b
UNIVERSITÄT
BERN

Analysis of the Tangle

The Impact of Conflicting Blocks to the Tangle Network

Bachelor Thesis

Michael Brunner
Reichenbachstrasse 79
Bern, Switzerland

Faculty of Science, University of Bern

31.01.2021

Prof. Christian Cachin
Ignacio Amores Sesar
Cryptology and Data Security Group
Institute of Computer Science
University of Bern, Switzerland

Abstract

This thesis explains the tangle and shows results of its behaviour based on an implemented simulation.

Tangle is a blockchain protocol invented by IOTA, a public distributed ledger. This mechanism creates a directed acyclic graph, meaning that, contrary to “normal blockchain” it necessitates two parent blocks. Its properties, such as weight assignment of blocks, tip selection algorithm and confirmation confidence, are explained step by step in this thesis.

The results are based on different simulations to mirror the real world as closely as possible. The most important results include the following points.

The average amount of tips and the confirmation time change with the number of mining parties and their ability to react to the network. It requires a lot of new blocks to confirm an existing one, which is shown with an implemented simulation.

The network does not work properly with conflicting blocks. The simulation thereby depicts that there will be no confirmed blocks after the insertion of conflicting blocks.

While slow mining parties do not change the output significantly, they drastically slow down the network and delay the confirmation process.

Contents

1 Introduction

This work examines the tangle, a blockchain protocol invented by IOTA, a public distributed ledger. The protocol will be analysed in various ways whilst inspecting its efficiency, security as well as other critical factors. For better understanding, the theoretical part of this paper explains the tangle and provides some further background information.

The main part of this work includes an examination of the efficiency as well as the general structure of the tangle. A simulation was implemented to inspect elements, such as confirmation confidence and the number of unconfirmed blocks over a longer period of time. As there are many open parameters and rules, several different simulations are constructed to cover a variety of scenarios and assumptions. The goal of this approach was to mirror the tangle as realistically as possible.

Moreover, this work provides deeper insight into the confirmation time, which is a very important factor for efficiency. There is also a detailed analysis regarding the handling of conflicting blocks and why they could be problematic.

2 Theory

This chapter will describe the tangle and its function more precisely, as well as other topics, such as its structure, security and efficiency. Firstly, the basics of blockchain will be discussed and, secondly, specific parts of the tangle will be depicted. The hash function and the process of mining are mentioned shortly in the next section, however, they are not the core of this work.

2.1 Blockchain

This section briefly covers the basic elements of blockchains. Essential parts, such as what a block embodies and how the connection between two blocks works, are explained.

A blockchain is a decentralised database on different computers. Parties, people or companies with computers can mine new blocks with information cryptographically embedded within them.

Figure 2-1 shows a simplification of a block. A simple block consists of three parts: a header and a hash part of a previous block; transactions or other information, and; a free part, which is important for the hash function [1, p. 184].

Figure 2-1. Representation of a Block

In order to mine a new block, it needs to satisfy a given condition that involves a hash function. The last part is filled with random bits to try to solve the hash function puzzle. There are usually billions of tries necessary to find a possible solution.

$$Hash(header + last\ block + transactions + free\ part) < Target$$

This ensures that it takes a long time to mine a new block but only a short time to verify a given one.

Figure 2-2 shows a graphical representation of a blockchain. Each circle represents a block. As every block needs information about the previous it is always clear, which is the new one. Consequently, block 4 in figure 2-2 is the newest and block 1 is the oldest.

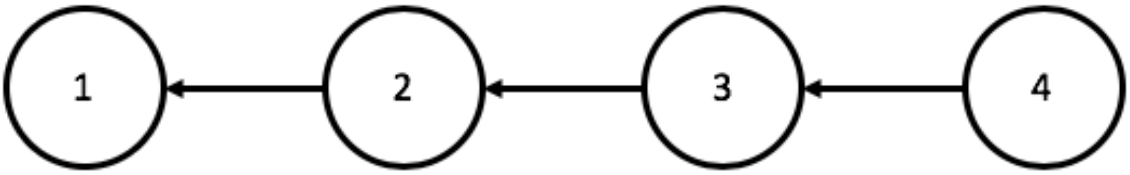


Figure 2-2. Blocks in a Blockchain

2.2 Directed Acyclic Graph (DAG)

Unlike most blockchain protocols, the tangle is not a chain but a Directed Acyclic Graph, which is why there are a lot more possibilities to arrange and visualise the structure of the blocks.

A graph consists of vertices, which are the blocks with the information, and edges from one vertex to another. The graph has no cycles due to the impossibility of pointing forward to a newer block which has not been made yet: it only points back to older vertices and is therefore acyclic [2, p. 2].

Figure 2-3 shows an illustration of what a simple tangle network may look like.

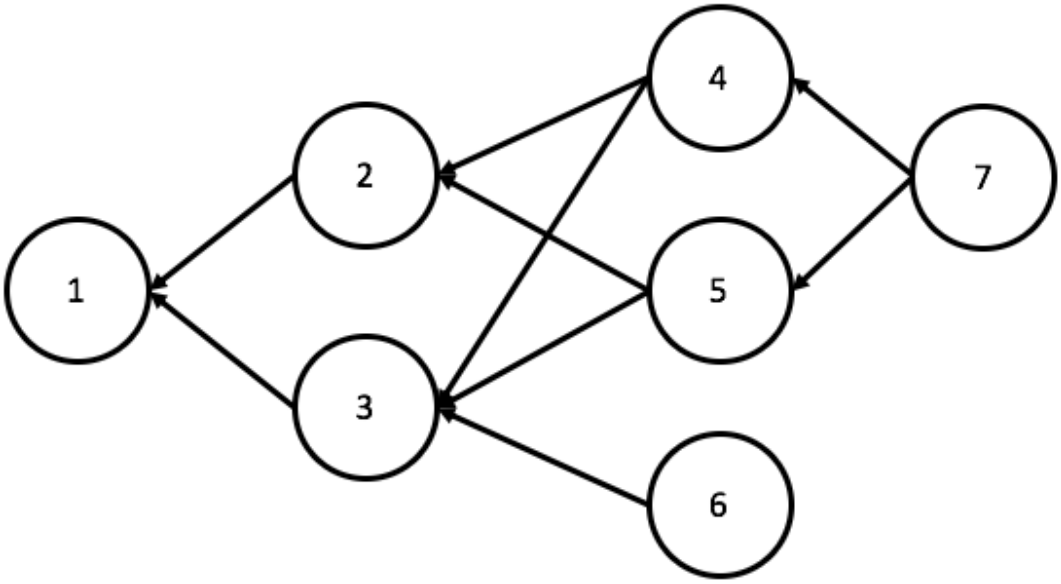


Figure 2-3. Illustration of a Simple Tangle Network

2.2.1 Two Parents

In the tangle, every block points to two previous blocks and contains information about both. It is however possible that those two blocks are the same [2, p. 3]. Hence, there can be blocks pointing back to two (in figure 2-3: block 4, 5 and 7) or just one block (block 2, 3 and 6). The only one which does not have any parent block is the genesis.

In most blockchain protocols, the most recently published block is set as a parent by default. In the tangle, the parents are chosen a bit differently, which will be further discussed in section 2.5.

2.3 Terms of Blocks

Blocks can have different attributes and names. This section will cover the most important terms used in this paper.

Parent blocks and *child blocks* are directly related to each other. Being the newer ones, child blocks contain information about the older ones, the parent blocks. The child blocks therefore point back to the parent blocks. For example, in figure 2-3, block 7 is the child block of blocks 4 and 5.

The *genesis* is the first block of the network and thereby has no parents. However, it is indirectly approved by every other block [2, p. 2], as it is connected to every other block.

Tips are blocks that have not yet been approved [2, p. 5]. They have no connection to a child block. Examples of tips are block 6 and 7 in figure 2-3.

A new block must *directly approve* two other blocks. If A directly approves B and B directly approves C, A indirectly approves C [2, p. 2]. In the case of a direct approval, the blocks are related as parent and child blocks. For example, in figure 2-3 block 6 directly approves block 3 and indirectly approves block 1, as represented by the arrow.

A block can either be *confirmed* or *unconfirmed*. There are different ways to confirm a block, which will be depicted in section 2.6.

2.4 Weight

In the tangle, every block has a particular weight. There are different methods to assign a block's weight. This section explains one proposed by Attias & Bramas in 2019.

For this method, every tip is assigned the weight one and that of all the other blocks is thereafter calculated using the following formula [3, p. 277]:

$$w(B) = 1 + \sum_{C \in c(B)} \frac{w(C)}{2}$$

The function $w(B)$ is used to calculate the weight of block B. $c(B)$ represents the set of all children of block B.

In words, block B's weight is equal to the sum of all of its children's weights divided by two, plus one.

As a general rule, no matter which algorithm is used for the weight assignment, the weight of a block increases with the number of approvals [2, p. 14].

2.5 Tip Selection Algorithm (TSA)

A new mined block needs to approve two previous blocks. There are different algorithms to select the two parent blocks. The network needs to make progress, this means new blocks should be confirmed as quickly as possible. In order to ensure the progress, new blocks need tips as their parents. Therefore, those algorithms are called "tip selection algorithm". A TSA needs to be run twice in order to get two new parents. It is possible to select the same tip twice, which means this block has only one parent.

There are a few different TSAs, but all have the following points in common: Firstly, the input is the graph with all its existing blocks and their connections. Secondly, the output is a single tip. And lastly, randomness, the algorithm can output different tips even with the same input.

In the following subsections, the most important TSAs will be explained with a short comparison at the end.

2.5.1 Uniform TSA

The uniform TSA selects a random tip with a uniform distribution, whereupon every tip has the same chance to be selected [4, p. 2]. The weights of the blocks and the arrangement of the graph have no effect on the output of the algorithm.

2.5.2 Random Walk

To visualise the random walk, a person standing at the genesis can be imagined. He walks along one of the edges, which he chooses randomly, until he arrives at a child block. The person continues this process until he reaches one of the tips. It uses a Markov Chain Monte Carlo simulation to output a tip. The probability of choosing a specific edge depends on the parameters used in the algorithm.

2.5.2.1 Unweighted random walk

In the unweighted random walk, the probability of choosing a specific edge is uniformly distributed [4, p. 2]. As the following formula displays, choosing a specific child C_i from block B only depends on the number of children of block B:

$$P_{B \to C_i} = \frac{1}{\sum_{C \in c(B)} 1}$$

The probability of choosing a particular tip does not depend on the weights of the blocks, but only on the structure of the graph.

2.5.2.2 Weighted random walk

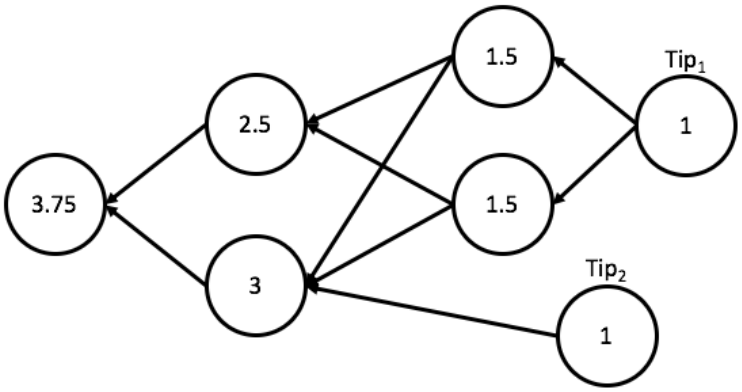
The same method used in the unweighted random walk can be used in the weighted random walk. The only difference is that the probability of choosing child C_i is given by the following formula [4, p. 3]:

$$P_{B \to C_i} = \frac{e^{-\alpha(w(B)-w(C_i))}}{\sum_{C \in c(B)} e^{-\alpha(w(B)-w(C))}}$$

In this formula, the weights of the blocks affect the output of the algorithm. The probability of choosing child C_i is higher if C_i has a greater weight. With α equal to zero, the weights have no effect at all. A larger α implies a greater impact from the weights.

2.5.3 Comparison

This subsection compares the three TSAs using an example to provide in-depth understanding of their mechanism. Figure 2-4 shows an example graph with two different tips. Table 2-1 displays the probability of choosing Tip_1 or Tip_2, comparing the uniform, the unweighted and the weighted tip selection algorithm. The numbers in the circles in figure 2-4 indicate the weight of each specific block. The weights are calculated with the method explained in section 2.4. Tip_1 and Tip_2 both have a weight of 1. The following calculations on the right-hand show how the remaining weights are determined.



$$1.5 = 1 + \frac{1}{2}$$

$$2.5 = 1 + \frac{1.5}{2} + \frac{1.5}{2}$$

$$3 = 1 + \frac{1.5}{2} + \frac{1.5}{2} + \frac{1}{2}$$

$$3.75 = 1 + \frac{2.5}{2} + \frac{3}{2}$$

Figure 2-4. Tangle Network with Weights

Algorithm	Tip_1	Tip_2
Uniform	0.5	0.5
Unweighted	0.83333	0.16667
$\alpha = 0.001$	0.83335	0.16665
$\alpha = 1$	0.855	0.145

Table 2-1. Comparison of TSAs

2.6 Confirmation

In order to execute the transactions within a block, the block needs to be confirmed. This can be performed using one of two different methods: the coordinator and the confirmation confidence. The following subsections explain the two methods.

2.6.1 Coordinator

The first method, used by IOTA to run the tangle, employs a coordinator [4, p. 5]. This entity sets a new block, the coordinator block, after a certain time. Its only function is to approve and confirm already existing blocks. The coordinator, which is operated by IOTA, allows to control the network, hence the tangle is currently not decentralised.

2.6.2 Confirmation Confidence

The second method is called confirmation confidence. In order to decentralise the tangle, IOTA plans to change the confirmation process at some future point in time.

The confirmation confidence algorithm works the following way: The TSA is executed 100 times in order to get a list with 100 tips, which may contain duplicates. A block is confirmed if it is approved, directly or indirectly, by more than 95 of those 100 tips [4, p. 5].

Figure 2-5 serves as an example to explain the confirmation confidence algorithm. The result of 100 TSAs: Tip_1 is chosen 82 times, while tip_2 is chosen 18 times. The confirmation confidence of the remaining blocks is equal to the sum of its approving tips.

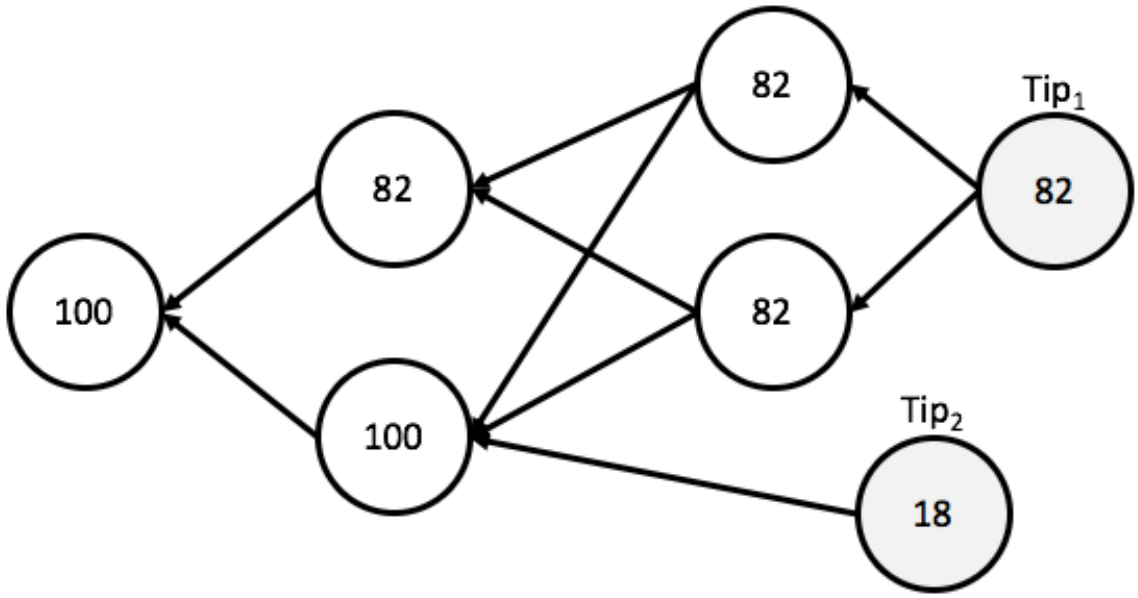


Figure 2-5. Confirmation Confidence Example

2.7 Conflicting Blocks

In a block, there are several transactions. It is possible that two transactions from two different blocks conflict with each other. This may happen when, for example, a person

spends 10 dollars in two different transactions, aggregating to 20 dollars, whilst only possessing 10 dollars. In this case, only one of those blocks can become confirmed. Otherwise, one of the receivers will not receive the intended money, as the spending person does not have enough money. Hence, the two blocks contradict each other.

A newly mined block cannot approve, neither directly nor indirectly, both of those conflicting blocks at the same time [2, p. 3]. Consequently, if there are two conflicting blocks, there is a fork in the network. In those cases, the tangle ensures that both of them do not become confirmed simultaneously [5, p. 16].

2.8 Security and Efficiency

This section depicts the properties of a secure and efficient network. The points in italic are examined in the simulations in the remainder of this work.

Firstly, an efficient network needs all of the following properties:

- A mining party needs to be able to analyse the existing graph with all blocks and their connections in a reasonable amount of time.
- The upload and integration of new blocks needs to be easy and quick.
- *A new uploaded block needs to be confirmed as quickly as possible.*

Secondly, for a secure network, all of the following properties must hold as follows:

- All the transactions in a block must be valid.
- *A once confirmed block must remain confirmed.*
- *Two conflicting blocks cannot both be confirmed.*
- *There shall be no old unconfirmed blocks, as they bring unpredictability to the network.*

2.8.1 Double-Spending Attack

To illustrate what an attack of an adversary looks like, the double-spending attack will shortly be discussed.

As an example, Alice makes two transactions of 10 dollars each: one to Bob, in block 1, and one to Charlie, in block 2. But Alice only has 10 dollars in total. If block 1 becomes confirmed, Bob thinks he received the money and returns the goods to Alice. After a while, block 1 becomes unconfirmed and block 2 becomes confirmed. Charlie

thereby receives the money and sends the goods to Alice as well. However, as there are only 10 dollars available, Bob and Charlie cannot both receive the money. As a consequence, Bob is left empty-handed and Alice receives both goods [2, p. 15].

3 Simulation and Methodology

In this chapter different models and simulations for the tangle are discussed. The system is analysed in terms of efficiency and security. There are a lot of different options and parameters; this analysis therefore dives into a variety of simulations to cover as many scenarios as possible. Section 3.1 is devoted to networks with no conflicting blocks.

3.1 Simulation in a Simple Network

The simulation and the necessary assumptions are firstly explained. With a simulation, it is possible to analyse a long time-period in a matter of seconds. This allows the evaluation of different states of every block in the tangle network.

Time is represented in rounds. Every round consists of three parts:

1. Parties evaluate the network (every block with its connections).
2. Parties try to mine new blocks.
3. New blocks are broadcasted to the network.

A block is an array with the following information:

- Block's name, usually a number
- Parent 1 and parent 2
- Weight, which needs to be updated in every round
- The round in which the block was mined

The Poisson distribution function is used to determine how many new blocks are mined every round.

Figure 3-1 serves as an example of a tangle after 30 rounds with $\lambda = 4$ and *delay* = 0. The blue block represents the genesis. The ones in green are already confirmed, while the yellow ones are not confirmed yet. The red blocks are the tips.

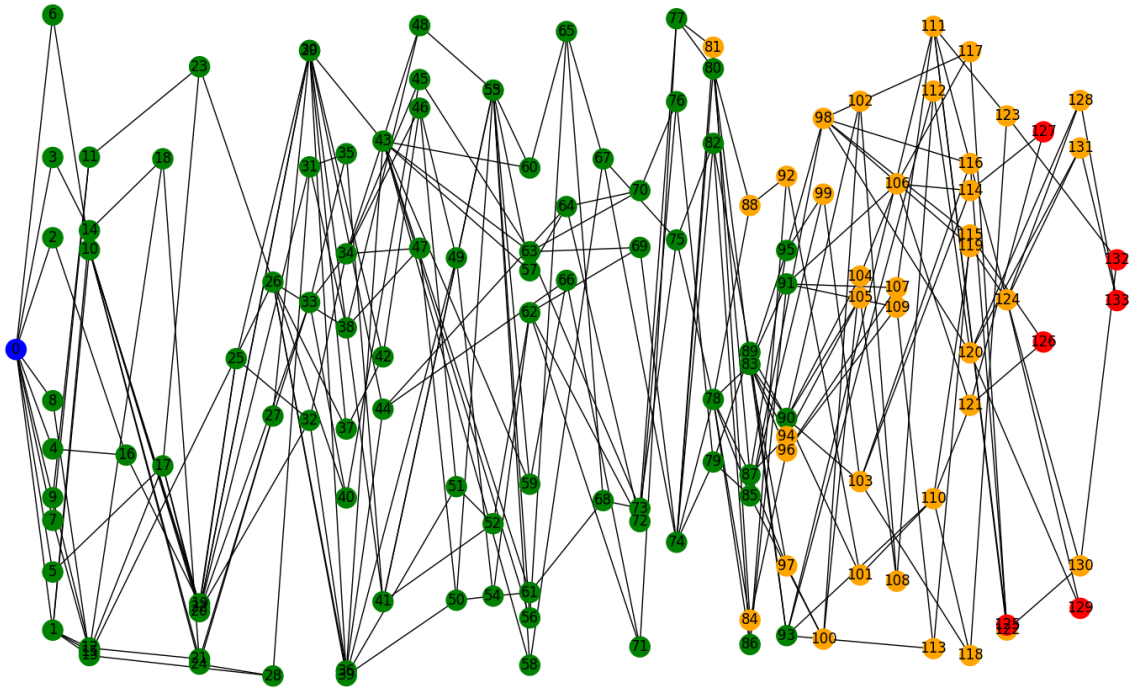


Figure 3-1. Example of a Tangle Simulation

It is not possible to understand the properties of a tangle by visual inspection. Therefore the analysis and results in the remainder of this work only focus on specific points.

3.1.1 Parameters

There are several variables which are important for the simulation:

λ is the product of the number of parties and the probability of mining for a specific party in a round. λ is therefore approximately equal to the average number of new blocks per round. The higher λ is, the more new blocks are mined per round. In the simulations in this work, λ is usually between 1 and 10.

α is used in the weighted random walk. It is usually set to 0.001, but flexible if there is something interesting to display.

A *delay* is used to simulate the mining process more accurately. For example, this will occur if party P checks the graph at time *t*, selects the parents for the new block and tries to find a solution for the hash function. P finds a solution at time *t* + *x*. But between *t* and *t* + *x*, new blocks may already have been attached to the tangle. Therefore, the *delay* parameter simulates time *x* and states how many rounds are not considered for the new block mined by P.

3.2 Simulations with Conflicting Blocks

In all simulations with conflicting blocks, the following parameters are standardised:

- Number of rounds: 1000
- *delay* = 0
- λ = 5
- α = 0.001

The main focus of simulations with conflicting blocks is to observe different patterns and the differences to simulations without conflicting blocks.

There are two different simulations to study the behaviour of conflicting blocks. One with only two conflicting blocks and one with more conflicting blocks.

3.2.1 Two Conflicting Blocks

In order to observe conflicting blocks, two conflicting blocks are inserted into the network in round 200 or short after. In the remainder of this work the initial conflicting blocks are called *origin*. During that time the tangle is already massive and stable with an immense number of blocks, but there are still enough rounds left to examine the impact of conflicting blocks.

To simulate a time difference between the mining of the two blocks, they can be inserted either in the same round or in different rounds. The time difference has to be small. Otherwise the later has no possible tips to mine on, as they already conflict the second block.

The choice of the parents for a conflicting block has a huge impact for the network. Therefore, it is necessary to repeat the simulation to observe different behaviourisms.

3.2.2 More Conflicting Blocks

To expand the model, another simulation is used. This allows to insert several groups of conflicting blocks at different points in time.

Figure 3-2 shows a simple example with 4 rounds. There are five different groups, indicated by the number in a block. While group 0 does not conflict with anything, group 1 conflicts with group 2 and group A conflicts with group B. However, it is possible for a block to belong to different groups, for example group 1 and group B.

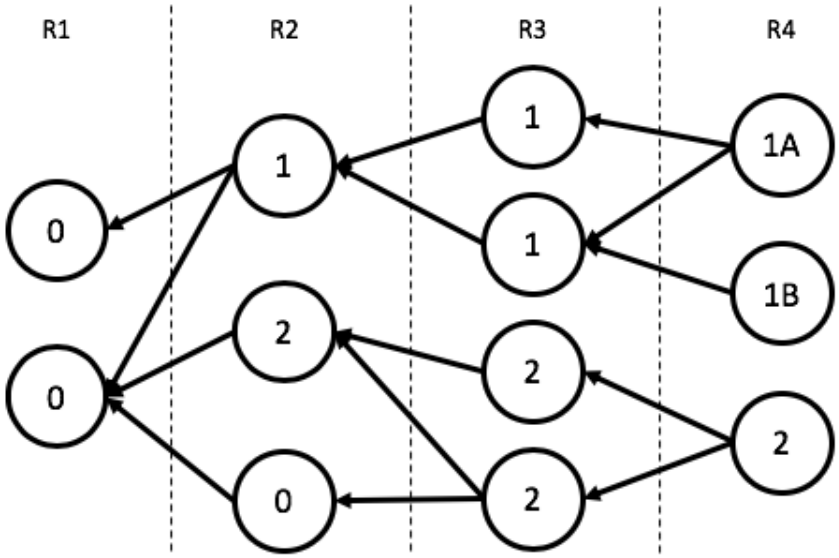


Figure 3-2. Network with More Conflicting Blocks

3.3 Analysis

At the end of each round, and at the end of the simulation, the graph will be analysed. For example, the confirmation confidence for each block after each round will be calculated and saved. This makes it possible to analyse the confirmation confidence over a given time-period.

3.3.1 Exact Confirmation Confidence

The algorithm to approximate the confirmation confidence involves 100 random walks and is binomially distributed. This may lead to inaccurate results. In the simulation, precise values are necessary to compare them with each other. However, it is possible to calculate the exact value for the confirmation confidence, which is used in all simulations due to its practicality. Subsection 4.4.2 provides an analysis regarding the difference between the exact value and the approximation value of 100 random walks.

4 Results

This chapter is divided into four different parts. The first section is devoted to networks with no conflicting blocks. Its main purpose is to study the network under the effect of changing parameters. The second and third part contain simulations with conflicting blocks and discuss the differences to simple network without conflicting blocks. The last section shows a variety of other results and observations.

4.1 Network without Conflicting Blocks

Performing the simulations with the same parameters twice leads to almost exactly the same results. This only applies to simulations with no conflicting blocks.

Every simulation is running through 1000 rounds of mining while the parameter α is set to 0.001. A longer simulation does not have a significant impact on the results other than increasing the computational time. The main focus of this section is to inspect the impact of the variables λ and *delay*.

4.1.1 Number of Tips

In this section, the number of tips is counted after each round. Table 4-1 shows that a change in λ and *delay* results in a different minimum, maximum and average number of tips.

delay	0	0	0	0	2	2	2	2
λ	1	2	5	10	1	2	5	10
Min	1	1	2	7	1	2	5	11
Max	6	8	14	27	14	22	39	70
Avg	2.2	3.3	7.7	14.7	6.3	11.9	27.8	53.6

Table 4-1. Number of Tips in a Simple Network

It is visible that a higher λ and a higher *delay* lead to more tips. Figures 4-1 and 4-2 illustrate the distribution of how many tips are in the network after each round.

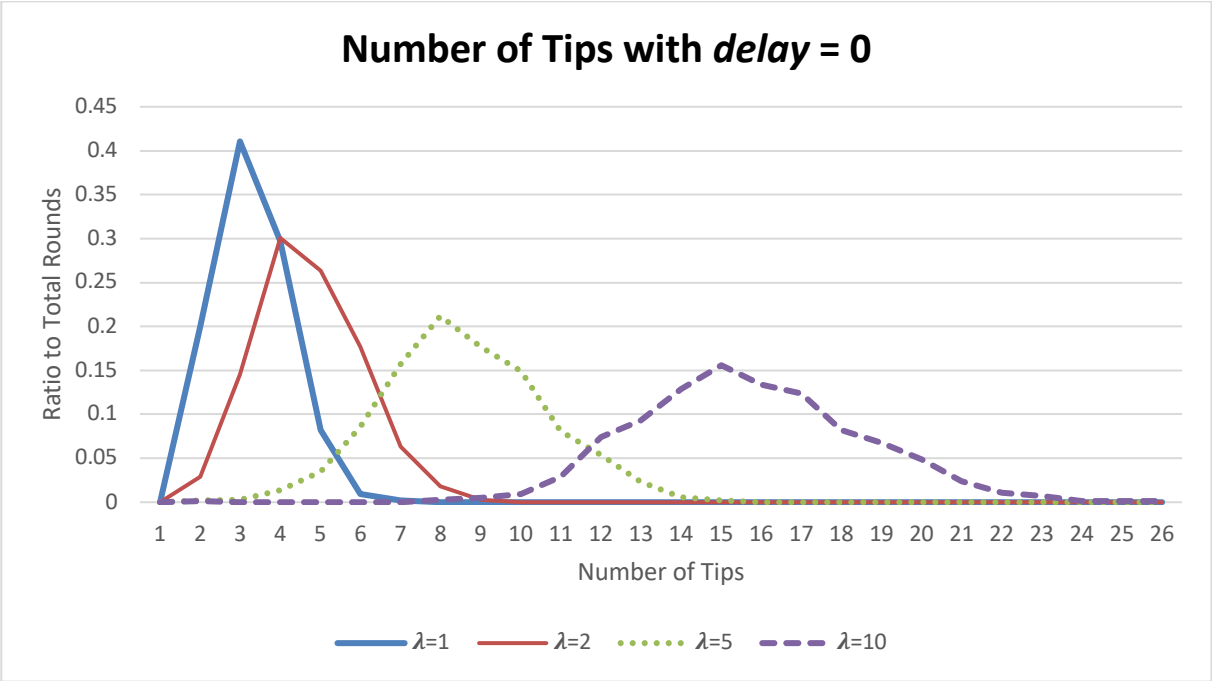


Figure 4-1. Number of Tips with delay = 0

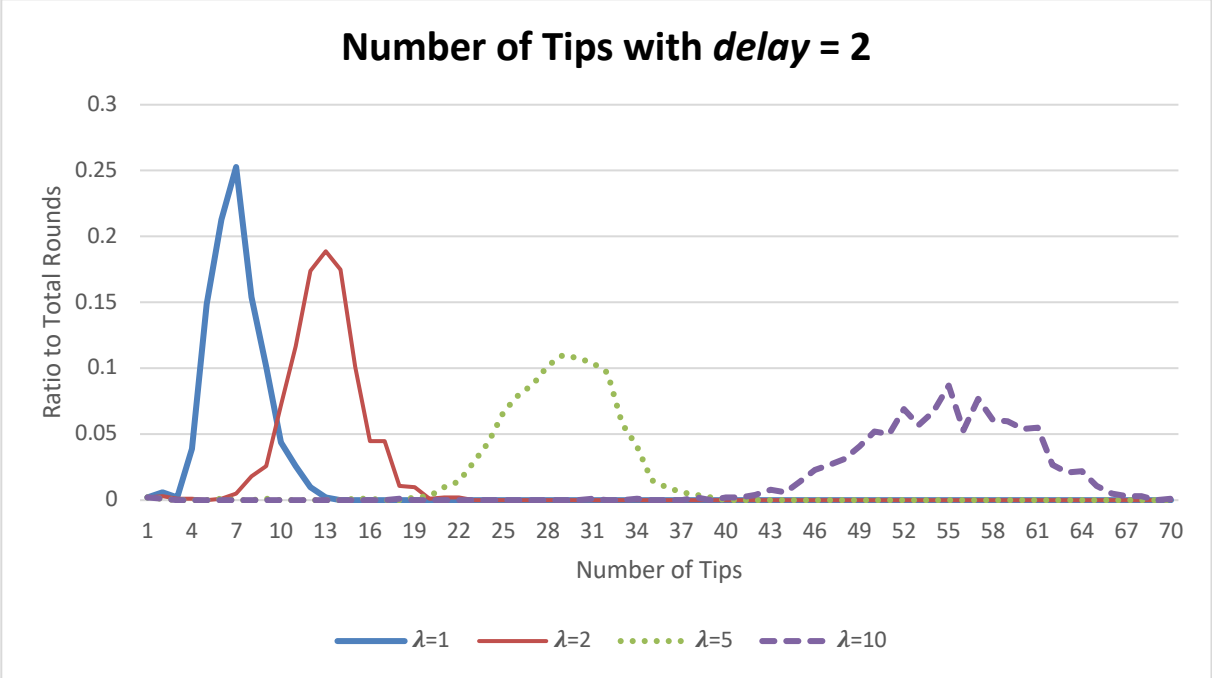


Figure 4-2. Number of Tips with delay = 2

While the number of tips does not have a direct impact on the network, it affects other results, which are presented in the remainder of this section.

4.1.2 Blocks with One Parent

This subsection explains how likely it is for a new block to select the same parent twice. The ratio is the number of blocks with only one parent divided by the total number of blocks. The likelihood of choosing the same parent twice decreases with a higher λ

and a higher *delay*. Obviously, this correlates with the fact of having more tips available to choose from.

delay	0	0	0	0	2	2	2	2
λ	1	2	5	10	1	2	5	10
Total	1003	1941	4901	9989	1024	2060	5049	10043
1 P	598	819	829	859	218	204	226	281
Ratio	.59	.42	.17	.09	.21	.10	.04	.03

Table 4-2. Blocks with One Parent

Subsection 4.4.4 will provide reasoning as to why blocks with only one parent are rather disadvantageous.

4.1.3 Development of Weights

Figures 4-3 and 4-4 visualise how the weights of blocks develop over time. Each line represents the weight of the 50th block in the network. In the simulation with a higher λ and a higher *delay*, the weights increase to a larger extent. However, in all of the simulations, the weights appear to be bounded from above.

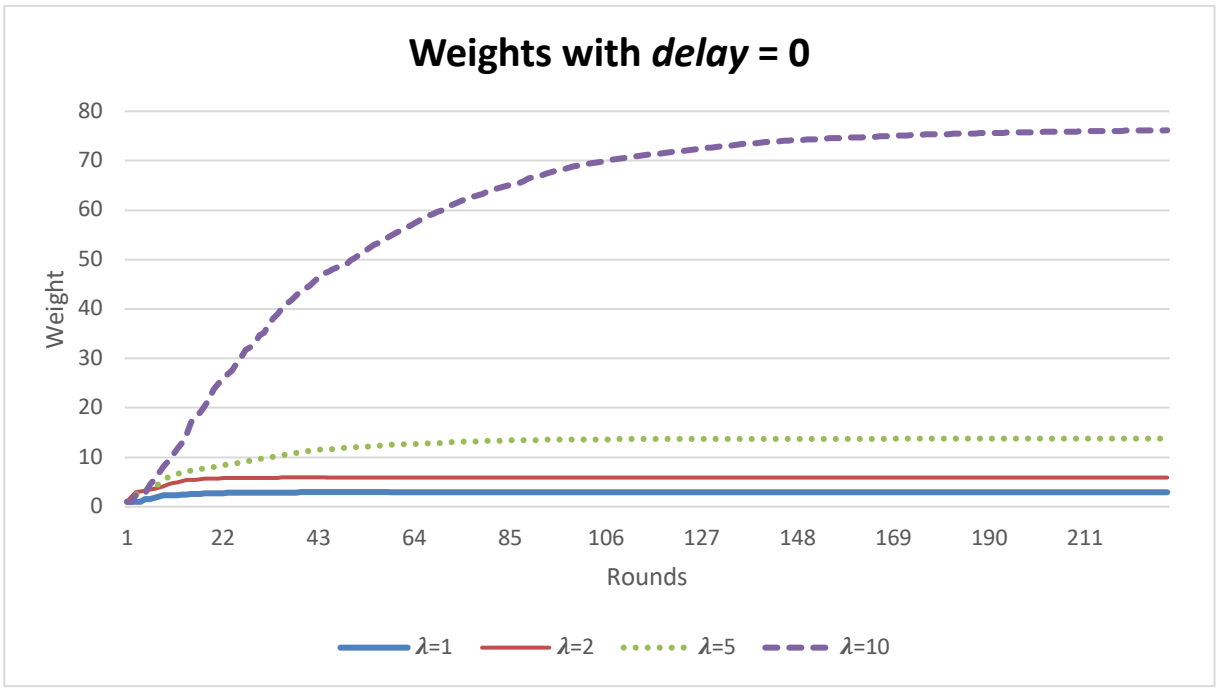


Figure 4-3. Development of Weights with delay = 0

Figure 4-4. Development of Weights with delay = 2

The fact that the weights are not changing significantly is neither good nor bad. However, the probability for a weighted random walk, and therefore the confirmation confidence for each block, stays the same after a few rounds. This becomes important in section 4.2, where simulations with conflicting blocks are discussed.

4.1.4 Confirmation Time

An important part for the efficiency of a network is the confirmation time. In other words, the time it takes to confirm new blocks is crucial.

Table 4-3 illustrates the average number of rounds and new blocks until a block becomes confirmed. The maximum number of rounds, which represents the worst-case scenario, is listed as well.

delay	0	0	0	0	2	2	2	2
λ	1	2	5	10	1	2	5	10
Max_R	35	38	36	25	88	89	102	95
Avg_R	5.0	5.6	7.3	8.3	28.7	33.6	39.4	43.8
Avg_B	5.0	11.2	36.5	83	28.7	67.2	197	438

Table 4-3. Confirmation Time in a Simple Network

The average amount of rounds increases drastically with a higher *delay*. It becomes clear that it takes a significant number of rounds to receive confirmation.

Figures 4-5 and 4-6 display the distribution of rounds needed until a new block gets confirmed. It shows that a change of λ only has little impact on the confirmation time. Hence, it takes more or less the same amount of time for most blocks to be confirmed if all parties are mining quickly and broadcasting immediately.

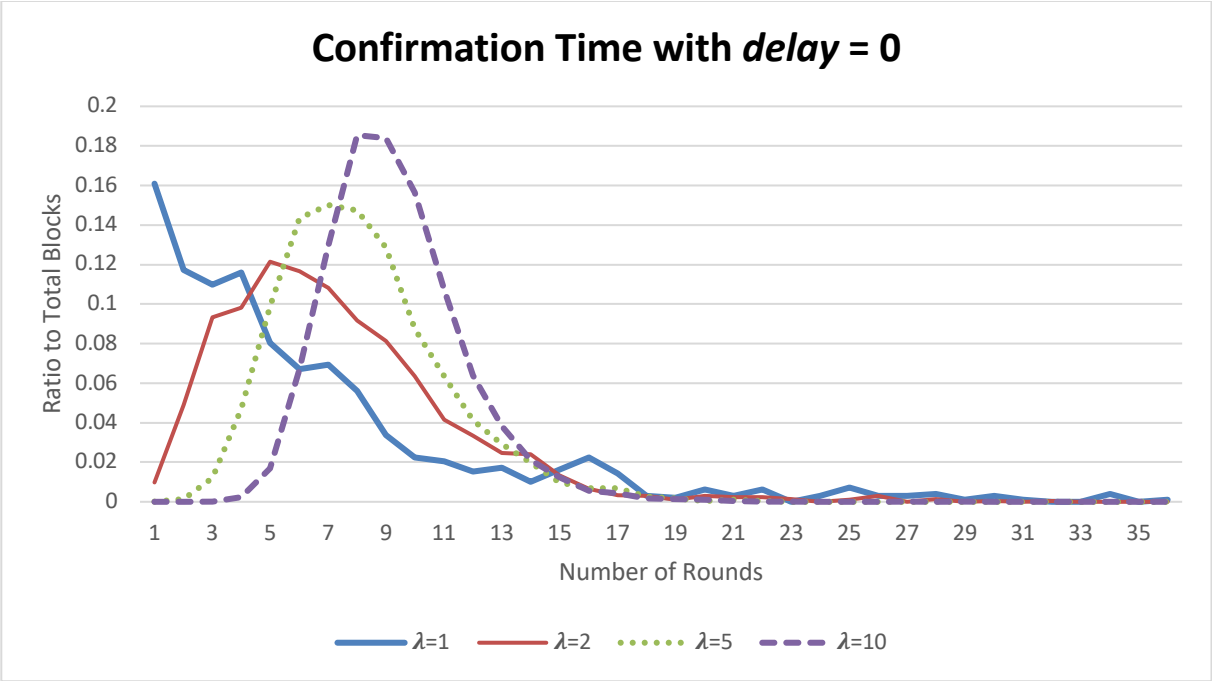


Figure 4-5. Confirmation Time with delay = 0

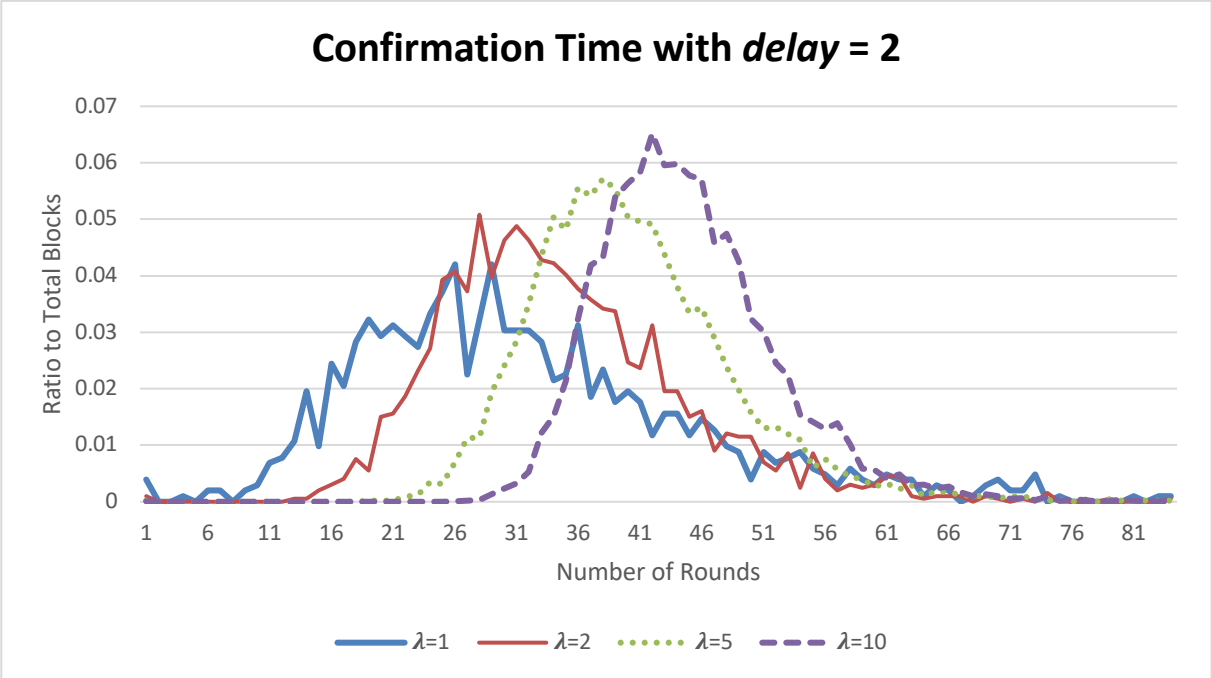


Figure 4-6. Confirmation Time with delay = 2

Figure 4-7 serves as illustration that a change in *delay* has a significant impact on the confirmation time. Thus, it is important for parties to adapt quickly to a new graph if the confirmation time should stay as little as possible.

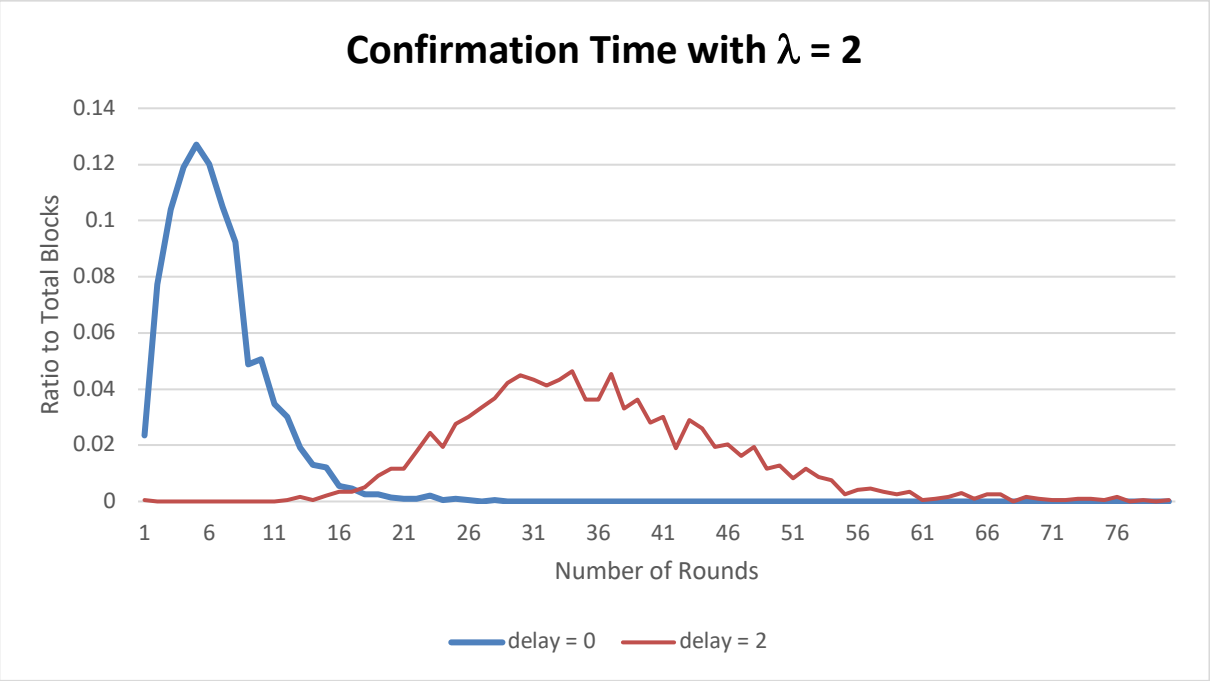


Figure 4-7. Confirmation Time with λ = 2

4.1.5 Confirmed vs. Unconfirmed Blocks

Table 4-4 describes the ratio of confirmed blocks compared to the total number of blocks after the simulation.

delay	0	0	0	0	2	2	2	2
λ	1	2	5	10	1	2	5	10
Ratio	99.5%	99.2%	99.3%	99.2%	97.2%	96.5%	96.1%	95.4%

Table 4-4. Ratio of Confirmed to Total Number of Blocks

When the simulation ends, there are no old unconfirmed blocks, only rather new blocks are unconfirmed, which correlates with the confirmation time in subsection 4.1.4. This also explains a higher value for simulations with *delay* equal to zero. So the only reason this number is not equal to 1 is that new mined blocks usually take some time to become confirmed. In general, higher ratios are deemed more beneficial, as a lot of unconfirmed blocks are not preferable. The reasons for this are listen in subsection 5.1.1.

In section 4.2, there will be a bigger difference analysing the ratios. Tangle networks with conflicting blocks are being discussed next.

4.2 Two Conflicting Blocks

In this section's scenario, two conflicting blocks are mined and broadcasted to the network at more or less the same time. This section is concerned with what happens to the tangle network as soon as conflicting blocks are being introduced.

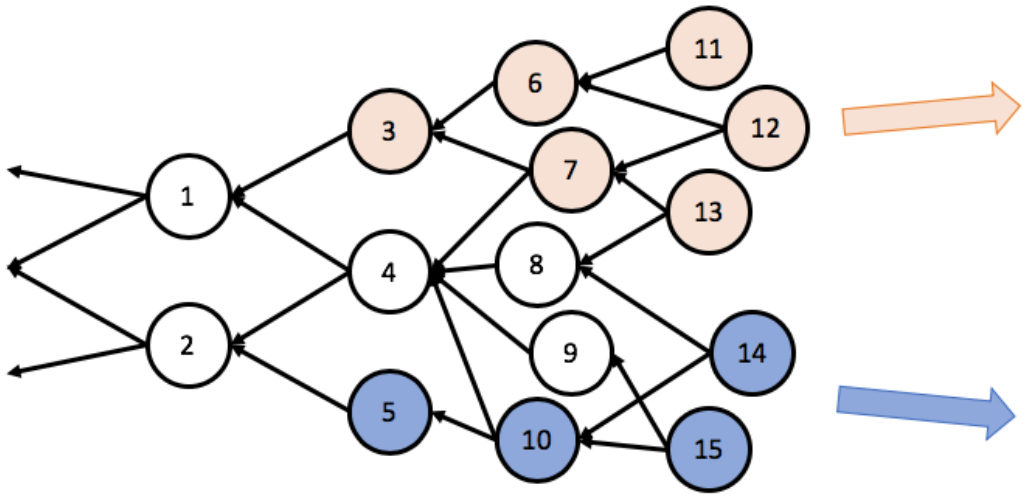


Figure 4-8. Tangle Network with a Fork and Two Branches

In figure 4-8, block 3 and block 5 are conflicting and two branches evolve. This pattern represents what happens in all the simulations within this section.

Taking a look at figure 4-8, it becomes clear as to why only one of two conflicting blocks can be confirmed. Here a simple proof:

The confirmation confidence of all the tips is equal to one:

$$T_{11} + T_{12} + T_{13} + T_{14} + T_{15} = 1$$

The confirmation confidence of block 3 is the sum of its connected tips:

$$B_3 = T_{11} + T_{12} + T_{13}, 0 \leq B_3 \leq 1$$

Block 5 cannot be approved by the tips approving block 3. Therefore, B_5 must be smaller than or equal to $1 - B_3$:

$$B_5 = T_{14} + T_{15} = 1 - B_3, 0 \leq B_5 \leq 1$$

$$Min\big(Max(B_3), Max(B_5)\big) = Min(Max(B_3), Max(1 - B_3)) \leq 0.5 < 0.95$$

■

The following subsections present the results of simulations with two conflicting blocks. Each subsection is devoted to a specific topic. The first origin gets inserted in round 200, the second one in the round stated in the table.

4.2.1 Tips

There is no significant difference comparing the number of tips in the networks with and without conflicting blocks.

Round	200	202	205	No conflict
Min	3	3	3	2
Max	15	15	14	14
Avg	8.5	8.3	8.4	7.7

Table 4-5. Number of Tips in a Conflicting Network

4.2.2 Ratio Confirmed vs Unconfirmed

This subsection shows a comparison of the number of confirmed blocks to the total number of blocks. This is to analyse the number of unconfirmed blocks.

Round	200	202	205	No conflict
Total	5209	4903	5141	4901
Confirmed	1023	989	1038	4867
Ratio	19.6%	20.2%	20.2%	99.3%

Table 4-6. Ratio of Confirmed vs Unconfirmed in a Conflicting Network

The simulation with no conflicting block shows a confirmation ratio of almost one, meaning there are only a few unconfirmed blocks. The simulations with conflicting blocks only show a confirmation ratio of about one fifth. This is because after the insertion of the two conflicting blocks no block becomes confirmed anymore.

This fact is really important, as it shows that the network is not efficient, and seems rather useless with conflicting blocks. The remainder of this section contains justifications and a closer analysis why there are no confirmed blocks after the insertion of conflicting blocks.

4.2.3 Origins Ratio and Number of Blocks in Branches

In this subsection, the origins of the conflicting blocks are analysed. In figure 4-8, this would be the blocks 3 and 5. They have the highest confirmation confidence of all the blocks within the same branch. This is because, every block belonging to a branch, automatically approves the origin of this branch.

The results are heavily dependent on the choice of the parents of the origin. To have a good picture, more than one simulation is necessary. Every row thereby represents a new simulation.

#B1 and #B2 display the number of blocks in their respective branch. CC1 and CC2 display the confirmation confidence of the origin of these branches. The sum of CC1 and CC2 is equal to one.

The two ratios in table 4-7 are calculated as follows:

$$Ratio_n = \frac{\#B1}{\#B2}\,,\; {Ratio_c}^2 = \left(\frac{CC1}{CC2}\right)^2$$

Round	#B1	#B2	$Ratio_n$	$Ratio_c^2$	CC1	CC2
200	2181	1896	1.150	1.092	.511	.489
	3686	294	12.5	11.4	.772	.228
	1351	2703	.500	.520	.419	.581
	17	4069	.0042	.0031	.053	.947
	3343	650	5.14	5.39	.699	.301
	3956	1	3956	4312.1	.985	.015
202	3768	132	28.5	24.9	.833	.167
	657	3335	.197	.198	.308	.692
	3588	370	9.70	9.81	.758	.242
205	3398	698	4.86	4.64	.683	.317
	3966	47	84.4	72.7	.895	.105

Table 4-7. Confirmation Confidence and Number of Blocks in Branches

Clearly visible, a confirmation confidence of more than 95% of either one of the origins is rare. This means none of the blocks, mined after the insertion of the origin, become confirmed.

The following fact stands out:

$$Ratio_n \approx {Ratio_c}^2$$

Now, there will be a short reasoning why this is the case.

Shown in the beginning of section 4.2, the confirmation confidence of the origin is equal to the probability of choosing a tip, with the TSA, approving the origin:

$$P_1 = CC1,\ P_2 = CC2$$

Choosing a tip from each branch makes it necessary to repeat the process. Therefore, the following cases do not have any influence on the probability of choosing a tip:

$$P_{1\wedge 2\ or\ 2\wedge 1} = 2 * CC1 * CC2$$

$P_{1\wedge 2\ or\ 2\wedge 1}$ stands for the probability of choosing two tips from different branches, which is not allowed by the protocol.

The probability of choosing two tips in branch one or two are:

$$P_{1\wedge 1} = CC1^2,\ P_{2\wedge 2} = CC2^2$$

Hence:

$$\left(\frac{CC1}{CC2}\right)^2 \approx \frac{\#B1}{\#B2}$$

The sum of the confirmation confidence of the two origins is equal to one:

$$CC1 + CC2 = 1$$

This is because the two confirmation confidences are “fighting” against each other, and usually none of them lies above 95%. However, there are some very rare cases where one branch is successful and the confirmation confidence is above 95%. The outcome of a branch depends on the origin and its initial confirmation confidence, as this barely changes after a few rounds (see subsection 4.1.3). The change of a block’s

confirmation confidence can be observed in figure 4-9. Each line represents the confirmation confidence of a conflicting origin in a different simulation.

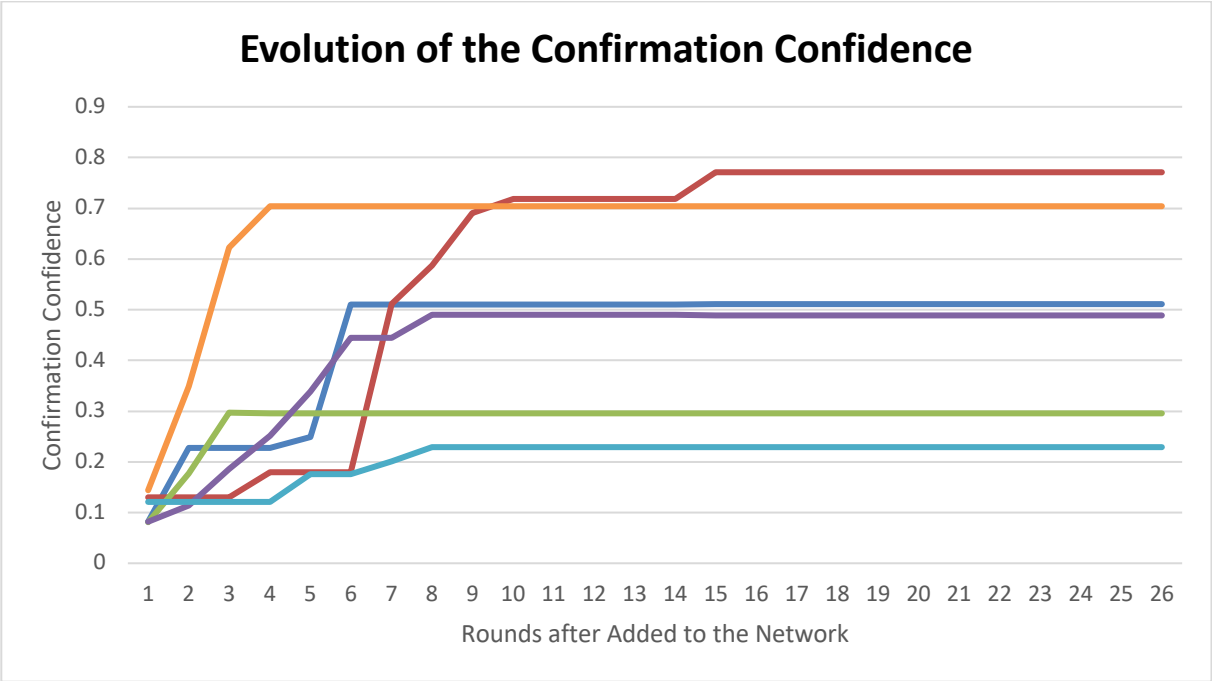


Figure 4-9. Development of the Confirmation Confidence of Conflicting Origins

It is clearly visible that it only takes around ten rounds until the confirmation confidence is settled. After that, only insignificant changes are visible. No matter how many blocks are mined on either branch, the confirmation confidence only changes insignificantly.

4.3 More conflicting blocks

As shown in the previous section, two conflicting blocks already have a large impact on the tangle network. This section will illustrate what happens if more conflicting blocks are added to the network.

In this simulation, several conflicting blocks are inserted in rounds 100, 175, 250 and 325. The simulation otherwise uses the same parameters as previously. The effects remain the same, yet it is nonetheless interesting to observe the output.

Figure 4-10 illustrates what such a simulation looks like. The numbers look different in each simulation. Figure 4-10 shows the output of one single simulation.

Each box stands for a cluster of blocks belonging to the same group. The number in a box depicts the number of blocks a box represents. The number above the box yields information concerning the confirmation confidence of the origin of the respective box.

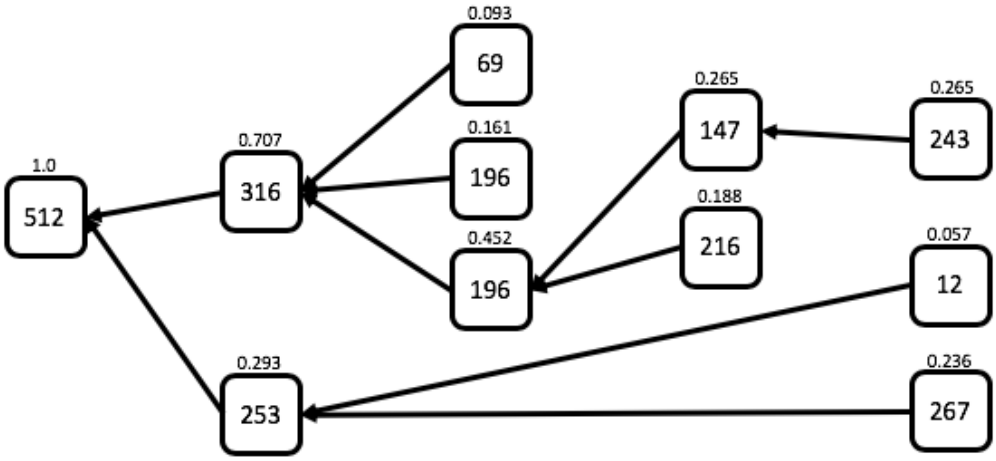


Figure 4-10. Network with More Conflicting Blocks Represented in Clusters

The following observations remain identical for each simulation:

The confirmation confidence decreases with each new conflicting group.

After the insertion of the first conflicting group, every block belongs to at least one conflicting group. Hence there are no nonconflicting blocks mined after the insertion.

4.4 Other Results

This section is devoted to various topics concerning the structure of the tangle. Each subsection inspects a different aspect of the tangle.

4.4.1 Alpha

The parameter α is used in the random walk and thereby affects the TSA and the confirmation confidence. Taking a deeper look and comparing weighted ($\alpha > 0$) and unweighted ($\alpha = 0$) random walks, it becomes clear that α does not have a big impact on the results.

Table 4-8 shows the probability of choosing either B_2 or B_3 with α and the weights as variables. Figure 4-11 illustrates what the starting position looks like.

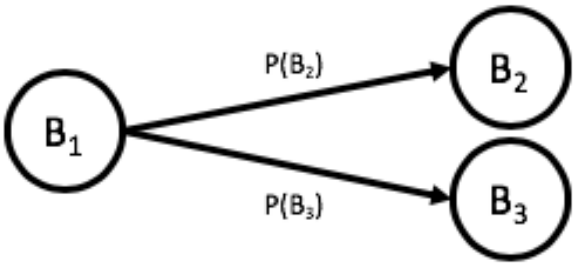


Figure 4-11. Path of Random Walk

Weights $[B_1/B_2]$	1 / 1	1 / 10	1 / 100	1 / 1000
$\alpha = 0$	.5	.5	.5	.5
$\alpha = 0.001$	.5	.502	.525	.731
$\alpha = 0.01$	.5	.522	.729	1
$\alpha = 0.1$	.5	.711	1	1
$\alpha = 1$	.5	.999	1	1
$\alpha = 10$	.5	1	1	1

Table 4-8. Alpha Comparison in a Random Walk

The is only a visible difference of $\alpha = 0$ and $\alpha = 0.001$, if the weight difference is really Subsection 4.1.3 shows that a weight of over 100 rarely and only after a long time happens. So taking the results from subsection 4.1.3 into account, it is clear that α does not have a significant impact.

Figure 4-12 shows an example of a simulation after 1000 rounds with λ equal to five and *delay* equal to zero. The following subsections shows a comparison of the un-weighted random walk ($\alpha = 0$) and the weighted random walk ($\alpha = 0.001$).

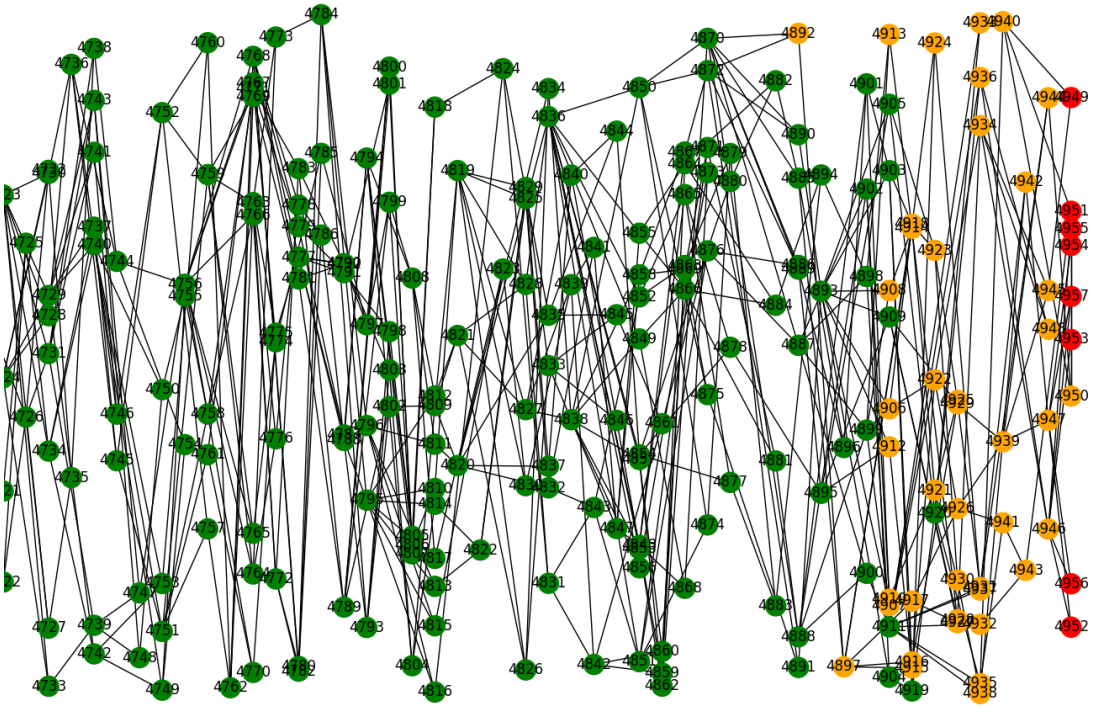


Figure 4-12. Example of a Tangle used for the Analysis in Subsection 4.4.1

4.4.1.1 TSA

There are eight tips at the end of the example simulation, shown in figure 4-12. Each bar in figure 4-13 represents the probability of choosing one of those tips with the random walk either set at $\alpha = 0$ (unweighted) or $\alpha = 0.001$ (weighted).

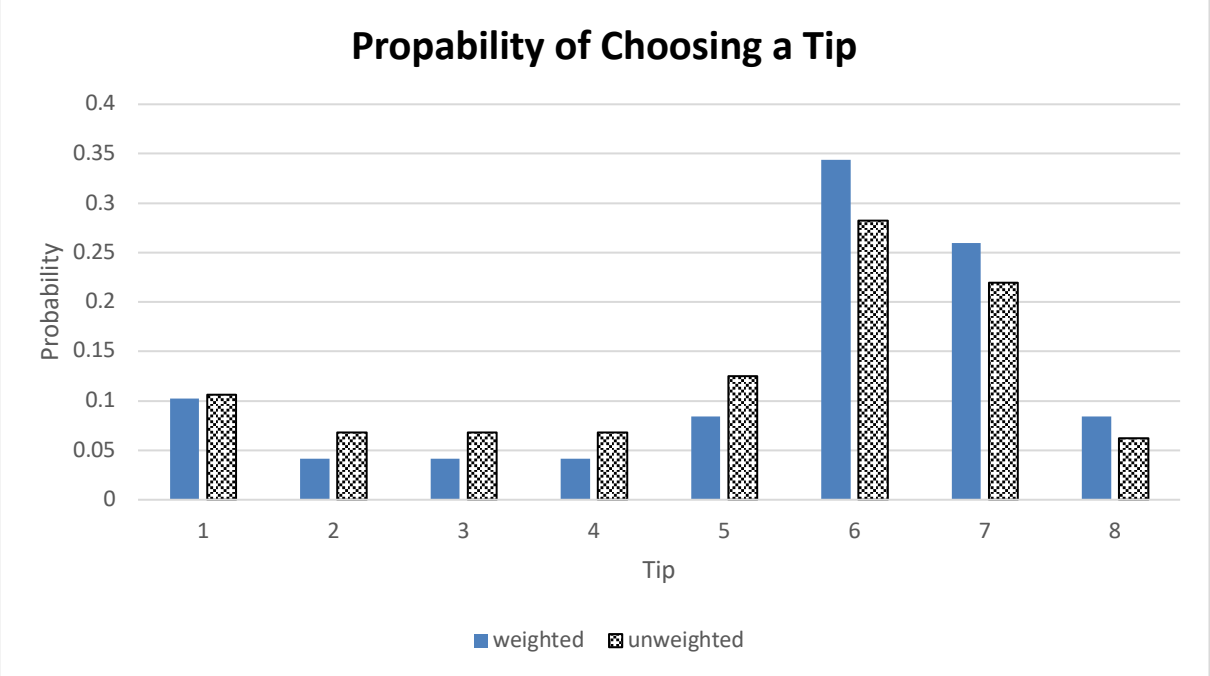


Figure 4-13. TSA Probability of Weighted and Unweighted Random Walk

A weighted random walk thereby favours the tips which have a high probability of getting chosen anyway. But the difference between unweighted and weighted TSA is small. The biggest difference is about six percent. In both cases the aggregate probabilities equal one.

4.4.1.2 Confirmation Confidence

Figure 4-14 shows the confirmation confidence of the last 16 blocks mined in the simulation in figure 4-12. If the confirmation confidence is equal to one, α does not matter. As most of the blocks have a confirmation confidence equal to one, which is shown in subsection 4.1.5, only the last mined blocks are used for the comparison.

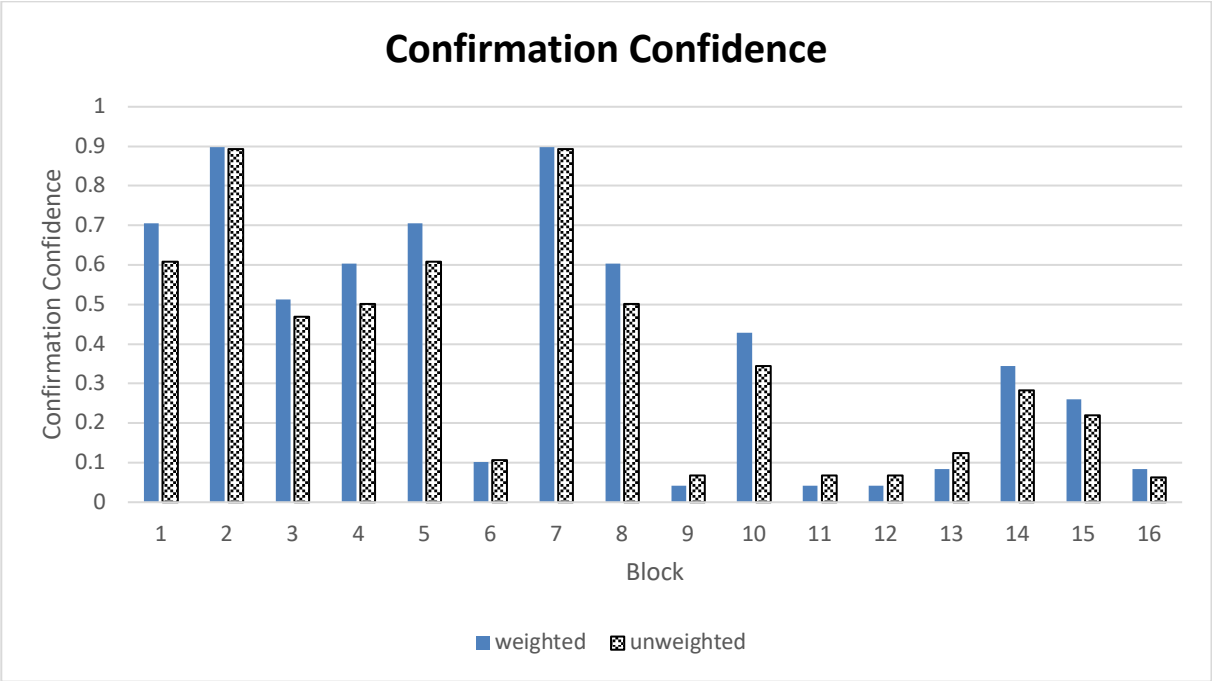


Figure 4-14. Confirmation Confidence of Weighted vs Unweighted Random Walk

The difference is again small. Taking the results from subsection 4.4.2 into account, difference of a weighted and an unweighted random walk is insignificant in terms of the confirmation confidence.

4.4.2 Confirmation Confidence: Exact Value vs 100 Random Walks

Using the given algorithm of 100 random walks to approximate the confirmation confidence of a block is not as accurate as using an algorithm which calculates the exact value. The difference is rather small, but it can still be misleading. Figure 4-15 displays the difference between those two algorithms. If a block has an exact confirmation confidence of one, the results will be identical if 100 random walks are conducted. Therefore, figure 4-15 only shows the last 37 blocks of the simulation, as all the previous blocks have a confirmation confidence of one, regardless of the algorithm.

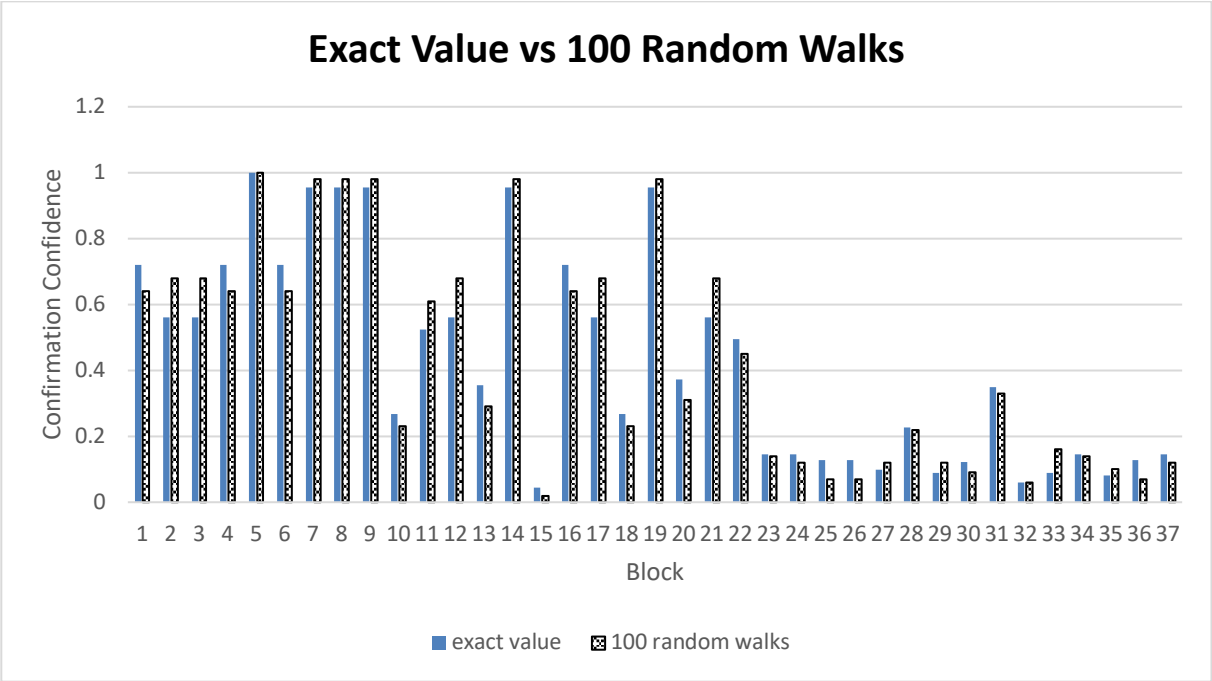


Figure 4-15. Exact Value vs 100 Random Walks

If a block is at exactly 95% confirmation confidence, it can have a big spectrum of approximation values using the 100 random walks.

It is equal to the binomial distribution with $n = 100$ and $p = 0.05$:

$$P_k = \binom{n}{k} p^k (1-p)^{n-k}$$

Figure 4-16 shows that it is almost equally likely for a block with exactly 95% confirmation confidence to be confirmed or not to be confirmed with 100 random walks.

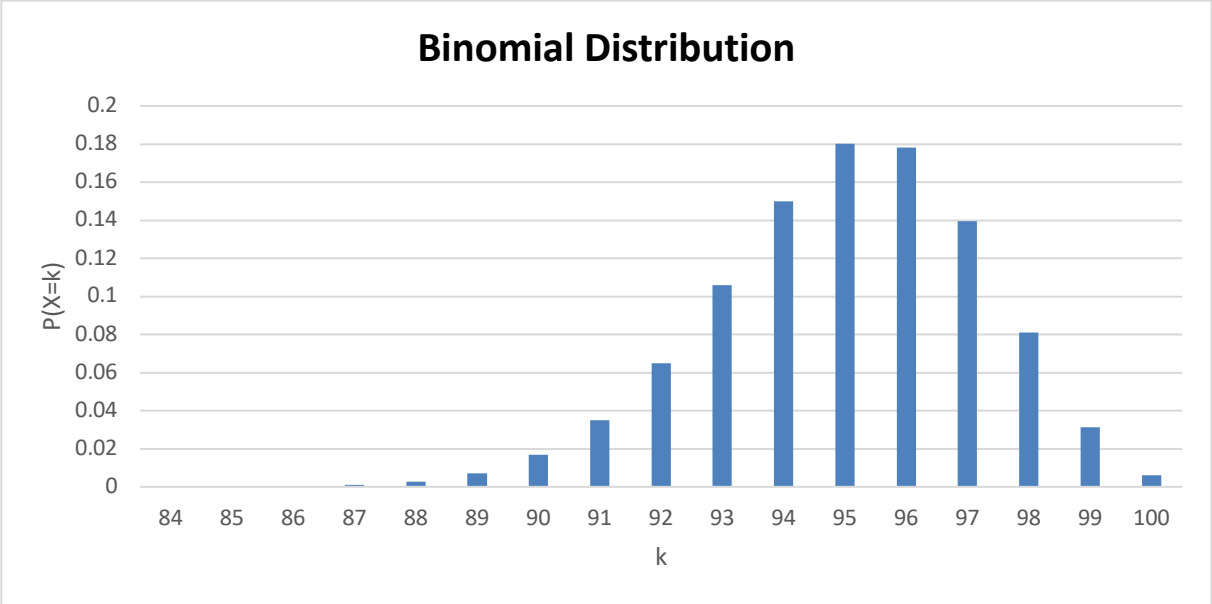


Figure 4-16. Binomial Distribution of the Confirmation Confidence

$P(X = k)$ shows the probability distribution of a block with an exact confirmation confidence of 95%.

4.4.3 Network of Mining Parties

Simulations, with parties being slow at mining do not change anything significant in the results. Indeed, it leads to similar outcomes as in previous simulations with a higher *delay*. This implies that parties with less computing power can participate in the network, but however, they slow down the confirmation process for all blocks. Also, blocks published later to the network have almost the same possibility of becoming approved than those added instantly.

4.4.4 Blocks with Only One Parent

A block with only one parent entails some disadvantages. For example, it is less likely to be approved, as less paths lead to that block. Figure 4-17 serves as a hypothetical scenario to illustrate this disadvantage. If T_1 has one parent (only one of the dashed lines exists), its possibility of becoming chosen is only about half as likely (depending on the weights in the graph) as it would be with two parents.

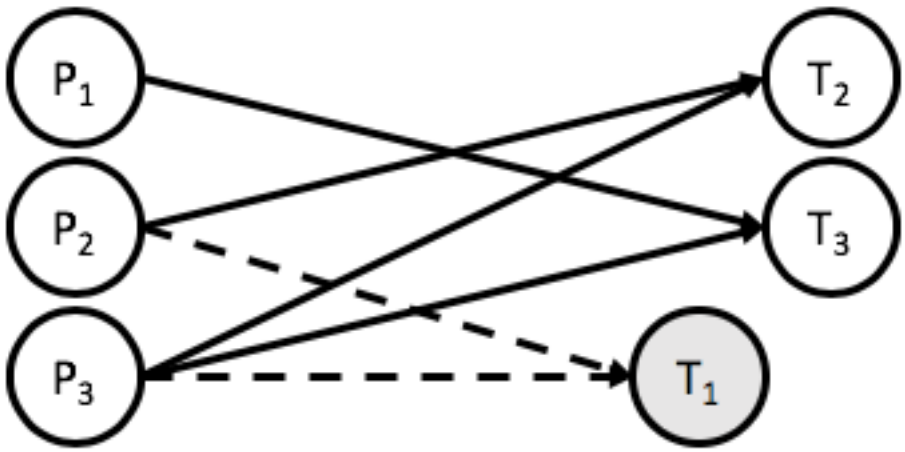


Figure 4-17. Choosing Two Parents

For the party mining block T_1, it would make more sense to choose two parents instead of the given TSA, even if this would imply to choosing an already approved block as its second parent rather than a tip.

Figure 4-18 shows an impossible scenario with a TSA. Block 3 uses block 1 and block 2 as parents, which means block 1 is not a tip: it is already approved by block 2. But

those scenarios would be hard to detect as it gets much more complicated with more blocks.

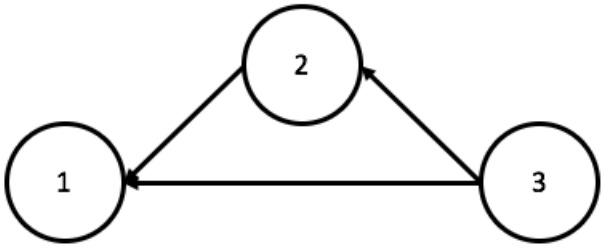


Figure 4-18. Impossible Scenario with a TSA

Even though this scenario is not made with a TSA, it is more likely that block 3 gets approved later in the process. Imagining new blocks approving block 1 and 2, the probability of choosing block 3 as a parent decreases. With only one parent, the probability is reduced even more.

5 Conclusion

This chapter is devoted to analyse the results from the previous chapters.

5.1 Networks with no Conflicting Blocks

5.1.1 Unconfirmed Blocks

Old unconfirmed blocks are bad for the network. They add a factor of uncertainty, as they could become confirmed at any moment. If only one new block approves an old unconfirmed block, the probability of the old block getting confirmed rises significantly. This scenario is shown in figure 5-1, with example numbers for illustration in table 5-1. During round *x – 1* block A is still unconfirmed with a low confirmation confidence. A gets approved in round *x* by B. B also approves D and has consequently a higher confirmation confidence, which hugely elevates the confirmation confidence of A.

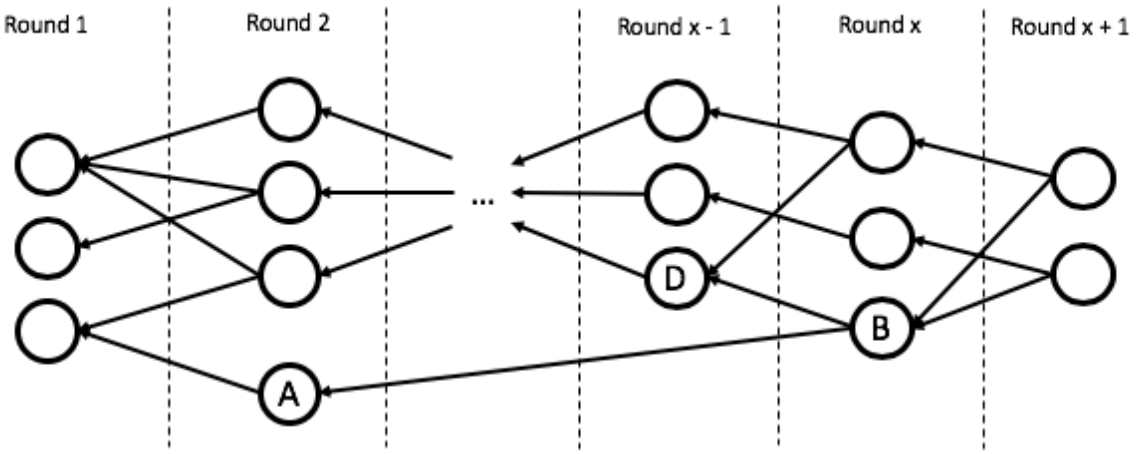


Figure 5-1. Approving an Old Unconfirmed Block

Time [round]	2	x – 1	x	x + 1
CC(A)	.01	.01	.33	1
CC(D)	-	.32	.67	1

Table 5-1. Example Confirmation Confidence

This example shows it is still easily possible for an old block to get confirmed quickly, even with a very low confirmation confidence at first.

In networks with no conflicting blocks, there are usually no old unconfirmed blocks. The maximum row in table 4-3 shows an upper boundary for the confirmation time

which is mainly influenced by the *delay*. The time of a block being unconfirmed in the network is restricted by this upper boundary.

Table 4-4 suggests the network, with no conflicting blocks, is secure. Almost all of the blocks are confirmed and the computing power needed to lower the confirmation confidence is immense.

5.1.2 Confirmation Time

It is preferable to have a confirmation time as low as possible, so parties can ensure a quick handling of the transactions.

While λ only has a small impact on the results, the *delay* is more important for the outcome of the results. Figure 4-7 indicates a big difference comparing different parameter inputs for *delay*.

This means mining parties need to adapt quickly to new mined blocks, in order for the network to be as efficient as possible.

Otherwise, the confirmation time of the network rises significantly. It is maybe necessary to change the parents and start the mining process again. But this depends on the probability of satisfying the hash function.

5.1.3 Constant Weights

Subsection 4.1.3 illustrates the development of weights for regular blocks. It is easily observable that weights of blocks stay constant after a given amount of time and figure 4-3 also suggests weights are bounded from above.

Constant weights ensure stability. The probability of choosing a particular tip does not change significantly if it is not chosen instantly. It only decreases by a small margin. This means all tips should be approved in a reasonable amount of time.

5.2 Networks with Conflicting Blocks

In section 4.2, it is shown that conflicting blocks have a large impact on the network. In almost every case, there are no confirmed blocks after only two conflicting blocks. The setup does not matter as there are going to be problems irrespective of how the TSA or the parameters are set up. This would make the system useless in general, as depicted by the number of unconfirmed blocks in table 4-6.

The reasoning shown in subsection 4.2.3 illustrates that there will be no solution to this problem as soon as conflicting blocks are added to the network. Therefore, the system needs a mechanism which prevents the mining of conflicting blocks in the first place. This mechanism appears missing today, however.

5.3 Further Observations

5.3.1 Slow Blocks

A slow block is one that approves a block, which was approved already a significant amount of time ago. Figure 5-2 illustrates an example, as the slow block B approves A which is already approved indicated by the bold arrow.

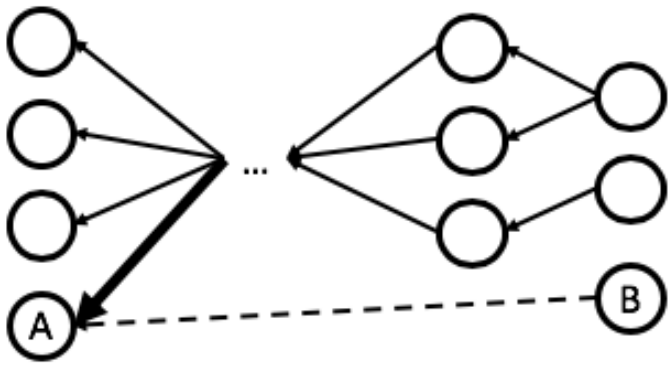


Figure 5-2. Example of a Slow Block

The occurrence of a slow block indicates one of the following scenarios:

- B was mined slowly.
- B was not published instantly.
- B was not mined with a real TSA, meaning the party purposely chose a block, which was not a tip, as the parent for block B.

None of these scenarios are desirable. Parties which add slow blocks to the network, slow down the confirmation process significantly. Hence, there should be rules which prevent the adding of slow blocks.

Recognizing slow blocks can be hard. The only indicator is that B approves A, which was already approved a long time ago. This information however, is not in A itself but in the block approving A.

A punishment for adding slow blocks or a mechanism which prevents the addition in general would guarantee a lower confirmation time and thereby better efficiency.

5.3.2 Removing Alpha

The factor α does not have a significant impact on the network, as shown in subsection 4.4.1. However, it makes the process of mining and the analysis of the network for mining parties much more complicated. Removing α and making the weights for all the blocks equal would not change the long-time output of the network and save a lot of resources for mining parties.

6 Bibliography

[1] M. Nofer, P. Gomber, O. Hinz and D. Schiereck, “Blockchain,” *Business & Information Systems Engineering,* vol. 59, pp. 183-187, 2017.

[2] S. Popov, “The Tangle,” 2018. [Online]. Available: http://www.descryptions.com/Iota.pdf. [Accessed 29 December 2020].

[3] V. Attias and Q. Bramas, “How to Choose Its Parents in the Tangle,” *Networked System,* vol. 11704, pp. 275-280, 2019.

[4] A. Gal, “The Tangle: an Illustrated Introduction,” 2018. [Online]. Available: https://blog.iota.org/the-tangle-an-illustrated-introduction-4d5eae6fe8d4/. [Accessed 10 January 2021].

[5] Q. Bramas, “The Stability and the Security of the Tangle,” 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01716111v2. [Accessed 31 December 2020].

7 List of Figures and Tables

Erklärung

Erklärung gemäss Art. 30 RSL Phil.-nat. 18

Ich erkläre hiermit, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäss aus Quellen entnommen wurden, habe ich als solche gekennzeichnet. Mir ist bekannt, dass andernfalls der Senat gemäss Artikel 36 Absatz 1 Buchstabe r des Gesetzes vom 5. September 1996 über die Universität zum Entzug des auf Grund dieser Arbeit verliehenen Titels berechtigt ist.

Für die Zwecke der Begutachtung und der Überprüfung der Einhaltung der Selbständigkeitserklärung bzw. der Reglemente betreffend Plagiate erteile ich der Universität Bern das Recht, die dazu erforderlichen Personendaten zu bearbeiten und Nutzungshandlungen vorzunehmen, insbesondere die schriftliche Arbeit zu vervielfältigen und dauerhaft in einer Datenbank zu speichern sowie diese zur Überprüfung von Arbeiten Dritter zu verwenden oder hierzu zur Verfügung zu stellen.

Bern, 13.03.21

Ort/Datum

Unterschrift