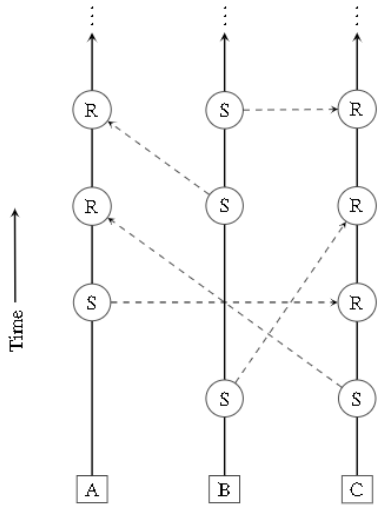


Figure 4.1: *Nano's DAG, the Block-lattice. A,B and C are individual accounts, nodes are either Send(S) or Receive(R) transactions. Figure taken from [13].*

The team behind Nano claims it to have *instant transactions, zero fees* and be *infinitely scalable* [73]. The first claim, instant transactions, stems from the sending process described in the Nano protocol. Only a small PoW to combat spam is required to post a transaction, and this PoW can be pre-computed before the transaction is to be sent. This means that there is virtually no delay experienced by the user when sending a transaction on the Nano network. This is backed up by the fact that

currently, the average confirmation time for a Nano transaction is 0.75 seconds [61]. This can be seen in Figure 4.2 where the most recent percentiles of confirmation times in the network are displayed. In this figure the p50 line means that 50% of transactions are completed within this time, same goes for the other percentiles. The second claim, zero fees, builds purely on design decisions. Since there are no miners that need to be paid, no fees are utilized in the Nano system. Some critics will claim that even a small PoW is a fee, but this seems unreasonable. Colloquially fee refers to some monetary value paid to middlemen or facilitators of some sort. A PoW does not fit this since no value is transferred to someone else. The third claim, infinite scalability, gets at the fact that nothing in the Nano protocol inherently limits its scalability. The only limits on Nano transaction times are outside the system, such as network propagation times and hardware read/write cycles. Moores law indicates that these obstacles will also scale given time.

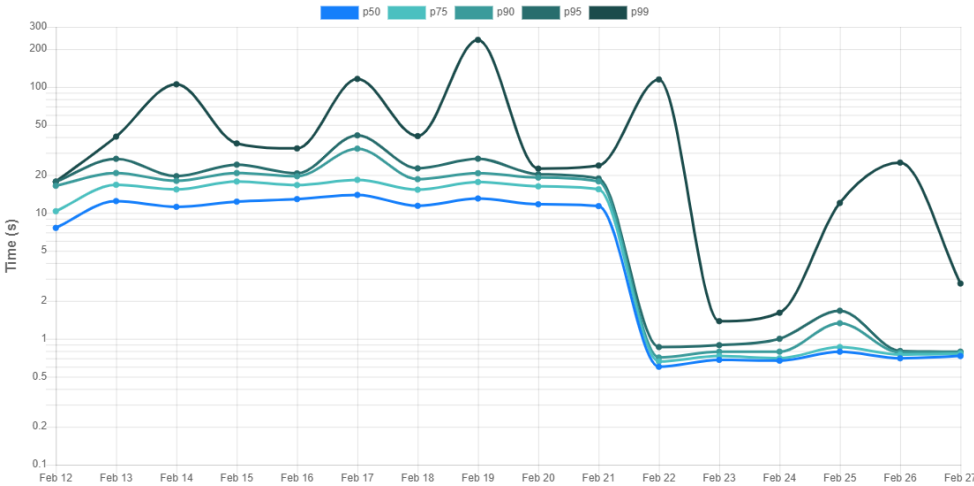


Figure 4.2: *Percentiles of transactions that fall under different confirmation times. At February 22nd Nano released their latest update leading to a significant increase in transaction speed. Figure taken from [61].*

The main differences in Nano that stems from the implementation of a DAG as the data-structure are the switch that can be done in regards control of the ledger and the asynchronous nature of the DAG. Due to Bitcoin having one blockchain in the whole system, all users have to operate on this single chain. This means that all users must have the same image of the chain, thus introducing propagation delay as a significant factor. This single chain design also severely limits scalability, even if one tweaks the parameters of Bitcoin towards a higher block-rate. As mentioned before, Nano gives all accounts their own blockchain, leading to more scalability and

autonomy over the user's transactions. Asynchronization means that propagation delay is less of an issue since users can update their chain at their discretion.

4.1.2 IOTA

IOTA and its main component called the Tangle was first announced on the Bitcointalk forums on 21st of October in 2015 [79]. It aims to be the go-to protocol for IoT solutions and enable machines to do micro-transactions with each other. To produce the scale and speed needed for this task, they introduce a DAG solution they refer to as 'The Tangle' [19].

To increase scalability, speed and throughput, IOTA replaces the traditional blockchain with a global Directed Acyclic Graph which contains all transactions ever recorded in the network. In this graph, the vertices are transactions, while the edges indicate verification of other transactions. Any transaction added should always verify two other transactions and preferably transactions that are not previously verified. Transactions awaiting verification are referred to as 'tips' in the graph, and the selection of these tips are done by an algorithm to prevent adversaries from trying to exploit manual tip selection [59].

As mentioned, for a new transaction to be allowed to join the graph it must first confirm two earlier transactions. Verification thus constitutes the Proof of Work used in the protocol. The PoW cannot be pre-computed since it depends on the network state, which could change at any given time. When a transaction approves another transaction, it also indirectly verifies all earlier reachable transactions in the verification chain that stretches to the Genesis transaction.

To protect the network from consensus attacks and double spends, IOTA is currently employing what they call a Coordinator to ensure the security of the network. The coordinator functions such that it puts out a milestone transaction about once every three minutes. All other transactions referenced by this milestone transaction, are considered confirmed. IOTA deems this necessary since it currently is rather easy to mount at 51% attack on the system due to its low activity. The team has harvested a lot of criticism for this feature and has promised several times that the coordinator will be removed when it is safe to do so. The coordinator thus limits the confirmation time in the system to at least the arrival of the next milestone.

IOTA wants to position itself as the default cryptocurrency used in IoT solutions for the future. This means enabling micro-payments in a machine to machine economy. To achieve this goal, they promise a cryptocurrency that is *highly scalable*, *consumes few resources* and *has zero fees* [56]. Scalability in IOTA is delivered through the choice of a DAG, the Tangle, as their data structure. This means that there is no time between blocks like in blockchains, but that users can post their transactions to

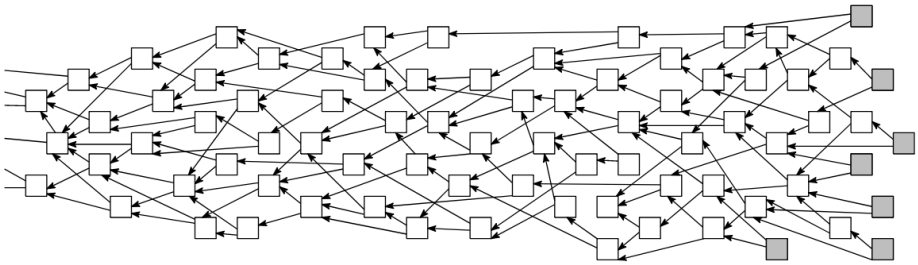


Figure 4.3: *The IOTA tangle. Gray boxes are transactions awaiting confirmation. Figure taken from [19].*

the network at their own discretion. The transactions of any given user are confirmed and validated by other users posting transactions, such that in theory confirmation times will go down as usage of the network increases. Currently, this is not the case, due to the use of the coordinator. The current PoW scheme of IOTA is reported to take about 30 seconds on a normal desktop computer, and doubles as an anti-spam measure.

This PoW is also one of the main challenges for IOTA to realize their goal of becoming a currency for the Internet of Things. The PoW scheme used in IOTA is currently less resource intensive than the one in Bitcoin, but in Bitcoin, the senders of transactions do not have to do the PoW. At this point it is unclear whether this difference is due to design decision in IOTA or if its up to the vast difference in number of users. Furthermore, if IOTA wants to break into the IoT market, there will be a lot of smaller devices that do not have the capacity to do any computations that are not part of their primary use case. Remote computations of PoW might solve this on behalf of smaller devices. Services like this currently exist, where owners of devices can pay to have their PoW for transactions computed remotely [58].

Coinciding with the ideal of zero fees, cryptocurrency is also a resource, and IOTA facilitates not wasting this resource by not charging any fees for transactions. This is essential for enabling machine to machine continuous payments in an IoT ecosystem. As in Nano, no miners are needed for the system to add new transactions, and thus no one is paid to maintain the network. Note that though the remote PoW mentioned above seems close to paid mining, but it is optional unlike transaction fees in other systems.

Integrating a DAG as their underlying data storage enables IOTA to use this user-driven verification model. This model could not be used in a blockchain since far too many works would occur if hundreds or thousands of users tried to create new

blocks at the same time. The DAG facilitates asynchronous updates in different parts of the data structure, while still maintaining order and not wasting many resources.

4.1.3 Byteball

Byteball was first announced on the Bitcointalk forums on the 5th of September 2016 by Anton Churyumov [6]. It proposes a solution where users can store arbitrary data in a decentralized database against fees proportional to the size of the data stored. Storage units are linked by containing hashes of earlier units. These references serve both as confirmations and as a way to create an ordering of the graph. These units and confirmations are abstracted as the vertices and edges in the DAG that is formed. Byteball uses an internal currency that is known as *bytes*. 1 byte currency can pay for 1 byte of storage in the database. This currency can also be traded with other users of the system. To issue transactions, users create storage units containing one or more messages. Such messages can then have the type *payment*, which contains details of the payment made.

As in most implementations of a DAG-based model, the Byteball network can also contain conflicting transactions at any given moment. To resolve these, the protocol relies on the ordering of transactions to determine validity. The simplest way to determine the order is if one of the conflicting transactions reference the other via direct hash, or implicitly such that it references a predecessor. In this case, it is clear the one referencing the other must be invalid since transactions only can reference earlier transactions.

When the two conflicting transactions do not have a path between them, Byteball turns to something it refers to as *total ordering*. First, the network chooses a *main-chain* that the rest of the transactions will be ordered according to. To choose this chain, Byteball introduces *witnesses*. Witnesses are trusted parties that are known to everyone, and that has a stake to lose, both in the network and in the physical world. Witnesses are generally long term community members, whose real identity is known. Because of this, the network trusts these witnesses more than the average users. Byteball has 12 of these witnesses. The protocol constructs many candidate main-chains and chooses the one which has the most transactions created by unique witnesses. Then whichever of the conflicting transactions gets referenced by the main-chain first, gets chosen as the most trustworthy. The end state of this process his depicted in Figure 4.4.

Even though there is no mining in Byteball, there are still fees. Fees exist both to combat spam and to incentivize users to behave in ways that are beneficial to the system at large. This happens via different features, but most relevant for the graph is that fees paid by vertices can be partially claimed by the first new vertex who references it through a hash. This feature ensures the DAG trends towards a tighter

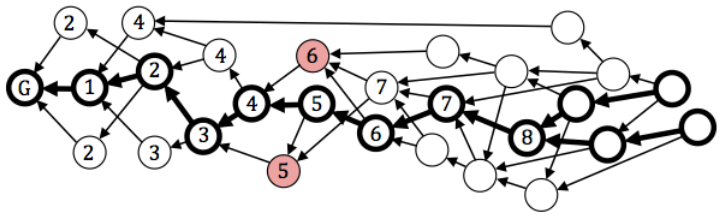


Figure 4.4: *The Byteball consensus algorithm. Protocol decides to keep the coloured transaction labeled '5' since it is referenced in the main-chain (represented in bold) before its conflicting transaction. Figure taken from [5].*

graph since the fees incentivize users to refer to as many vertices as possible. This inclusion of fees is the most significant difference between Byteball and IOTA/Nano.

Byteball wants to facilitate an easy and intuitive user experience of multiple cryptocurrency aspects. The project aims to provide both transactions and smart contract to everyday users, through intuitive interfaces and user-friendly features [34]. In Byteball one can quickly pay peer-to-peer via chat, buy goods from merchant chat-bots or place bets on the graph. Both contracts and transactions can be easily set up by keywords in a chat to bring their features to a broader audience. To make a wide array of services such as bets, predictions and insurance available in the graph, Byteball has added trusted data providers, also known as *oracles*. Oracles are generally the term used for entities that can bind together what is happening in the world at large and represent this information onto the cryptocurrency network. This data can then be used to settle betting and other events that depend on third-party events.

4.2 Proposed Academic Implementations

This section is focused on proposed implementations, mainly from an academic background. They are generally not live and do not have any tokens or networks at this time. The implementations here are all what would be referred to as *blockDAGs*, meaning that the DAG consists of blocks containing multiple transactions, instead of a single one [82].

4.2.1 SPECTRE Protocol

As a proposed solution to Bitcoins poor scalability, researchers Sompolinsky, Lewenberg and Zohar [22] from The Hebrew University of Jerusalem introduce the SPECTRE protocol in December of 2016. SPECTRE leverages more loose consensus

restrictions to allow for higher throughput of transactions in the network [22]. The researchers argue that Bitcoin's 10 minute block time and 1MB block size can not scale enough to be useful for a payment system. They also claim that changing these parameters to increase speed, while possible within the Bitcoin protocol, would only cause more forks in the chain, which ultimately leads to a waste of resources and lower stability of the user experience.

SPECTRE proposes to swap the data structure in Bitcoin, the blockchain, for a DAG of blocks containing the transactions of the system (referred to as a block-DAG [82]). The DAG will then consist of blocks of transactions as vertices and backwards references in time through hashes of earlier blocks as edges. This is done through miners referencing in a new block, all blocks with no incoming edges that the miner can see instead of the Bitcoin solution with only one reference per block. The research also introduces a third state of blocks, apart from accepted and rejected, called *pending*. Compared to Bitcoin, blocks deep in the chain would be considered accepted, shorter forks rejected, and blocks close to the end of the blockchain would be considered pending. In SPECTRE, conflicting transactions can become stuck in the pending state indefinitely.

To obtain better throughput than Bitcoin, SPECTRE has to make a compromise. The compromise made is that SPECTRE only achieves what the researcher's name *weak liveness*. Liveness is the guarantee that actors will eventually agree on a value, in this case of valid and invalid transactions. The SPECTRE definition of weak liveness is that if a transaction TX is published and no conflicting transaction TX* is published in a while, TX is accepted quickly. The weakness is displayed when there are conflicting transactions, especially when the conflicting transactions are recent. Weak liveness manifests through the three possible outcomes for a SPECTRE transaction:

1. If no conflicting transaction TX is detected, accept the transaction.

2. If a conflict is detected and the vote is conclusive, accept one and reject the other.

3. If a conflict is detected and the vote is not conclusive, leave both transactions pending.

Generally speaking, transactions will only be stuck as pending if the conflicting transactions are sent at approximately the same time. A late double spend will be rejected.

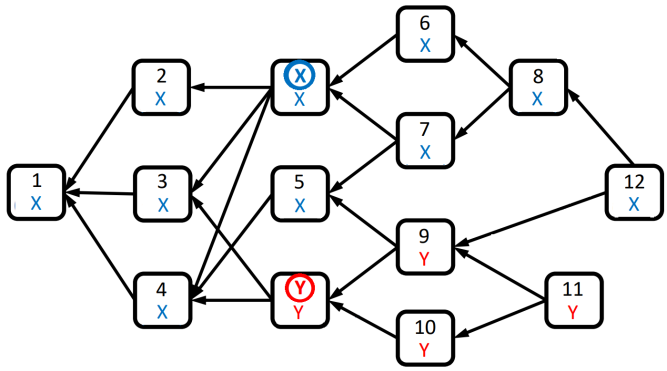


Figure 4.5: *Vote counting in the blockDAG of SPECTRE. Nodes 1-12 vote on conflicting blocks X and Y. Image is a modified version from [22]*

The main operation of the protocol can be divided into two distinct steps: (1) conduct a pairwise vote on conflicting blocks and (2) use the result of the vote to accept or reject blocks.

The first step, in which it is determined what blocks vote for what transactions, is more of a complicated voting procedure than an actual poll. There is no actual active voting from the blocks occurring, just an interpretation of the graph structure as votes to determine the weight of different blocks. The simplest case for interpreting these votes is if the block in question can reach either one of X or Y in the graph, and not both. If the block can reach one and not the other, it will be counted as a vote for the reachable one. This is the case for block 6-8 in Figure 4.5 which will vote for X and 9-11 who will vote for Y.

The next possible case is that a block can reach both X and Y in the graph. In this case, the block will be counted as voting for whichever of X and Y that has the majority of votes in the vertices reachable from the block in question. This is the case for block 12 in Figure 4.5, who votes for X since blocks 10 and 11 are not reachable from 12 giving X the majority in reachable vertices.

The last case is if neither X nor Y is reachable from the block in question. Then the blocks vote is counted towards the majority winner in the set of all blocks that can reach the block in question. This is the case for blocks 1 through 5 in Figure 4.5, who all vote for X since it has the majority of votes in the more recent part of the graph.

The results of this voting are then used to decide on what block of transactions to regard as true. To do this, there are two criteria that both have to be met for the system to choose one over the other : (1) All inputs to block B are accepted and (2) for every conflicting transaction in another block, B wins the vote.

The main idea behind SPECTRE is to combine the trusted and tested components of the existing Bitcoin system with new technology that can provide the means for better scaling of throughput and latency in the system. This is different from the commercial systems described earlier in the system, where the differences to blockchain systems are much more substantial.

4.2.2 PHANTOM and GHOSTDAG Protocol

A few years after the introduction of SPECTRE, Sompolinsky and Zohar [23] introduced PHANTOM and GHOSTDAG. They argue that discarding the *longest chain rule* of Bitcoin will significantly improve scalability without compromising security. This rule, in essence, requires all honest nodes to be aware of other blocks quickly after their creation, and thus limits block creation to allow for this knowledge to propagate before a new block can be made (This is the main reason behind the 10 minutes block interval in Bitcoin according to the authors). PHANTOM was the originally proposed protocol intended to solve this problem, but when the researches discovered that it could not scale adequately, they modified it into the GHOSTDAG protocol.

PHANTOM proposes to maintain most of the Bitcoin protocol in its current state, but replace the longest single chain with a Directed Acyclic Graph containing the blocks in the system. Like in Bitcoin, PHANTOM blocks will store multiple transactions. Where Bitcoin relies on honest miners producing the longest chain, PHANTOM relies on them to produce the largest well-connected cluster of blocks. To make a DAG instead of a chain, PHANTOM introduces an additional rule to the Bitcoin mining protocol: Instead of only referencing the latest blocks, a miner should reference all the *tips* it can see locally. A tip in this context is a block with no incoming edges in the graph, which generally means that its one of the more recent blocks added (nodes I, J and K are tips in Figure 4.6). To reach consensus on what blocks are legitimate and not, the protocol divides the blocks in the graph into trusted (blue) and less trusted (red) blocks. This selection process is the heart of the protocol, along with an ordering scheme that prioritizes blue blocks and penalizes the red ones.

As described, categorizing blocks is the focal point of this protocol. In PHANTOM, the blue subgroup consists of all blocks that are included in the largest *k-cluster* of connected blocks in the graph. The k parameter can be adjusted to facilitate higher throughput since a higher k would lead to a more massive cluster of blue blocks.

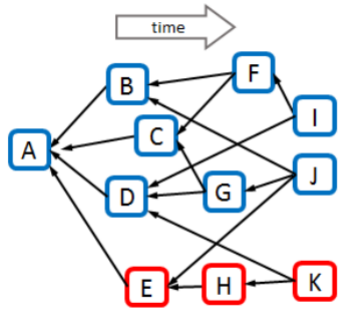


Figure 4.6: *Ordering of a DAG with k=3 in Phantom. One possible ordering of this is: A, D, C, G, B, F, I, E, J, H, K. Figure taken from [23]*

This can be deduced from Definition 4.1. From this it can be seen in that a higher k allows for more nodes in each others *anticone* inside the cluster. Nodes A and B are in each others anticone if there is no path in the graph from A to B or from B to A.

Definition 4.1. Given a DAG G = (C, E), a subset S $\subseteq$ C is called a k-cluster, if $\forall$B $\in$ S : | *anticone* (B) $\cap$ S | $\leq$ k.

After this process of classification is done, the order of all the nodes in the graph are obtained through the following steps:

1. Determine the order of all blue blocks according to some sorting algorithm.
2. For any blue block B, add red blocks reachable from B to the order *before* B itself.
3. Add lastly all red blocks that are not in the past of any blue block.

An example of this ordering can be seen in Figure 4.6. The ordering of the graph is pivotal for consensus, since PHANTOM consensus relies on having a total order of all transactions, and defaulting to the first one should conflict occur.

The observant reader will by now have, a correct, suspicion that the *maximum k-cluster subDAG* problem is NP-hard, which mean that it cant scale well enough to solve the throughput problems of Bitcoin. Sompolinsky and Zohar acknowledge this and proposes GHOSTDAG as a more practical alternative. To circumvent the NP-hard problem, GHOSTDAG utilizes a greedy algorithm for finding a sufficiently

large k-cluster, instead of finding the largest one like PHANTOM proposes. Other than this difference, they are mostly the same protocol.

The greedy algorithm in GHOSTDAG follows this pattern:

- Inherit the blue set of blocks from the *tip* in the graph with the largest blue set in its past.
- Add blocks outside *past(B)* to the blue set as long as the k-cluster property is preserved.

This algorithm is recursive and has to start executing at the genesis block level to give accurate answers.

GHOSTDAG from this chapter and SPECTRE from the last are both solutions that try to leverage existing tested technology into a system that circumvents the current blockchain problems. It is hard to comment on the practicalities of this since both systems are only theorized at this point. An organization called DAGlabs are currently working on making these protocols into a live system, but seem to be only in the early stages at the time of writing [38].

4.3 Security in DAGs

One of the biggest challenges facing DAG based cryptocurrencies is the non-negotiable security requirement that comes with the role of monetary transactions. Any currency that is not sufficiently protected from various attacks, both from the outside and within the system, cannot be viable. When dealing with monetary transactions, nothing is as important as the safety of the funds of all parties involved. The same can be said for adoption since very few users would be attracted to a system with known security risks. Security is much harder to prove than other metrics, such as throughput. One can do calculations and tests of throughput, and they tend to align well with reality. When security is concerned, it is tough to prove that a system is secure or what security means in this context.

Security is best demonstrated, especially to users, over time without any successful attacks against the systems. This is an issue for DAG-based cryptocurrencies due to their relatively young age in this market. The most pressing and discussed security issues in modern cryptocurrencies are the variations of denial of service attacks and double spend attacks. These two issues are singled out since they attack key characteristics required for a monetary system. A successful denial of service attack can diminish the usability of the system to a point where it is effectively useless. A double spend attack would compromise both the integrity and usability of the system,

since a double spend would let users use the same funds to do multiple transactions. This chapter looks at these security issues, and potential threats to the different systems discussed in Chapter 4.

4.3.1 Denial of Service and Spam Attacks

Different kinds of spam attacks have been proposed against the live DAG based cryptocurrencies currently operating. Common for the attacks is the realization that when fees are out of the picture, nothing is stopping adversaries from creating thousands of transaction and sending them to their account. This would not work on Bitcoin, where each transaction has the sender paying a fee to the miners who include it in a block. A sender can choose not to attach a fee to their transaction, but this would result in no miner choosing to include it in a block. Spam attacks have the potential to increase traffic and clog the network, impacting the experience of other users and in the extreme case rendering it completely useless for real transactions.

Of the systems discussed in Chapter 4 of this thesis, Nano and IOTA are the ones most vulnerable to spam attacks. Byteball somewhat negates this attack vector by having fees included in the system. These fees are proportional to the size of the payload being stored in the ledger. This could, in theory, be circumvented by sending empty messages to pay almost zero fees, but Byteball enforces that the hashes of two parent vertices must be included in the size estimate used for fees [5]. This ensures that there is always some fee to pay and that users cannot cut corners and only use one parent to cut fee costs. Any monetary cost greatly disincentivizes the use of spam attacks to clog the network, since it would inflict an enormous accumulated cost to the attacker, even if the individual costs per transaction is minimal. Since Nano and IOTA do not have any fees in their systems at all, they have to come up with other mechanisms to prevent spam attacks.

Proof of Work

Nano takes a different approach to anti-spam measures, by introducing a small *proof of work (PoW)* for the sender to calculate before it can send any transaction onto the main network. This is somewhat similar to Byteball, in that the sender has to spend some resources to use the network. In the case of Nano, this resource is computing power. The PoW in Nano is defined in Equation 4.1, and the threshold for the current main network is set to '0xfffffffc000000000'. This results in any random nonce input of 64-bit length having a $1.49 * 10^{-8}$ chance of being a correct solution, averaging 67 108 864 guesses to generate a valid PoW.

$$blake2b(nonce||prev\_block\_hash) >= threshold \tag{4.1}$$

The blake2b hash function is chosen for its quick speed relative to other hashing algorithms such as SHA3. In contrast to Bitcoin, Nano has no parameters in the PoW that is not known before wanting to submit something onto the network. For example, in Bitcoin, one must know what transactions to include before trying to calculate a hash for a given block. This can be seen from Equation 4.1 where the input to the hash function is only some random nonce and the previous block hash. This means that as soon as a Nano user is done transmitting a transaction onto the network, it can start calculating the PoW for the next transaction. This leads to almost no delay for most users since it is rather rare for a user to send lots of transactions after each other. It is worth noting that even though the PoW is far less in Nano than in Bitcoin, it is balanced out somewhat by the fact that Nano has a PoW associated with every transaction while Bitcoin uses one PoW puzzle for every block of transactions. This block generally holds around 400 transactions, which means should be viewed as the equivalent of 400 Nano Pow puzzles. This comparison still heavily favours Nano, since it takes the Bitcoin network 10 minutes to find a block, while the network is operating at 40,000,000 Tera-hashes per second [29].

IOTA also uses a PoW scheme to combat spam attacks towards its network. Information about this is scarce, with the white paper describing the puzzle as "similar to those in the Bitcoin blockchain" and the IOTA web page claiming that the Hashcash scheme is used [54] [1] [19]. What is clear is that the hash function in IOTA is used to create edges from any new transaction to older transactions in the DAG that it wants to confirm. As a rule, all new transactions in IOTA must verify and thus create an edge to at least two earlier transactions.

Any valid IOTA transaction has to contain a nonce that produces a valid hash for each tip it is referencing. This means that a different PoW has to be conducted for each tip. Some proponents of IOTA claim that this leads to spam attacks only strengthening the DAG with more vertices, edges and confirmations of transactions. IOTA could, however, still be susceptible to spam attacks, mainly in two different ways: overloading the Coordinator and targeting individual nodes. In both these cases, the spam attacks would be analogous to Denial of Service (DoS) attacks, in that they would flood specific actors in the system with more traffic than they can handle.

The Coordinator is essential in IOTA since it secures the network against attacks where the majority of hash power is acting maliciously. Currently, the Coordinator is a single point of failure in IOTA, and all confirmations would halt if it stops functioning [57]. In short, the Coordinator functions by regularly referencing all new valid transactions in the network. If an attacker could generate vast amounts of transactions, it is imaginable that the Coordinator could not keep up with the traffic. The same tactic could be employed against individual nodes in the network,

spamming them with thousands of transactions to build up a backlog for that node and make it unusable for other users. This attack could not happen in Bitcoin since transactions there are not sent to individual nodes, but rather to a pool where miners can choose which transactions to include.

4.3.2 51% Attacks and Double Spending

One of the major theorized attacks on the Bitcoin network is what is known as the 51% attack. This attack stems from Nakamoto noting in his original paper that the network is only secure under the assumption that honest miners control at least 51% of the total hashing power used for creating new blocks in the system [17]. Since the longest chain in Bitcoin will always decide what part of accepted history is, someone who can dictate the contents of and produce the longest chain will have complete control of what transactions are accepted and rejected. Anyone with more than half the hashing power of the network, will generally always be able to produce a longer chain than the remaining nodes. Being able to produce a longer chain at will enables the attackers to double spend their own transactions, first on the original chain and again on the new constructed attack chain.

The protection against this attack has always been the claim that a sufficiently decentralized network will naturally resist such a concentration of hash power. This decentralization was discussed in Chapter 2. The DAG-based systems accounted for in this thesis still is susceptible to 51% attacks, though in quite different manners than Bitcoin. Worth mentioning is the fact that it is generally speaking more expensive to 51% attack Bitcoin over an extended period of time than some other coins, due to the extreme amount of electricity required to keep such a huge hashing power operative. This differs from other schemes such as proof of stake where the expenditure would be a one-time payment.

Nano

In Nano, a 51% attack would have to take the form of either a large scale token buyout or an attacker convincing nodes that he represents and votes on behalf of a huge set of Nano coins. This modified 51% attack vector follows from the fact that Nano uses a distributed *proof of stake* (PoS) protocol to achieve consensus on what transactions are valid or not in the case of duplicate transactions also known as double spends. PoS in Nano is, as mentioned in Chapter 4, implemented such that token holders can choose representatives (a node running the Nano protocol) to vote on their behalf.

The voting is weighted according to the number of coins any representative is voting on behalf of. Practically this means that an attacker, instead of having to control 51% of the total hash power, has to obtain 51% of the voting weight in the

system. This could theoretically be done both though straight up acquiring 51% of all coins in the system, or by buying a large number of tokens and then try to incapacitate honest node such that the attacker holds 51% of the *active* voting power at any given time. The latter seems much preferable to the prior, since amassing half the tokens would be incredibly costly and leave the attacker without any way to recoup this cost. Storing value in tokens and then destroying the network that gives the tokens value does not seem like a worthwhile endeavour, although this depends on the economics of the situation. A 51% attack in Nano would only require the money to be spent once, and it would immediately lose all value when the network realizes that it is under attack. This differs from Bitcoin where attacks are more expensive to sustain, and the network could more easily recover from such a situation.

IOTA

Although IOTA uses a DAG for storing its transactions, it still employs PoW verification of transactions to ensure consensus on what transactions are valid. This is done through users having to verify earlier transactions to add new transactions to the network, while transactions with a larger amount of verifies are more likely to be the ones accepted as true if any disputes arise. This seems to suggest that attackers with huge computational capabilities can exert a large influence on what is accepted and not in the network. It seems clear that 51% of the hash-rate should be enough to dictate transactions, but some research suggests that lower amounts of hash power might still be enough to take over the system if coupled with other kinds of attacks [9]. This should be especially true for asynchronous networks like IOTA and Nano since the timing restrictions in these systems are more relaxed.

Since IOTA is still relatively new, a 51% attack would not be very costly. This stems from the fact that there are a small number of nodes generating traffic in the system, and thus, only a small investment is needed to surpass the 51% figure. When scaled up to usage in IoT systems, most believe mounting a 51% attack to be far too costly of an endeavour, much like how Bitcoin is currently protected by economics. To remedy this (hopefully) time-limited weakness, IOTA uses the Coordinator as a shield towards attempts by malicious actors to gain the majority of hash power. Currently, IOTA only considers transactions referenced by the Coordinator as valid transactions [57]. This means that any 51% attacker needs to trick the Coordinator into referencing its malicious transactions for any attack to be successful. This has not yet been the case, but many cite this as a critical point of contention with IOTA since the Coordinator is closed source and thus makes it hard for the public to verify its security capabilities (As of April 10th the Coordinator is now open source in the IOTA network)

Byteball

Byteball differs from other systems in consensus attacks mainly due to one crucial difference: it does not try to be decentralized. Consensus on valid and invalid transactions in Byteball stem from a total ordering of the graph, meaning that for all vertices, it is possible to determine which transactions were posted before the other. This ordering is obtained through trusting 12 *witnesses* to behave in a certain honest way (this is described in more detail in Section 4.1.3). This centralized consensus method is far less susceptible to any attempts to hijack the consensus method in systems. An attack similar to the 51% attack in Bitcoin would have to compromise more than half of the 12 witnesses. Currently, the founder of Byteball is operating 12 out of 14 possible witnesses, making the system even more centralized than it seems at first glance. This could be compared to IOTA, where the Coordinator is a single point of failure for the consensus.

Chapter 5
Experimental Results

Here follow the results gathered to facilitate a comparison between the DAG systems on the grounds laid out in Section 2.4.1. Each section contains a small description of how the results were produced or gathered before the results are presented, and comments are noted. The process followed here is inspired by the methodology laid out in Chapter 3 for data gathering, experiments and some metrics, in particular, are commented there. After all the individual results have been presented, they are summarized in a table for easier comparison both between DAG protocols and to the blockchain systems laid out in Table 2.1.

5.1 Max Theoretical Throughput

Nano claims, in their whitepaper, to be able to process over 10,000 transactions per second. The whitepaper, however, does not explain how it arrived at this number, other than that is was reached in a private network with "commodity" SSD hard drives [13]. This would have to be on a version of the network where the PoW for sending blocks was either removed or massively precomputed, since computing 10,000 pieces of PoW per second is extreme. For reference, an NVIDIA Tesla V100 can compute about 6.4 PoW per second, at a price point of roughly 80,000 NOK. The PoW scheme for Nano is explained in Equation 4.1. However, the Nano team and community are not particularly interested in what the upper limit for TPS in its system is. This stems from two different points:

1. Currently, the network is operating at a very low load, far from any throughput bottleneck. This means that scalability is not its main concern at the moment.

2. According to the whitepaper and the Nano team, the TPS in the network is only bound by the IO speed of its hardware components and the bandwidth available between nodes.

Since both of these metrics will naturally progress on their own without the

effort of the cryptocurrency community, it is now viewed as a major concern for the scalability of the network. It is hard to strictly define an upper limit for the maximal theoretical throughput the Nano network can process. This is due to there not being any limiting factor in the protocol defining the mechanics of the network. The limiting factor is the bandwidth of nodes and the computing resources available to them. Since this differs wildly from node to node, it is thus tough to estimate the upper bound of this metric.

IOTA is comparable to Nano in this metric since the design of the system does not limit throughput in any way when it is fully implemented. The most significant difference in this regard is that IOTA currently is running a protocol not entirely analogous for their final desired network. This mostly stems from the Coordinator being live and the confirmations of transactions entirely depending on it. It is hard to determine what an upper bound for throughput with the Coordinator operating is, even though some of its limitations are known. The Milestones released by the Coordinator can only directly reference two other transactions, but it will also confirm all transactions referenced downstream by these two referenced transactions. The coordinate issues milestones regularly with an average interval of 3 minutes.

Since TPS depends on users submitting transactions, IOTA also operates with a *Confirmed TPS* (CTPS) measurement. This metric stems directly from the coordinators' ability to confirm transactions. Since these confirmations only come from milestones, the graph of CTPS is more spiked and less continuous than that of the regular TPS [60]. CTPS is thus just the confirmed version of TPS, where the confirmation comes from being referenced by a milestone released by the Coordinator. This behaviour can be viewed in Figure 5.1. Short bursts of confirmed transactions make it seem like the CTPS sometimes outperforms the TPS, which can never really be the case. This holds for the Coordinator as it has operated until the 10th of April 2019. On this date, the closed source coordinator will be replaced with the open source coordinator named *Compass*. This will change the node software from 1.16 to 1.17, and all nodes must follow this upgrade to maintain its activity in the network.

5.2 Average Measured Throughput

Nano by design is a little harder than other protocols to measure TPS accurately. This is due to only blocks being propagated through the network. A transaction will not be complete until both a send and a receive block has been made, meaning two blocks are needed for a successful transaction. With the release of universal blocks in v11 of the protocol, released in March of 2018, Nano now only issues different *"state"* blocks in the network, instead of individual *"send"* and *"receive"* blocks [65]. To investigate how many of these blocks the node have seen in a time interval, one can call a *block_count_type* command via the RPC interface [66].

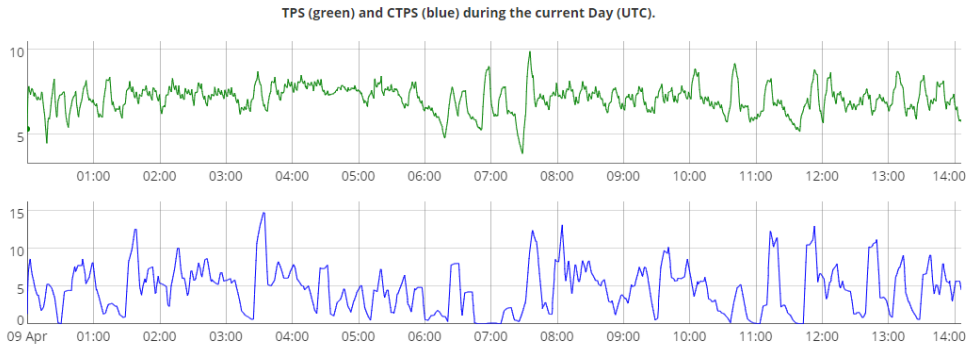


Figure 5.1: *TPS (green) and CTPS (blue) observed on the IOTA mainnet on the 9th of April 2019. Graphs from [60].*

```
root@NanoNode:~/Nano# curl -d
{
    "send": "5016664",
    "receive": "4081228",
    "open": "546457",
    "change": "24193",
    "state_v0": "4216407",
    "state_v1": "8156136",
    "state": "12372543"
}
```

Figure 5.2: The block count observed by our node, divided by type, on the 24th of March at 12:11.

```
root@NanoNode:~# curl -d '{
{
    "send": "5016664",
    "receive": "4081228",
    "open": "546457",
    "change": "24193",
    "state_v0": "4216431",
    "state_v1": "8433255",
    "state": "12649686"
}
```

Figure 5.3: *The block count observed by our node, divided by type, on the 10th of April at 12:40.*

As can be seen in Figure 5.2 and Figure 5.3 the only blocks that are being sent over the Nano network currently are different kinds of *state* blocks. This is due to a protocol change where Nano switched to only ever-changing account states through *state* blocks. This means that the four different types of blocks that existed before all have their functionality integrated into state blocks. For this experiment, this change in protocol results in a less accurate estimation of TPS, since it is now hard to discriminate between different types of blocks in the system. The following calculation is this an upper bound for throughput in the system.

To calculate the locally observed TPS from our node, the total amount of blocks seen in the interval is needed. To get a precise as possible measurement, we only consider blocks labeled as *state_v1* in the count displayed in Figure 5.2 and Figure 5.3. This is because *state_v0* blocks can be sent, but will not get added to the ledger since they were only used as a bridging tool from the old scheme to the new state

block one. Including these would, therefore, make the result less reliable. It is then possible to take the most recent number of *state_v1* blocks and subtract the number found at the start of the measurement. As we can see from Figure 5.2 and Figure 5.3 this difference comes to 276 299 blocks. With this information, the only missing piece is time between the measurements. Time stamps obtained from the pictures will suffice for this. Inspecting the metadata of the files reveal that the first measurement was taken on the 24th of March at 12:11 while the second measurement was taken on the 10th of April at 12:40. This yields a time difference of 1 470 540 seconds. Using these numbers, it is then possible to calculate the throughput of the network, in the form of transactions per seconds, as observed from our node.

$$TPS = \frac{Blocks}{2 * Time} = \frac{279299}{2 * 1470540} = 0.09395 tx/s \tag{5.1}$$

Equation 5.1 concludes that the Nano node observed an average of 0.09 transactions per second over the course of 17 days of regular operation. It shows the Transactions Per Second as observed from the Nano network node. The number of blocks is halved to reflect that two blocks are created to finalize a transaction.

The **IOTA** reference implementation has a much better interface for measuring the statistics relevant for a scalability comparison. Instead of second-hand calculation of data, the IOTA node can measure, calculate and plot the relevant metrics. In order to give a fairer comparison to the blockchain systems, only confirmed transactions would count towards the rate calculated. This is to match the fact that only transactions added to the chain in blockchain are considered when observing the TPS from online explorers. The data this comparison will build on is displayed in Figure 5.4.

From the CTPS (Confirmed Transactions Per Second) measurements in Figure 5.4, a clear and stable trend can be deduced, with some deviant periods with increased or decreased traffic. This graph is the result from a call to the *rate(iota_zmq_confirmed_tx_count)* endpoint in IOTA nodes. This endpoint answers with the confirmed transactions per second for the time interval supplied to it, with 12-hour intervals for the last two weeks used in the figure.

It is also interesting to note the substantial difference between the number of transactions with no monetary value associated displayed in green, and the transactions represented by yellow, which carry a cryptocurrency value with them. Both of these will be included in the calculation of average CTPS since there is no difference in how the system handles their transactions. From the node output it is gathered that on average, during the two weeks proceeding the 30th of April 2019,

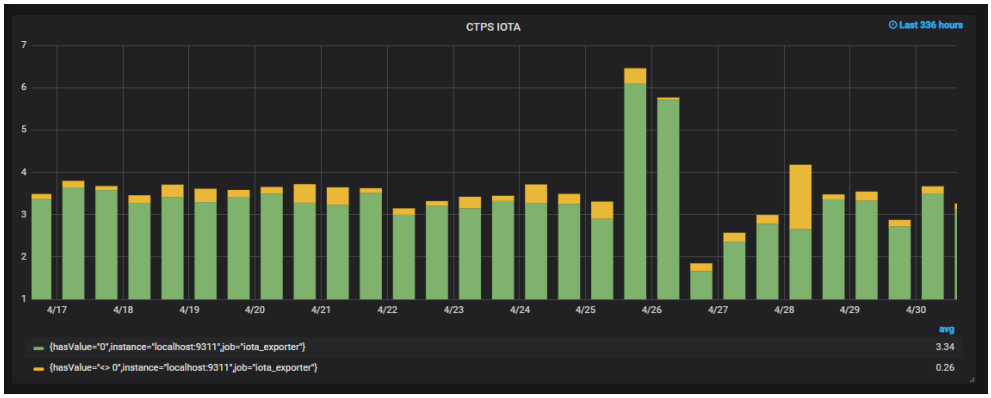


Figure 5.4: *Graph shows TPS and CTPS from the IOTA node for the last two weeks. The green part of the bar represents transaction with no cryptocurrency value associated with it, while yellow has a non-zero value. In the bottom right, the two-week averages for both of these types of transactions are listed. Image captured on the 30th of April.*

there were 3.34 CTPS with no value and 0.26 CTPS with value. This sums to a total of 3.6 CTPS for the network in the observed period.

5.3 Maximum Measured Throughput

The team behind **Nano** is officially cooperating with the third party block explorer Nanode to display blocks, transactions and statistics of the network [69]. Reading into their historical chart of total blocks in the Nano ledger, there is a visible spike at the step from the 24th to the 25th of February 2019. Reading this data yields 19 800 919 blocks in the ledger at the 24th and 21 544 590 at the 25th, leading to a difference of 1 743 671 blocks added in these 24 hours. Reading into the discussion at the time and later, this seems to be the result of a spam test of the network, following the release of v18 of the official Nano node software. Having the difference in blocks enables the calculation of TPS as denoted in the calculation shown in Equation 5.2.

$$TPS = \frac{Blocks}{2 * Time} = \frac{1743671}{2 * 60 * 60 * 24} = \frac{1743671}{172800} = 10.09 tx/s \tag{5.2}$$

Equation 5.2 shows the mathematical calculation used to devise the maximum measured throughput in the Nano network. This is done by dividing the number blocks added to the ledger during a 24 hour day by the number of seconds in this

period. The number of blocks is divided by two, due to reasons of the Nano protocol discussed in Section 3.4.1. This concludes that the highest throughput observed in the Nano network, sustained over 24 hours and that data is available on, is 10.09 transactions per second.

When also considering peak TPS during stress testing of the network, the numbers increase substantially. Looking at the two best-documented stress tests of the Nano network, they both land in the hundreds of transactions per second. The first test, conducted in January of 2018 by a single person, shows that the network sustained 100 TPS over 47 seconds with a peak of 306 transactions broadcast in a second [48]. This experiment used the aforementioned precomputed PoW model. All of these transactions were broadcast from a single node, without the network being fully saturated, and the author notes that the network behaved as usual throughout the test. The second stress test was performed by the community at large, with coordination and encouragement from the Nano foundation. This differs from the first test in the way that a higher number of nodes broadcast the transactions, instead of a single point of origin. The Nano foundation reports that a sustained rate of 75 TPS was observed over 30 minutes while the highest throughput measured was at 378 TPS. This test was conducted in August of 2018, after the release of v15 of the Nano node software.

IOTA has a worse record of their transactions on generally when one is concerned with the data contained in their ledger. This is mainly because of their method of *snapshots*, where the network agrees on a state and then erase all unneeded history of how it obtained this state. This practice is needed to keep the size of the ledger down such that smaller nodes do not fall off. The unfortunate result of this is that it is hard to keep records for an extended period. The most extensive records of throughput found through the research done in relation to this thesis went back three months [52]. In this data, it is found that on the 4th of March, the network sustained an average of 10 CTPS for the entire 24 hour day. Due to the limits of the data, this will have to serve as the highest measured sustained throughput.

On a similar note, the sporadic stress tests for the IOTA network does not appear to be properly documented either. Rumours and numbers get thrown around quite a bit in the community, but few real sources are found. As it is outside the scope of this work to carry out stress tests in these networks, second-hand reporting is the only option. The only reports of stress test results that are both available and somewhat reliable come from Dominik Scheiner, one of the IOTA co-founders. He has posted two results from internal stress tests on Twitter, clocking in at 112 and 105 CTPS peak performance [49] [76].

5.4 Minimum Theoretical Delay

The **Nano** protocol has no intrinsic delay sources described in the specifications that mimic that of the block delay in the blockchain. As earlier mentioned, this leads to the main force behind delays in the Nano system are the transfer speeds of networks and hardware that the protocol is running over. Contrary to what the Nano whitepaper claims, there are votes on every transaction in the network about their validity [13] [72]. The votes would then contribute to the theoretical delay for all transactions. Votes have no built-in time delay component to them, making the delay only dependent on how long it takes for enough votes to be registered.

As will be seen in measurements of confirmation time, the time needed for this is measured in milliseconds. The minimum theoretical delay is thus dependent on the hardware of the nodes and the delay in the network where the transactions are broadcast over. This makes it hard to put an exact number on the delay, but it is at least in the sub-second range. One could argue that the computational time needed for the PoW in Nano should have been included in this, but since for almost all ordinary use this PoW is precomputed ahead of time it is chosen to keep this separate when dealing with *minimum* theoretical delay.

IOTA has a slightly more complicated relationship with minimum theoretical delay, both due to its protocol design and its current implementation. Running with the Coordinator as a central entity in the system turns the delay and confirmations into a blockchain like structure. The reason for this is that milestones from the Coordinator behave like blocks, with the time between milestones comparable to the inter-block time in blockchain solutions. A discussion about the intricacies of this will follow in Chapter 6. The method of determining minimum delay will thus be similar to that of Bitcoin and Ethereum, taking the average time between milestones and using a probabilistic approach of assuming half of this time as the average waiting time. Data from explorers seem to converge at about 150 seconds between milestones published in the last 24 hours as of April 18th [60]. The results obtained shows an average delay of 75 seconds from awaiting confirmation.

The other component driving up delay is the proof of work that is needed to add a transaction to the Tangle. In contrast to Nano where this PoW can be precomputed, IOTA relies on current network state to compute the PoW and thus it cannot be determined ahead of time. Like Nano, IOTA also has to adhere to the limits of hardware and network delays, but on top of these delays, the time required for computing the PoW is added. Brief tests show that an Intel i7 CPU uses about 4.1 seconds to compute the required workload [74]. It is reasonable to assume that this hardware is in the upper echelon of performers for this task. It is worth noting that for IOTA's target audience of IoT devices, this PoW calculation will require far more

time.

5.5 Median Measured Delay

In order to compare delay in the **Nano** network with that in a blockchain network, it is most fitting to look at the delay as the time measure until a transaction is confirmed in the ledger by nodes voting on its validity. This is the case since when investigating blockchain delay is the time between broadcast and confirmation followed by publication on the chain. Given the node set up to operate in the Nano network, it is quite easy to obtain measurements for this metric, although they do not go extensively back in time. The RPC calls are limited to obtaining information about the last 2048 blocks to be confirmed in the network.

```
root@NanoNode:~# curl -d '{ "action" : "confirmation_history" }' [::1]:7076 | grep average
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0       "average": "1804"
100  478k  100  478k  100    37  6733k    521 --:--:-- --:--:-- --:--:-- 6734k
root@NanoNode:~#
```

Figure 5.5: *Average duration of the last 2048 elections of broadcast blocks in the Nano network at the time of this image. This is the main source of delay since blocks will have to be confirmed before they can be regarded as finalized in the ledger. Image is taken from the Nano node used in this thesis. Image captured on the 21st of April 2019.*

As mentioned earlier in the methodology chapter (Chapter 3) it is advisable, for the Nano network, to do multiple measurements to give a more robust answer. Table 5.1 lists measurements of confirmation time and the dates they were taken. Figure 5.5 displays the call and reply from the RPC interface that was used to obtain these measurements.

The average from Table 5.1 is supported by third-party explorers, through measurements of different percentiles of confirmation and propagation time. Online block explorers show that, except for some outliers, the combined propagation and confirmation time for 95% of individual transactions have been below 700 milliseconds since the protocol update on the 22nd of February [61]. This falls in line pretty far behind the measurements carried out in these experiments, giving confidence to the fact these measurements do not flatter the speed of the network more than is deserved.

As laid out in the methodology chapter, **IOTA** nodes have a better interface for these kinds of essential measurements. By calling different commands that sum up

Time of Measurement	Average Time
21st of April 16:15	1804ms
22nd of April 11:30	1260ms
23rd of April 11:00	799ms
24th of April 09:00	1079ms
25th of April 09:00	1061ms
Average measurement	1201

Table 5.1: *Measurements of average confirmation delay in the Nano network. Measurement is of 2048 last blocks, and time and date is listed to prevent overlapping samples. The average of all measurements in the table is calculated in the last entry.*

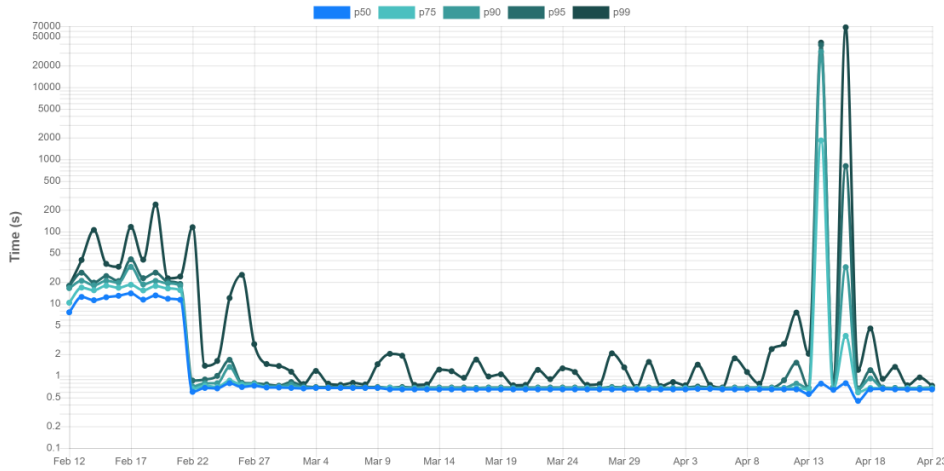


Figure 5.6: *Percentiles of combined propagation and confirmation time for transactions on the Nano network. Propagation time measured between nodes located in the UK and Germany. Percentiles are computed with 12 data points per hour for each day in question. Image captured from [61]*

the total confirmation time and divides it by the number of transactions that occurred in this time interval, it produces both time series and averages of confirmation times in the chosen interval. Figure 5.7 shows the graph of 10 minute average confirmation time in seconds, plotted over the last seven days as of capture. On the bottom right, the average delay for the whole period is displayed in seconds.

From this measurement, it is concluded that the average confirmation delay for

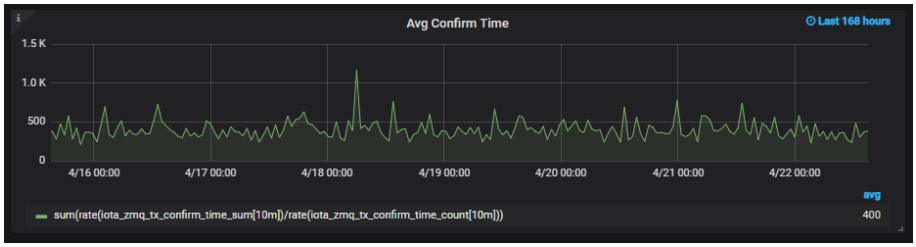


Figure 5.7: *Seconds until a transaction is confirmed from when it is broadcast. Confirmed in this case means referenced directly or indirectly by a milestone. Image captured from IOTA node on 22nd of April.*

IOTA transactions is 400 seconds, or 6 minutes and 40 seconds, as of the 22nd of April 2019. This also means that the other delay sources become negligible in comparison since this delay is so considerable. For reference, it was found that the PoW would add about 4 seconds on high-end hardware.

5.6 Time to Stability

Nano's stability and finality in transactions rely on the votes conducted on transactions in the system. These votes have been described earlier, but in short, they are a real-time election used to confirm blocks across the network. A key point about these elections is that currently the results are not stored, such that if a node is in doubt about a block, it has to conduct a new election since it cannot review results of earlier elections concerning the specific block. This will be addressed in v19 of the Nano protocol, which introduces confirmation history. This is significant due to the fact that currently if a transaction wins a vote, there is no guarantee that it will win the same vote in the future.

With confirmation history, this problem would be circumvented, and the resources used for duplicate elections can be used elsewhere.Thus, with the current stability regime in Nano, it is hard to determine a value for time to stability due to nodes being able to call a new vote on any block at any time. With the introduction of confirmation history in v19 of the Nano protocol, the time to stability will be the same as the confirmation time, which is to say that it would be in the sub-second range[63]. It may be somewhat naive to trust the promises of coming technology to solve current issues, especially in the ecosystem of cryptocurrencies, and this will be discussed in Chapter 6.

IOTA has a rather straightforward relationship with stability in their current network state. Given that the Coordinator is still in operation, and since IOTA base confirmation of transactions solely on this actor, confirmation is a binary state

on the current IOTA implementation. If a transaction is either referenced directly from a milestone issued by the Coordinator or is downstream (reachable) from a transaction referenced by a milestone, it is confirmed and will not be overturned. This means that the time to stability is equal to that of the confirmation via the Coordinator. Historical data from the network shows that the Coordinator releases a new milestone about every 150 seconds as the time of this writing [60]. Like shown in the *Median Measured Delay* results, this leads to a time to stability measurement of about 400 seconds on average. When IOTA eventually removes the Coordinator from their network, this will change rather drastically. The consequences of this is discussed in Chapter 6,

5.7 Node Storage

Determining the storage resources needed for running a node in the **Nano** network is a pretty straight forward task. About two months into running the node, the total disk space in use by the Nano protocol comes to 15 GB. This could be due to its low amount of transactions compared to Blockchain systems. At 24 305 132 total blocks in the ledger, to find an estimate of the total transactions, the count is divided by two as earlier explained. This results in 12 152 566 transactions and thus 1.23KB per transaction in the ledger.

```
root@NanoNode:~/Nano# stat data.ldb
  File: data.ldb
  Size: 14980300800     Blocks: 29258408   IO Block: 4096   regular file
Device: fc01h/64513d    Inode: 270017      Links: 1
Access: (0600/-rw-------)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2019-04-19 11:06:31.817214871 +0000
Modify: 2019-04-26 12:35:44.077304232 +0000
Change: 2019-04-26 12:35:44.077304232 +0000
 Birth: -
```

Figure 5.8: *A simple storage evaluation of the data.lbd file which hosts the ledger used by the Node operating in the Nano network. Image captured on the 26th of April.*

IOTA storage resource utilization is equally easy to determine. After about three weeks of running in the caught-up state, the disk space usage was measured as shown in Figure 5.9, to about 8 GB for the entire directory containing all configuration files, scripts and the ledger.

```
root@IOTA:/var/lib# du -hs iri/
7.9G    iri/
```

Figure 5.9: *Storage evaluation of the Iota Reference Implementation (IRI) directory of the node operating in the IOTA network. Image captured on the 26th of April.*

Given a transaction count at 5 052 485 at the time of the size measurement, this results in a size per transaction metric of 1.58KB.

5.8 Initial Download

The **Nano** node took a few days to catch up to the network at large after being initialized. In this time, the node downloads all the blocks and verifies them locally. In order to ensure that no preference is given to the DAG systems in this comparison, the traffic statistics for March is used as a measurement for the traffic used to start a node. Since the nodes were restarted for measurements on the 26th of March, this results in 5 days of usage.

For Nano, there are two different network interfaces listed in the node configurations. This is due to the Nano node running a docker image, which requires a virtual network interface that can be used to communicate between the docker image and the machine it is running on. This "network" traffic will thus not be considered in the measurement since it is just I/O traffic internal to the machine. All traffic on the outbound *eth0* interface will, however, be considered as node related traffic. This leaves some margin for error as other processes could produce some traffic as well, but this should be significantly limited since the machine is not being used for any other purpose than being a node. Figure 5.10 showcases the results of the traffic measurements done over time for the node.

```
root@NanoNode:~# vnstat

                    rx      /      tx      /     total    /   estimated
docker0:
      Mar '19    118.50 GiB  /   56.18 GiB  /  174.68 GiB
      Apr '19    259.81 GiB  /  214.07 GiB  /  473.87 GiB  /  556.99 GiB
    yesterday      4.72 GiB  /    8.32 GiB  /   13.04 GiB
        today      1.75 GiB  /    4.08 GiB  /    5.84 GiB  /   11.16 GiB

eth0:
      Mar '19     56.59 GiB  /  120.24 GiB  /  176.83 GiB
      Apr '19    215.67 GiB  /  270.06 GiB  /  485.73 GiB  /  570.93 GiB
    yesterday      8.40 GiB  /    5.17 GiB  /   13.57 GiB
        today      4.13 GiB  /    1.94 GiB  /    6.06 GiB  /   11.59 GiB
```

Figure 5.10: *Network traffic measured to and from the Nano network node with the tool vnstat [78]. Traffic from the eth0 interface is the actual network traffic, docker0 is a virtual interface. Image captured on the 25th of April.*

The measurement shows that the Nano node consumed about 176 GB of network traffic, with 56 GB being download and 120 GB being upload. This is interesting since generally, download bandwidth is the more consumed resource.

The **IOTA** software does not run in a docker container, and thus only has one active network interface to consider. The IOTA node was harder to get to catch up to the network, and thus, the time interval of bandwidth consumption is more extensive. To help more accurately measure this a day to day measurement was done for April. Thus it is possible to add up the March data with the days until the 10th of April when the node is considered to have caught up. These data sets can be observed in Figure 5.11 and Figure 5.12.

```
root@IOTA:/var/lib# vnstat

                     rx      /      tx      /     total    /   estimated
docker0: Not enough data available yet.
eth0:
      Mar '19     70.71 GiB  /  110.67 GiB  /  181.38 GiB
      Apr '19    319.57 GiB  /  365.37 GiB  /  684.93 GiB  /  804.93 GiB
    yesterday     14.55 GiB  /   16.17 GiB  /   30.72 GiB
        today      7.03 GiB  /    6.81 GiB  /   13.84 GiB  /   26.25 GiB
```

Figure 5.11: *Network traffic measured to and from the IOTA network node with the tool vnstat [78]. Image captured on the 25th of April.*

```
root@IOTA:~# vnstat -d

 eth0  /  daily

        day         rx      |     tx      |    total    |   avg. rate
     ------------------------+-------------+-------------+---------------
      04/01/19     12.61 GiB |   24.18 GiB |   36.79 GiB |    3.66 Mbit/s
      04/02/19      9.43 GiB |   26.53 GiB |   35.96 GiB |    3.58 Mbit/s
      04/03/19     12.27 GiB |   30.37 GiB |   42.64 GiB |    4.24 Mbit/s
      04/04/19     14.19 GiB |   26.40 GiB |   40.59 GiB |    4.04 Mbit/s
      04/05/19     12.96 GiB |   20.20 GiB |   33.17 GiB |    3.30 Mbit/s
      04/06/19     12.44 GiB |   21.57 GiB |   34.01 GiB |    3.38 Mbit/s
      04/07/19     14.82 GiB |   22.99 GiB |   37.81 GiB |    3.76 Mbit/s
      04/08/19     19.74 GiB |   21.08 GiB |   40.83 GiB |    4.06 Mbit/s
      04/09/19     18.18 GiB |   15.65 GiB |   33.84 GiB |    3.36 Mbit/s
      04/10/19      8.61 GiB |    5.88 GiB |   14.49 GiB |    1.44 Mbit/s
```

Figure 5.12: *Network traffic measured to and from the IOTA network node with the tool vnstat [78]. Image captured on the 25th of April.*

Looking at the measurements, it is noted that March saw a total of 180 GB of network traffic, while the relevant days of April sum to 334 GB. This leaves a total of 514 GB of traffic in the period starting from initialization and lasting the node was caught up to the network. It is noted that this is probably a sizable overestimation due to a lot of this traffic being generated while the node was struggling to catch up, and serves little to no purpose in the context of catching up the node.

5.9 Scalability Summary

This is a summary of the results earlier in this chapter. For presentation, comparison and ease of reading all the metrics are summarized in Table 5.2. The findings, their limitations and implications will be discussed further in Chapter 6. Included in this table is also the values for Bitcoin and Ethereum for the same metrics, as the representatives from the blockchain category.

Scalability Metric	Nano	IOTA	Bitcoin	Ethereum
Max Theoretical Throughput	Unlimited	Unlimited	7 TPS	27 TPS
Average Measured Throughput	0.094 TPS	3.6 TPS	4.04 TPS	6.16 TPS
Maximum Measured Throughput	10.09 TPS	10 TPS	4.92 TPS	15.62 TPS
Minimum Theoretical Delay	<1 second	75 seconds	300 seconds	7 seconds
Median Measured Delay	1.201 seconds	400 seconds	507 seconds	26 seconds
Time to Stability	Unclear	400 seconds	60 minutes	180 seconds
Node Storage	15GB	8GB	200GB	150GB
Initial Download	176GB	514GB	200GB	2,200GB

Table 5.2: *Scalability metrics with values from Nano and IOTA. Metrics in table equal to those found in Table 2.1*

Chapter 6
Discussion

This discussion of the results will mainly focus on the summarized table displayed in Section 5.9 and the metrics contained therein. How these data points were obtained is already commented on in the results chapter, so this discussion will aim to shed light on what decisions were made and how they impacted the results. It is also a goal to discuss future remedies to some current issue and comment on difficulties that can have impacted the results.

6.1 Max Theoretical Throughput

After extensive research to try to pin down an upper bound on this metric, it was concluded that any number set would be hard to defend in any honest manner. In theory, one could do calculations on the usage of different resources, such as bandwidth and volatile memory usage, and then estimate the capacity of the networks to handle this usage. Based on this capacity, an upper bound could be constructed. This bound would, however, not be no more robust than the estimate of capacity across the network, and the basis for such an estimation is found to be too weak to base any exact calculation on. It was therefore decided to base this metric strictly on a protocol view of the system, and investigate whether there is anything intrinsic to its operations that would limit throughput. This was not found to be the case, and thus, it was concluded that in principle, nothing is stopping these networks from scaling as they please.

This is not to say that in their current iteration, there is no upper bound to throughput. The systems are heavily restricted through implementation trade-offs and hardware limitations. This can be seen for example in Nano where, during high traffic situations, weaker nodes will not be able to keep up with consensus voting and results and not being able to catch up until traffic slows down. In IOTA, the coordinator was introduced to increase security in a low traffic network[57]. This severely limits the performance of the network to increase security and is an

implementation trade-off. Given that the networks reach a successful implementation of their protocols, there is no apparent reason that they should not be able to scale in line with hardware evolution.

6.2 Maximum Measured Throughput

For this measure, several different metrics were considered. The most obvious contender for this kind of measurements were stress-tests that pushed their respective networks to the limit of what they are capable of. Due to the scale of such a test, it was out of the scope of this thesis to carry out such a test that would have a meaningful impact on the networks. Thus this work is limited to observe such tests from other sources. This proved to be a substantial problem due to the severe lack of proper documentation.

The lack of documentation can be seen in the results, where tweets from one of the IOTA co-founders are one of the few ways to obtain any data on these tests. These data points are not very reliable, and thus cannot be the basis for any comparison. Due to these circumstances, the highest average throughput measured in the whole network over a 24 hour period was chosen as the most consistent basis for comparison. This decision was made for several reasons, the first and most notable being the availability of reliable data. Data on usage over 24 hour periods is freely available, sometimes both through the node interfaces as well as online explorers and data collectors. This measure should also be more reliable since, on a scale of seconds, small errors or discrepancies in the data can quickly be blown out of proportions.

The need for verifiable data is rooted in the desire for a fair comparison between DAG and blockchain systems. On a similar note, stress testing the Bitcoin network would be a rather mundane task, since trying to force the TPS to its limit in that network would mainly consist of trying to make transactions small enough that many would fit inside the 1-megabyte block size. Transmitting many transactions to the Bitcoin network would only increase the mempool size, which happens in the normal operations of the network. Since it can only transmit and confirm about seven transactions per second, it is quite usual for the Bitcoin network to receive more transactions than it can handle. This means that it would be hard to make a fair comparison between the stress tests of different networks. It is, however, still useful for showing what peak capacities the DAG networks are capable of withstanding.

6.3 Minimum Theoretical and Measured Delay

Out of the four systems considered in the comparison (Nano, IOTA, Bitcoin, Ethereum), Nano is the only one to not rely on a probabilistic calculation of waiting for some event in their theoretical delay. IOTA, Bitcoin and Ethereum are all bound

by the publication/creation of blocks to publish confirmed transactions on their network. This observation can be used to claim that IOTA, in its current design iteration, is somewhat closer to a blockchain than a DAG system in some regards. Since confirmation of transactions can only happen from milestones published from the central coordinator, this can be thought of like a miner publishing a block confirming transactions by referencing them.

There are still two significant differences between conventional blockchain and this IOTA model: one single central miner and no fees. A single central miner generally infers that the process will move slower and be more vulnerable than if a decentralized network of individual miners were used. This is reflected in that in the Bitcoin system, the average transaction has to wait about 1.7 times the minimum theoretical delay before it is included in a block. In IOTA this rate is increased to 5.3 times the minimum theoretical delay. Thus IOTA performs far worse in this regard, while also being more vulnerable to attacks and faults in its one central *mining* entity.

The time for transactions to be included in a confirming block is also impacted by the second difference mentioned above, the absence of fees in the system. In Bitcoin, a sender can provide a higher fee to incentivize miners to include their transactions over other ones in the pool of available transactions. This differs from IOTA, where the sender of a transaction has no way of manipulating the process to more rapidly reference the transaction. While this can be viewed as a problem, the exclusion of fees carries many positive impacts as laid out in the descriptions of the systems in Chapter 4.

The fact that Nano does not rely on this blockchain-like publication method of confirmed transactions is reflected in its very low minimum theoretical delay. This delay consists only of the vote over which transactions are valid and the propagation delay for the transaction in the network. The median measured delay is slightly higher than theorized, but when cross-checking the data points with other sources, it is observed that the measurements here are in line and even a bit slower than those observed by other nodes in the system.

It is worth mentioning that IOTA aims to move away from this blockchain like design in favour of a more DAG based consensus mechanism as well. When, and if, IOTA succeeds in removing the coordinator from its network and operations, they will move towards their original vision for verification on transactions. This involves delegating verification to individual users who want to add transactions to the graph that constitutes the data structure of the network. Any user who wishes to add a transaction to this ledger has to verify two other transactions. IOTA proponents claim that this leads to a network that is stronger and faster the more use it sees. This is not yet implemented because the IOTA foundation concluded that a 51%

attack on this scheme would, at the current time, be too easy to carry out. The new scheme would be very reliant on traffic, as one can imagine a scenario where the amount of traffic is low, and thus confirmation will be slow. This is a crucial difference between Nano and IOTA confirmation, where Nano will slow down under high load, and IOTA will, in theory, speed up. Nano currently has the advantage of actually operating at their desired protocol level and testing it, while IOTA is limited to theory at the current time.

6.4 Time to Stability

Time to stability tries to put a time measure which serves as a probabilistic bound on when it is unlikely that a transaction will be overturned. In blockchains, varying degrees of certainty are the best guarantees available, since it is impossible to completely rule out the possibility of a longer chain being constructed and invalidating older transactions. In DAG systems, this is treated somewhat differently, as discussed in the experiments section of Chapter 3.

Nano is set in this table as *Unclear* due to its problematic relationship with stability. Currently, contrary to the claims in its whitepaper, the Nano network votes on all transactions in the system. If a transaction is voted to be trusted, all nodes actively observing the vote will take note of this and add the transaction to its copy of the ledger. However, the record of the vote is never stored anywhere and thus discarded. This is problematic since nodes who did not observe the first vote have no way of knowing which older transactions were voted for and which were not. This is mainly a problem for new nodes starting up, and downloading transactions in bulk from other nodes. The result of this design is that new nodes will call new votes for old transactions already deemed trusted by the network. This leads to unwanted traffic in the system, but more importantly, it could lead to different elections of the same transaction having different outcomes.

Since a node will start a new election of an old transaction, there is no guarantee that the election will have the same result as the original one carried out some time in the past. One could argue that it is probable since the likelihood of nodes changing their votes in the time between votes is rather low as no conflicting transactions can have been added to the ledger in the meantime. No research has been done on the probability of these outcomes, and no definite conclusion can thus be made.

As IOTA has for the coordinator, Nano has plans ready to remedy this situation of resource waste and uncertainty. In the next major Nano update, v19 planned for release during the summer of 2019, Nano will add a field that keeps track of what transactions on an account that has been deemed the winner of an earlier vote[72]. This concept is dubbed *confirmation height* and will be an integer telling users how

many blocks have been deemed trustworthy on this chain. The integer will start from 0 and increase with the block count, meaning that the creation block for the account chain will be 0, the first transaction will be 1 and so on. This measure will significantly limit unnecessary traffic in the network, and by doing that it will pave the way for even better scalability. As always in this field, one will have to wait for the actual release and test of this feature before one gets sold on the idea. Projects have proven again and again that promises and deliveries are not the same things, and thus, one should withhold judgment until the actual improvement has been released and not merely promised.

IOTA differs from other systems in this metric due to not having to rely on probabilities when the stability of transactions are concerned. IOTA confirmations rely solely on the coordinator referencing, either directly or indirectly, the transactions that are to be considered verified. This confirmation cannot be overturned, and thus confirmations in IOTA are irreversible. This leads to time to stability being a more robust metric in IOTA than in other systems which rely on probabilities. IOTA thus has the same time to stability as the time to initial confirmation of transactions.

6.5 Challenges faced by DAG systems

After a rather lengthy discussion of how DAG's can be utilized to solve or remedy issues faced by the current blockchain implementations, it only fits to in turn comment on what new problems this change in architecture can lead towards. In this section, a few of these problems will be introduced and briefly discussed. As demonstrated by the table of results displayed in Section 5.9, most of the technical aspects considered are deemed in favour of the DAG systems.

6.5.1 Lack of Incentives

One of the main criticism of the systems most considered in this thesis, IOTA and Nano, is their lack of incentives for outside agents to run nodes in their network. Running a node is a cost for those operating them, either in terms of monetary value directly in terms of server rental or indirectly in time or energy cost. In systems like Bitcoin and Ethereum, miners are incentivized through the minting of new currency and the collection of fees to help operate the system. Since no new currency is minted and no fees are collected in neither IOTA nor Nano, this kind of payment to benefactors is not possible. The big criticism is thus that there is no reward for supporting the network and making it stronger, and this leads to a slower and weaker network over time. It is argued that one cannot rely solely on community members and potential users of the network to keep it running and healthy.

The DAG proponents respond by comparing the cost of running a node in the network to those businesses have to pay to be able to accept debit and credit cards. These fees will generally scale with the amount of payment accepted, while the cost of running a node in networks will remain more or less the same. The critics fear that no business wanting to accept Nano or IOTA as payments are required to run a node. Thus many businesses will probably choose not to do so. If this becomes the majority, the network may face scaling issues down the road.

Different schemes to remedy this lack of incentives in the prominent DAG systems have been proposed, but it seems impossible to integrate into the protocols without sacrificing key characteristics. Monetary incentives must come in the form of collected fees or in the minting of new coins that can be distributed to the benefactors. Fees would compromise the feeless nature of these systems, which are especially crucial to IOTA who wants to do continuous machine to machine micro transactions which could become impractical if fees are introduced. Minting new coins is analogous to introduce inflation into the systems, which would decrease the value of all other coins in circulation. A primarily monetary system like Nano, which is intended as a pure currency with no smart contract integration, do not want this kind of inflation in their protocol. Thus the problem of incentives is not solved at this point, and no proposed solution seems to gather traction. It is again noted that this is not a problem intrinsic to DAG solutions, but more of a common problem for some of the solutions that have popped up utilizing the DAG as their data structure.

6.5.2 Lack of Experience

The somewhat recent rise of this DAG based technology is also commonly used to critique the protocols described in this thesis. This is also sometimes extended to the teams behind the protocols, especially the IOTA team which has caught a lot of criticism for some of their early development decisions.

Nano and IOTA, launched in February of 2016 and October in 2015 respectively, are substantially newer systems than Bitcoin, which launched in 2008. This in itself does not warrant legitimate criticism, but it does lead to proponents of older system pointing out the lack of experience from the newer ones. This is especially true when monetary systems are concerned since they will by default be a high-value target of attacks. Older systems that have avoided attacks for a more extended period will seem more trustworthy than newer ones who might not have done so for the same period of time. This criticism is sound for consumers and users who have to decide which systems to trust with their assets, but they do not carry any academic weight in that they do not make the systems inherently flawed. This lack of trust and experience may decrease given time, given that there are no significant incidents.

When the teams are concerned, IOTA especially has caught some condemnation

for some of their decisions and handling of criticism. The most significant controversy started when IOTA early on chose not to implement any existing and proven hashing algorithm and instead chose to design and implement their hashing scheme. This is of itself a huge red flag for anyone familiar with cryptography, as noted by Narula from the MIT Media Lab [33]. The second problem with this choice is that collisions were found in the hashing algorithm, which is proven to be a security issue for any cryptographic hashing algorithm. All hashing algorithms contain collisions by default, but they should be tough to find in a secure implementation. A group of researcher from MIT, Harvard and Boston University has conclusively shown that the hash function IOTA originally used, *curl-p*, contains collisions and thus is not secure to use in a cryptographic context [11].

The response from the IOTA team was first not very trust-inducing, claiming that the collisions were on purpose to inhibit others from stealing their design. They did, however, replace the hashing algorithms, and this attack is no longer possible. IOTA also argued that Curl is not a cryptographic hash function, and should thus not have to adhere to the same standards as other cryptographic functions [71]. They also argue that the collision is not a problem since the coordinator will remedy this situation before it escalates to an actual fault in the system. This is what would deter copy cats since they could not copy the then closed, now open source coordinator and thus would have a genuine fault caused by hashing collisions. These explanations raise some valid points, but the community was still not convinced by them, and this has since been a criticism raised against IOTA. Both the mistake in the first place and their responses to the criticism are still brought up regularly.

6.5.3 New Attack Vectors

There is also criticism of DAG systems when potential attacks are concerned. This is due to the new architecture not necessarily eliminating old attacks, but still introducing new ones. This increases the number of attack vectors into the system, making the security resource consuming and the attack surface larger.

As laid out in section 4.3, most of the attacks usually seen in blockchain are also applicable in DAG systems, if not in their original form then a slightly modified version exists. Examples of this are the 51% attack which, in modified versions, exist in all systems described in this thesis as laid out in Section 4.3.2. Both Nano and IOTA are currently susceptible to spam attacks, where attackers leverage the no-fee attribute of the systems into an opportunity to send thousands of transactions for free to slow the system down. Especially Nano is vulnerable to this since they facilitate precomputed PoW for their transactions.

Both systems do however have plans in place on how to mitigate this down the line. Nano will do this by moving to a dynamic PoW model, where the PoW required to

send a transaction will change based on the network state at the time of transmittal. This means that if a node senses high congestion in the network, it will raise the threshold for PoW needed, and this anyone with a stash of precomputed PoW would have to recalculate them or wait until the network congestion has decreased [68]. When or if IOTA removes the coordinator from their system, spamming it will not be a valid attack vector. Spam transactions will then confirm other transactions, speeding up the stability and confirmation times in the system. This is why IOTA claims that spam attacks will generally increase the security and usability of the system. Spam attacks are new to these DAG systems, due to their fee-less nature. This attack would not be advisable in, for example, Bitcoin, where the fees alone would deter attackers from even attempting it.

Chapter 7
Conclusion

This thesis set out to investigate the properties of DAG-based cryptocurrency systems and to investigate their scalability when compared to blockchain systems. It started with an extensive literary study, serving as an introduction to the different systems and the technology implemented in them. After the initial study, candidates for comparisons were chosen, and a more in-depth study of these was conducted. This resulted in Bitcoin and Ethereum being chosen as representatives for the blockchain systems in the comparison. Bitcoin was chosen due to being the first cryptocurrency, with a lot of name recognition and research about it. Ethereum was chosen as it is regarded as one of the most advanced blockchains, with a lot of development and support behind it. For DAG, multiple systems were researched and considered, but the final choice of systems to compare landed in Nano and IOTA. Nano was chosen for its broad claims and active community, while IOTA has a large organization and many corporate partners and supporters. Both of the systems also implement DAGs in different ways.

In order to prepare for comparison, research was done into what the most pressing issues facing blockchains today were. After identifying some areas for research, scalability was chosen as the most fitting for a direct comparison. This was partly due to the claims of the DAG systems were very scalability focused, and that this field offers a lot of detailed metrics for comparison purposes. To have a good base for comparison, metrics were chosen, and relevant data on the blockchain systems were gathered from third party sources. This was not too challenging due to the nature of the systems and the existence of many services dedicated to tracking them.

To gather data on the DAG systems, more direct steps had to be taken. The ecosystem around Nano and IOTA is less evolved than those surrounding Bitcoin and Ethereum, making some data harder to access. For the data that could be measured, nodes were initialized and operated to observe and measure the network. For historical data, third-party sources had to be consulted. This effort concluded in a table with different scalability metrics for all four chosen systems. This table and

the discussion following it was the main finding of the thesis.

7.1 Contribution

This thesis' main contribution has been the gathering and summarizing of information, as well as offering a new perspective on different facets of the systems in question. The thesis lays out the theory behind the systems concerned, how they perform and comments on different interpretations of both the behaviour and technology choices of the systems. The metrics identified and the table of results, as can be seen in Table 5.2, is also one of the main contributions of this work. The gathering, representation and justification of the metrics and the discussion of DAG as opposed to a blockchain are what can be built upon by further work.

The first research questions posed in the introduction of this thesis reads as follows:

> ***Research question 1: Can DAGs be used to solve the scaling issues faced by blockchain based systems?***

Through the work in this thesis, it is shown that it seems like DAG can at least scale *better* than the current blockchain solutions. It cannot be concluded whether DAG systems can scale as far as is needed for cryptocurrencies to be adopted as either a currency or, as IOTA aims to, for the internet of things economy. The systems need a more extensive and more robust history of operation, as well as better testing under stress and high load. It is also worth noting that it cannot be concluded that the better scalability properties of DAG systems are worth the sacrifices it makes in other regards. This trade-off must be assessed by the individuals looking to utilize the system, as to what properties it deems most valuable in a cryptocurrency system.

The second research question posed was:

> ***Research question 2: What new attacks and challenges do DAG systems introduce?***

The thesis answers this by reviewing and discussing both attacks that the DAG systems are vulnerable to and the new challenges they bring into the cryptocurrency space. This spans both general attacks on the system, as well as specific problems with the different protocols. The main finding in this regard is that DAG does not eliminate any attacks in particular, and seem to introduce new vulnerabilities and challenges. These will need to be incorporated when considering whether to use DAG or blockchain for future applications.

7.2 Future work

To advance the knowledge in this field, it is the conclusion of this thesis that the areas of security and robustness are the most wanting of more research. As commented earlier in this conclusion, the scalability of DAG-based systems are not confirmed to be enough to scale to a global economy, but it is found that they are indeed shown to scale better than blockchain. Since this is progress in itself, the scalability question is at this point less interesting than other challenges faced by these systems.

DAG-based systems are mostly in need of time and experience, enabling a maturity increase for the solutions. This would serve to both help remove scepticism and build trust around the systems. Further academic research is also something that would greatly help the systems along the path to adoption and maturity. As is concluded in this thesis, DAG systems certainly warrant further research, and a useful extension of this work could be to look more closely at weaknesses in DAG solutions.

7.3 Summary

The findings of this thesis are that a DAG can serve to replace or compliment blockchain in cryptocurrency systems. The DAG systems warrant more work and research in order to further expand the knowledge in this space. This study was useful to provide insight into different DAG-based systems and their internal mechanisms, and thus gain an understanding of the rationale behind their design. The thesis focuses on traits that stem from the data structure, but it is useful to note that there are a large number of other differences between cryptocurrencies. Internal in the groups of both DAG and blockchain systems there is plenty of differences that are not tied to the data structures. The thesis concludes that DAG systems offer better scalability than their blockchain counterparts, but that trade-offs are made to facilitate this. It will be up to the users of the systems to assess whether this trade-off is worth it for individual applications.

References

[1] A. Back et al. Hashcash-a denial of service counter-measure. 2002.

[2] A. Beikverdi and J. Song. Trend of centralization in bitcoin's distributed network. In *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2015 16th IEEE/ACIS International Conference on*, pages 1–6. IEEE, 2015.

[3] R. Böhme, N. Christin, B. Edelman, and T. Moore. Bitcoin: Economics, technology, and governance. *Journal of Economic Perspectives*, 29(2):213–38, 2015.

[4] B. Bollobás. *Modern graph theory*, volume 184. Springer Science & Business Media, 2013.

[5] A. Churyumov. Byteball: A decentralized system for storage and transfer of value. *URL https://byteball. org/Byteball. pdf*, 2016.

[6] A. Churyumov. Totally new consensus algorithm + private untraceable payments, https://bitcointalk.org/index.php?topic=1608859.0, Sept 2016.

[7] A. Gervais, G. O. Karame, V. Capkun, and S. Capkun. Is bitcoin a decentralized currency? *IEEE security & privacy*, 12(3):54–60, 2014.

[8] R. Grinberg. Bitcoin: An innovative alternative digital currency. *Hastings Sci. & Tech. LJ*, 4:159, 2012.

[9] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg. Eclipse attacks on bitcoin's peer-to-peer network. In *24th {USENIX} Security Symposium ({USENIX} Security 15)*, pages 129–144, 2015.

[10] E. Heilman, N. Narula, T. Dryja, and M. Virza. Iota vulnerability report: Cryptanalysis of the curl hash function enabling practical signature forgery attacks on the iota cryptocurrency. 2017.

[11] E. Heilman, N. Narula, G. Tanzer, J. Lovejoy, M. Colavita, M. Virza, and T. Dryja. Cryptanalysis of curl-p and other attacks on the iota cryptocurrency.

[12] C. Lemahieu. Cryptocurrency's killer app: Raiblocks micropayments, https://bitcointalk.org/index.php?topic=1381323.0, Feb 2016.

[13] C. LeMahieu. Nano: A feeless distributed cryptocurrency network. *nano. org. Accessed 03.10.2018*, 3, 2018.

[14] S. Meiklejohn. Top ten obstacles along distributed ledgers path to adoption. *IEEE Security & Privacy*, 16(4):13–19, 2018.

[15] J. MICHAEL, A. COHN, and J. R. BUTCHER. Blockchain technology. *The Journal*, 2018.

[16] A. Miller, A. Kosba, J. Katz, and E. Shi. Nonoutsourceable scratch-off puzzles to discourage bitcoin mining coalitions. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 680–691. ACM, 2015.

[17] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[18] K. J. O'Dwyer and D. Malone. Bitcoin mining and its energy footprint. 2014.

[19] S. Popov. The tangle. *cit. on*, page 131, 2016.

[20] Satoshi Nakamoto. Github commit implementing a maximum block size. `https://github.com/bitcoin/bitcoin/commit/172f006020965ae8763a0610845c051ed1e3b522`, 2010. [Accessed: Jan 19].

[21] Satoshi Nakamoto. Github commit introducing a maximum block size. `https://github.com/bitcoin/bitcoin/commit/a30b56ebe76ffff9f9cc8a6667186179413c6349`, 2010. [Accessed: Jan 19].

[22] Y. Sompolinsky, Y. Lewenberg, and A. Zohar. Spectre: A fast and scalable cryptocurrency protocol. *IACR Cryptology ePrint Archive*, 2016:1159, 2016.

[23] Y. Sompolinsky and A. Zohar. Phantom. 2018.

[24] M. Swan. *Blockchain: Blueprint for a new economy.* " O'Reilly Media, Inc.", 2015.

[25] K. Thulasiraman and M. N. Swamy. *Graphs: theory and algorithms.* John Wiley & Sons, 2011.

[26] Visa. Visa inc. at a glance. `https://usa.visa.com/dam/VCOM/download/corporate/media/visanet-technology/aboutvisafactsheet.pdf`, 2015. [Accessed: Jan 19].

[27] M. Vukolić. Rethinking permissioned blockchains. In *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, pages 3–7. ACM, 2017.

[28] Web. Bitcoin blockchain size. `https://www.blockchain.com/charts/blocks-size`. [Accessed: Jan 19].

[29] Web. Bitcoin network hash rate. `https://www.blockchain.com/charts/hash-rate?timespan=all`. [Accessed: Jan 19].

[30] Web. Bitcoin network statistics for the last 24h. https://www.blockchain.com/stats. [Accessed: March 19].

[31] Web. Block throughput in the nano network. https://www.nanode.co/blocks. [Accessed: April 19].

[32] Web. Blockchain trilemma. https://github.com/ethereum/wiki/wiki/Sharding-FAQs#this-sounds-like-theres-some-kind-of-scalability-trilemma-at-play-what-is-this-trilemma-and-can-we-break-through-it. [Accessed: Jan 19].

[33] Web. Blog post by nerula, detailing problems with iota. https://medium.com/@neha/cryptographic-vulnerabilities-in-iota-9a6a9ddc4367. [Accessed: May 19].

[34] Web. Byteball offical website. https://obyte.org/. [Accessed: March 19].

[35] Web. Confirmed transactions in the bitcoin network by date. https://www.blockchain.com/charts/n-transactions?timespan=all. [Accessed: April 19].

[36] Web. Confirmed transactions in the ethereum network by date. https://etherscan.io/chart. [Accessed: April 19].

[37] Web. Curl command line tool. https://curl.haxx.se/docs/manpage.html. [Accessed: March 19].

[38] Web. Daglabs project. https://blog.daglabs.com/. [Accessed: March 19].

[39] Web. Digitalocean. https://www.digitalocean.com/. [Accessed: May 19].

[40] Web. Digitalocean hosting options. https://www.digitalocean.com/pricing/. [Accessed: March 19].

[41] Web. Ethereum blockchain size pruned. https://etherscan.io/chartsync/chaindefault. [Accessed: March 19].

[42] Web. Ethereum full blockchain size unpruned. https://etherscan.io/chartsync/chainarchive. [Accessed: March 19].

[43] Web. Ethereum gas statistics. https://ethgasstation.info/. [Accessed: March 19].

[44] Web. Ethereum historical gas limit per block. https://etherscan.io/chart/gaslimit. [Accessed: March 19].

[45] Web. Ethereum historical time between blocks. https://etherscan.io/chart/blocktime. [Accessed: March 19].

[46] Web. Ethereum network statistics. https://etherscan.io/chartbi/transactionhistory. [Accessed: March 19].

[47] Web. Ethereum stability blog post. https://blog.ethereum.org/2015/09/14/on-slow-and-fast-block-times/. [Accessed: March 19].

[48] Web. External stress test of nano network. https://medium.com/@bnp117/stress-testing-the-raiblocks-network-part-ii-def83653b21f. [Accessed: May 19].

[49] Web. First developer stress test iota network. https://twitter.com/domschiener/status/858379721029111808. [Accessed: May 19].

[50] Web. Hash rate of different mining pools. https://www.blockchain.com/en/pools. [Accessed: Jan 19].

[51] Web. How to run a nano node. https://medium.com/@brunoerg/how-to-run-a-nano-node-c18aacf936da. [Accessed: March 19].

[52] Web. Iota 3 month historic throughput. http://tanglebeat.com/page/network. [Accessed: May 19].

[53] Web. Iota node quickstart tutorial. https://iri-playbook.readthedocs.io/en/master/index.html. [Accessed: March 19].

[54] Web. Iota official faq. https://www.iota.org/get-started/faqs. [Accessed: Feb 19].

[55] Web. Iota official glossary. https://iota.readme.io/docs/glossary. [Accessed: March 19].

[56] Web. Iota official homepage. https://www.iota.org/get-started/what-is-iota. [Accessed: May 19].

[57] Web. Iota official medium post on the coordinator. https://blog.iota.org/coordinator-part-1-the-path-to-coordicide-ee4148a8db08. [Accessed: Feb 19].

[58] Web. Iota pow as a remote service. https://powsrv.io/. [Accessed: Feb 19].

[59] Web. Iota tip selection. https://docs.iota.org/docs/the-tangle/0.1/concepts/tip-selection. [Accessed: May 19].

[60] Web. Measurements from the iota coordinator. http://coordinator.iotawatch.it/. [Accessed: April 19].

[61] Web. Measurements of nano propagation and confirmation times. https://www.repnode.org/network/propagation-confirmation. [Accessed: Feb 19].

[62] Web. Median confirmation time for transactions with a non zero fee in bitcoin. https://www.blockchain.com/charts/median-confirmation-time?daysAverageString=7. [Accessed: March 19].

[63] Web. Nano network measurement statistics. https://nanocrawler.cc/network. [Accessed: April 19].

[64] Web. Nano releasenotes for v11 of the protocol. https://hackernoon.com/bitconnect-anatomy-of-a-scam-61e9a395f9ed. [Accessed: May 19].

[65] Web. Nano releasenotes for v11 of the protocol. https://github.com/nanocurrency/nano-node/releases/tag/V11.0. [Accessed: May 19].

[66] Web. Nano rpc interface commands. https://github.com/nanocurrency/nano-node/wiki/RPC-protocol#wallet-ledger. [Accessed: March 19].

[67] Web. Nano transaction from online wallet to node account. https://www.nanode.co/block/9027CCEEC6563A5E21FB1A6BA230A5F48562B170D18CF6E5459A09FF3E22427B. [Accessed: March 19].

[68] Web. Nano v19 release notes, including dynamic pow. https://medium.com/nanocurrency/v19-solidus-feature-analysis-3c8f3d2d949c. [Accessed: May 19].

[69] Web. Nanode, a nano block explorer. https://www.nanode.co/. [Accessed: April 19].

[70] Web. Number of waiting transactions in the bitcoin network. https://www.blockchain.com/charts/mempool-count?timespan=all. [Accessed: April 19].

[71] Web. Official iota response to curl criticism. https://blog.iota.org/official-iota-foundation-response-to-the-digital-currency-initiative-at-the-mit-media-lab-part-4-11fdccc9eb6d. [Accessed: May 19].

[72] Web. Official nano medium post on votes and confirmation. https://medium.com/nanocurrency/looking-up-to-confirmation-height-69f0cd2a85bc. [Accessed: April 19].

[73] Web. Official nano website. https://nano.org/en. [Accessed: Feb 19].

[74] Web. Proof of work time and energy consumption in iota. https://hackernoon.com/lessons-learned-from-evaluating-iota-on-internet-of-things-devices-a44575e606de. [Accessed: April 19].

[75] Web. Release notes of the bitcoin pruning functionality. https://bitcoin.org/en/release/v0.11.0#block-file-pruning. [Accessed: March 19].

[76] Web. Second developer stress test iota network. https://twitter.com/domschiener/status/852985008398774272. [Accessed: May 19].

[77] Web. Visa transaction stats. https://usa.visa.com/run-your-business/small-business-tools/retail.html. [Accessed: Jan 19].

[78] Web. Vnstat homepage. https://humdi.net/vnstat/. [Accessed: March 19].

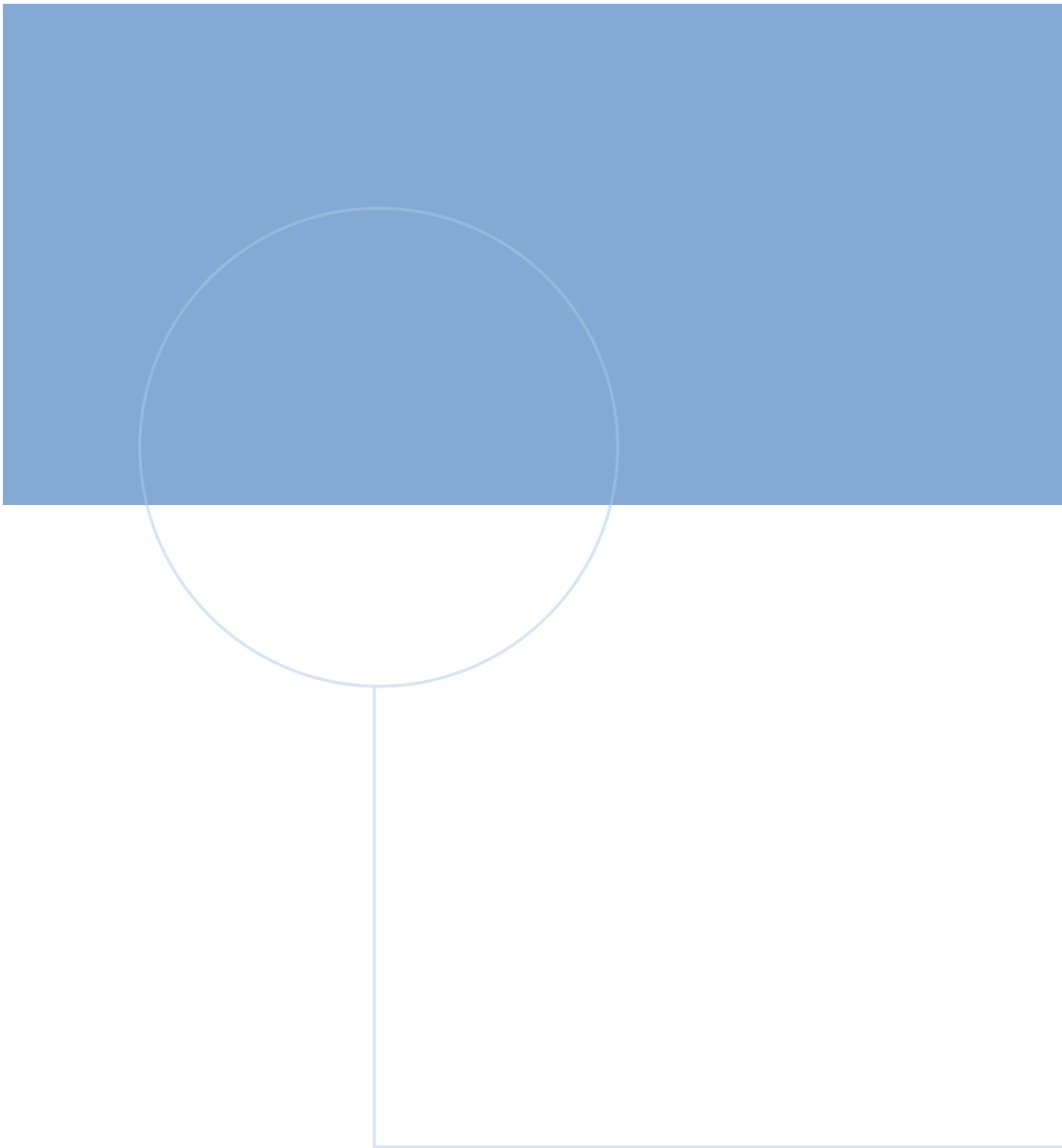
[79] Web. Iota announcement on bitcointalk.org, https://bitcointalk.org/index.php?topic=1216479.0, Oct 2015.

[80] Web. Smallest bitcoin transactions. `https://blockchair.com/bitcoin/transactions?q=&s=size(asc)#`, 2019. [Accessed: Jan 19].

[81] R. J. Wieringa. *Design science methodology for information systems and software engineering*. Springer, 2014.

[82] K. Yeow, A. Gani, R. W. Ahmad, J. J. Rodrigues, and K. Ko. Decentralized consensus for edge-centric internet of things: A review, taxonomy, and research issues. *IEEE Access*, 6:1513–1524, 2018.

[83] D. Yermack. Is bitcoin a real currency? an economic appraisal. In *Handbook of digital currency*, pages 31–43. Elsevier, 2015.



NTNU
Norwegian University of
Science and Technology