University of Zurich[UZH]

MASTER BASIS MODULE – Communication Systems Group, Prof. Dr. Burkhard Stiller

Analysis of the Tangle in the IoT Domain

Simon Bachmann
Zurich, Switzerland
Student ID: 14-709-893

Supervisor: Sina Rafati
Date of Submission: 20. Mai 2019

University of Zurich
Department of Informatics (IFI)
Binzmuhlestrasse 14, CH-8050 Zurich, Switzerland

ifi

Bachelor Thesis
Communication Systems Group (CSG)
Department of Informatics (IFI)
University of Zurich
Binzmuhlestrasse 14, CH-8050 Zurich, Switzerland
URL: http://www.csg.uzh.ch/

Abstract

Das ist die Kurzfassung...

Acknowledgments

Optional

Contents

Chapter 1

Introduction

1.1 Motivation

1.2 Description of Work

1.3 Thesis Outline

Chapter 2

Related Work

Chapter 3

The Tangle

This chapter covers the fundamental building blocks of the directed acyclic graph (DAG) architecture of IOTA called Tangle [2].

3.1 Participation in the network

The tangle does not know the discrepancy of validating nodes (miners) and issuing nodes in a sense that Bitcoin does. Every node in the network can issue and validate transactions. A new transaction is what motivates nodes to validate and propagate transactions. Each node calculates statistics about the activity of its neighbors. If a neighbor appears to be lazy, it can be dropped. For this reason, a node in the network is incentivized to participate in the network, even at times when it does not issue transactions.

3.2 Transactions

As there are no miners in the network, transactions are being validated by other nodes that issue transactions themselves. In order to create a new transaction on the network, a node does the following steps:

1. The node chooses two unconfirmed transactions according to an Monte Carlo Walk algorithm which will be described in more detail in Section 3.3.

2. The node is responsible for checking the validity of these two transactions. Conflicting transactions are ignored.

3. A node has to perform a cryptographic puzzle in order to make the new transaction valid. Similar to the Proof of Work (PoW) mechanism in Bitcoin, this puzzle is solved with computational resources. The puzzle is defined by finding a nonce such that the hash of this number concatenated with some data from the approved

transactions results in a number smaller than some predefined constant. This puzzle is necessary in order to prevent several attack scenarios which will be discussed in Section 3.5.

The next section describes the basic concepts of the tangle. For all figures, a box resembles a transaction and the directed edge between nodes illustrates the approval of a transactions. In order to understand the approval algorithm, the following five parameters are defined for every transaction.

weight The weight of a transaction is defined by the amount of work that the issuing node has invested in to this transaction. The weight of a transaction resembles its importance. This measurement helps to prevent spamming and other attacks since no node can create an abundance of transactions with meaningful weights within a short period of time.

cumulative weight The cumulative weight of a transaction is calculated by the weight of the transaction itself plus the sum of all transactions that directly or indirectly approve this transaction. Figure 3.1 shows how the weight and the cumulative weight change after the new transaction X is added, the smaller number denotes the weight of a node and the bold number represents cumulative weight of a transaction.

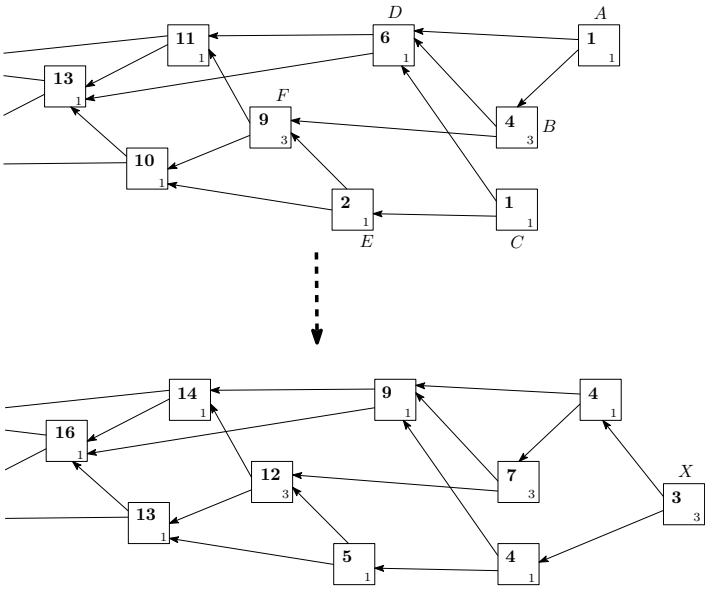

Figure 3.1: Cumulative Weight [2]

height The height of a transaction is the length of the longest oriented path to the genesis transaction. In Figure 3.2, transaction G has a height of 1 due to the blue edge.

depth The depth of a transaction is the length of the longest reverse oriented path to some tip. In Figure 3.2, transaction G has a depth of 4 due to the red approvals from newer transactions *F, D, B* and *A*.

score The score of a transaction is the sum of weights of all transactions that are approved by this transaction plus its own weight. The scores for transactions A and C are shown with the circled number in Figure 3.2.

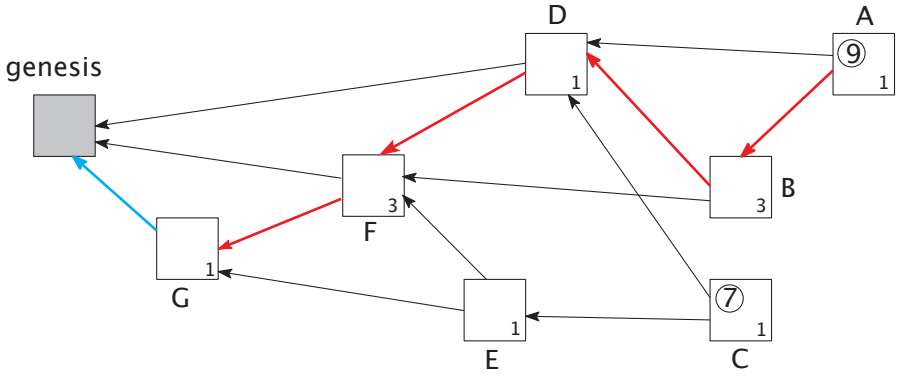

Figure 3.2: Height, Depth and Score [2]

These parameters will play an important role when discussing tip selection in Section 3.3 and attack scenarios 3.5.

3.3 Tip Selection

Unapproved transactions are called tips. This section covers the reasons why the process of selecting a tip is important for the network. In order to discuss the tip selection algorithm, a simulation of the DAG architecture is characterized by the following two parameters.

Transaction rate γ Transactions do not arrive evenly throughout time. To model such behaviour, a mathematical object called Poisson point process is used. The arrival rate of transactions is specified by the parameter γ. The higher γ is set in the simulation, the more transactions arrive within one time-unit. If γ is set to a really small number, the graph grows in the form of a linked list as there is always just one tip to be approved by a new transaction. Figure 3.3 shows a simulation, where $\gamma = 5$. The tips are drawn as grey boxes.

Delay h As new transactions must perform computational work for spam prevention and this computation is based on the selected tip of that transactions, there is a delay between choosing the tips and publishing the transaction. This delay is defined by the parameter h. For a device with less computational power, h will be larger than for computers with higher processing power. The simulation in Figure 3.3 assumes that $d = 1$ for every device that adds a new transaction. Thus, for example the node issuing transaction 5 did not know about the transactions 1-4 when he started with the PoW computation for transaction 5. The time difference to transactions 1-4 is less than d.

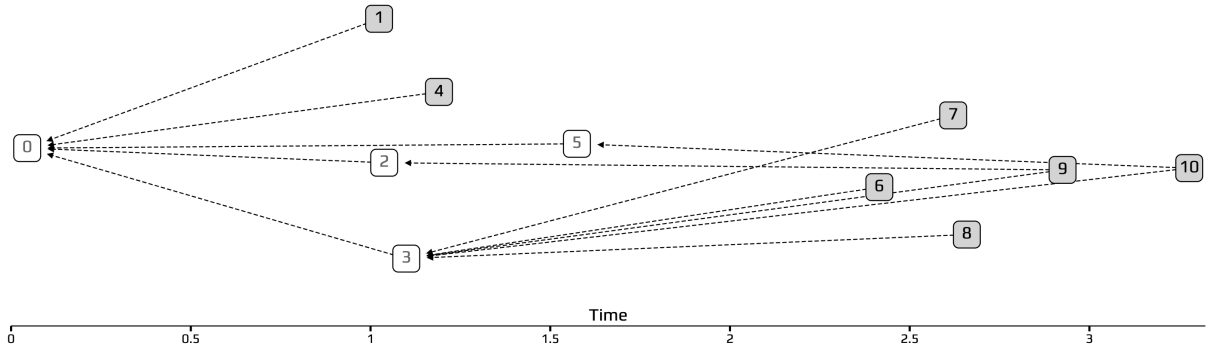

Figure 3.3: Simulation with $\gamma = 5$ and $d = 1$ [5]

There is a possibility that one or both of the validated transactions might no longer be a tip at the time of broadcasting the new transaction. Thus, the tangle must also accept transactions that approve already verified transactions. However, the tip selection algorithm must avoid these *lazy tips* which point to older transactions. Confirming old transaction is unwanted since it increases the branching factor of the graph and thus, it increases the number of tips. Furthermore, *lazy* nodes do not help the network to grow since no unapproved transactions are confirmed. In Figure 3.4, transaction 14 is added to the network by a lazy node.

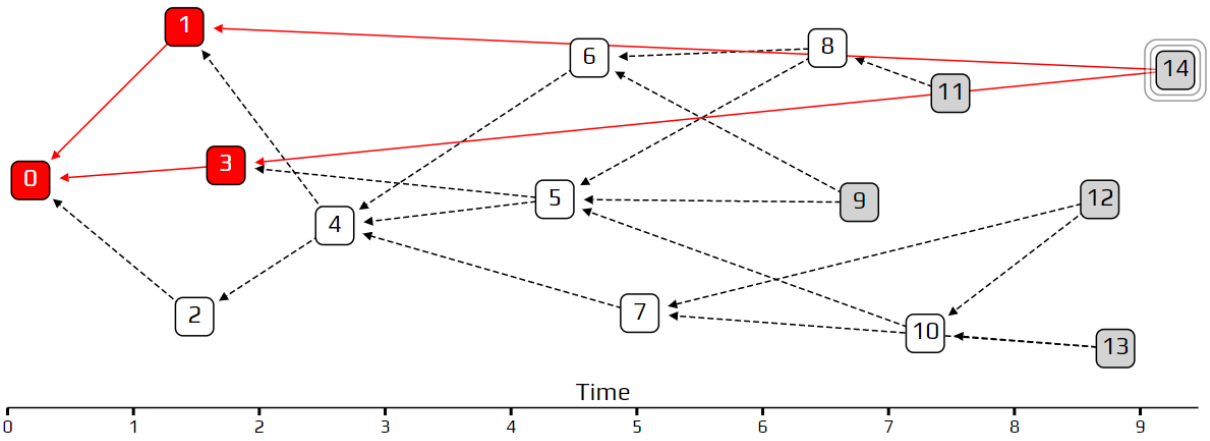

Figure 3.4: Lazy Tip [5]

For finding a tip, a node must walk from the genesis until it reaches an unconfirmed transaction. The node has to choose between multiple possible paths. If the node chooses the path solely based on the branching factor, transactions added by lazy nodes would be selected with the same probability as any other transaction. Thus, the random walk algorithm must be less biased towards lazy transactions. This is achieved by favoring transactions that have a higher cumulative weight. The bias factor is defined with the parameter α. Setting α to a high value results in many unconfirmed transactions since

most approvers choose the same path to the tip. These unconfirmed transactions are left behind and will never be accepted. Thus, determining an ideal value for α is crucial for the usability of the network and depends on the transaction arrival rate, the PoW delay of different devices in the network, network delay and the number of tips.

The method of setting a rule on how to find a path towards a tip is called a Markov Chain Monte Carlo technique (MCMC) [8]. In a Markov chain, each step enforces a rule which is defined in advanced and does not depend on the previous step. In the example of the Tangle, each step is a node in the graph and the rules are the probabilities of the available paths depending on the cumulative weights.

3.4 Transaction Validation

The transaction validation process is similar to Bitcoin's unspent transaction output (UTXO) model. An unspent transaction output is the output of a transaction that a user receives and is able to spend in the future. Thus, the validating node must check all previously made transactions of the sender address in order to verify a transaction. The smallest unit of the underlying currency is also called IOTA. All IOTA are minted in the genesis transaction and therefore, every IOTA can be traced back to genesis block.

Transactions cannot be seen as valid as soon as one approver has referenced it. Thus, a new parameter is introduced called confirmation confidence. The confirmation confidence for a transaction X can be calculated in the following way.

1. The tip selection algorithm is run 100 times.
2. The number of tips that approve transaction X is counted.
3. Every tip is weighted by the likelihood that it will be accepted in the future.
4. The confirmation confidence of transaction X is the fraction of approving transactions.

It is assumed that transactions are issued by a large number of independent entities, so the process of incoming transactions can be modeled Poisson point process [2]. λ denotes the rate of the Poisson process and it is assumed that it remains constant in time. At some point in time, every new transaction will approve transaction X since all tips include a path to transaction X. Thus, after the adoption period, the cumulative weight will grow linearly with $\lambda * w$ where w is the average weight of a transaction.

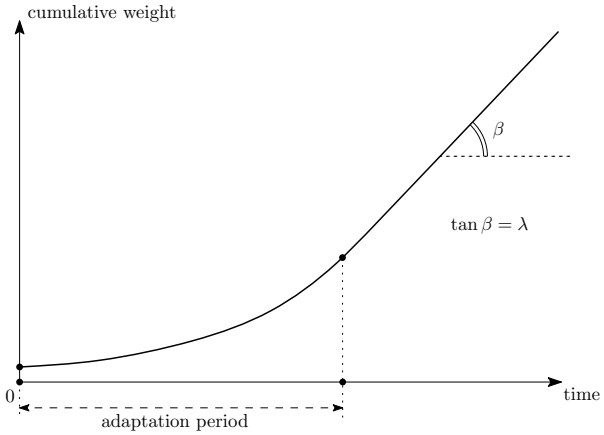

Figure 3.5: Confirmation Confidence and Cumulative Weight Growth [2]

In Figure 3.6, transactions with confirmation confidence of more than 0.95 have a thick border. Almost any new honest transaction that is added to this tangle will confirm these transactions (except lazy nodes with lazy tip selection). In the shown example, transaction 9 confirms all the red transactions and is confirmed by the blue transactions. There are four tips in the shown simulation - 6, 10, 11 and 12. The confirmation confidence of transaction 9 is 0.94 due to the fact that transactions 10, 11 and 12 have more importance than 6. Transaction 4 has confirmation confidence of 1 since all tips have a path to transaction 4 and therefore, there is no transaction in the network that does not confirm this transaction.

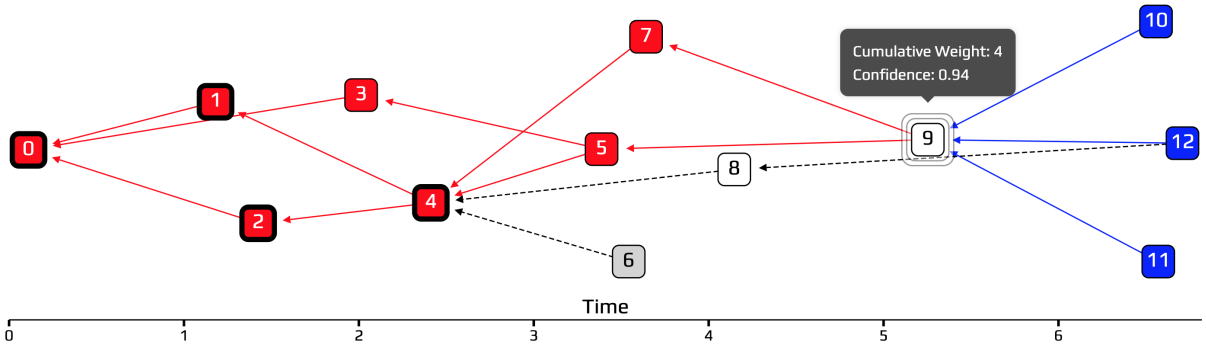

Figure 3.6: Graph Representation of the Confirmation Confidence

3.5 Attacks

This section covers some of the possible attack scenarios and how the Tangle can still maintain consensus among honest users.

3.5.1 Double Spend

A double spend situation occurs when a user tries to exceed his account balance by issuing two or more conflicting transactions. Figure 3.7 illustrates such a scenario. A box represents a transaction. The dashed box inside represents the current state in the graph but is not part of an actual transaction. In this simulation, Alice owns only 15 IOTA but issues two transactions with 10 IOTA each. Bob cannot approve both of Alice's transactions as they result in a negative account balance.

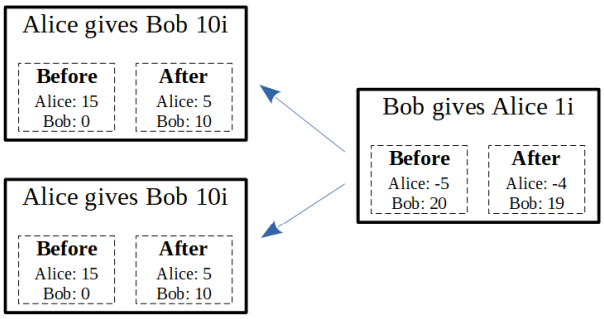

Figure 3.7: Double Spend Attack [7]

The solution to this problematic situation is the weighted random walk discussed in Section 3.3. One of the two transactions will become heavier and the lighter one will be abandoned. This implies that a confirmed transaction cannot be considered as valid as soon as it has been approved for the first time.

Confirmation confidence is introduced in Section 3.4 and provides a measurement of what percentage the network has directly or indirectly approved a transaction.

3.5.2 Large Weight Attack

The large weight attack has the same intent as the double spend but actively tries to invalidate a transaction with high confirmation confidence. This can be achieved by a malicious user as follows.

1. A transaction is created and broadcasted that is intended to revert.
2. The malicious user waits until the receiver believes the transaction has a high enough confirmation rate. The merchant ships the product/service.
3. The attacker uses its computational power and issues a double-spending transaction with a large weight followed by many more transactions. This transaction does not approve the first transaction and thus they compete with each other for finality.
4. The bad actor hopes that the dishonest subtangle gains more cumulative weight than the honest subtangle.

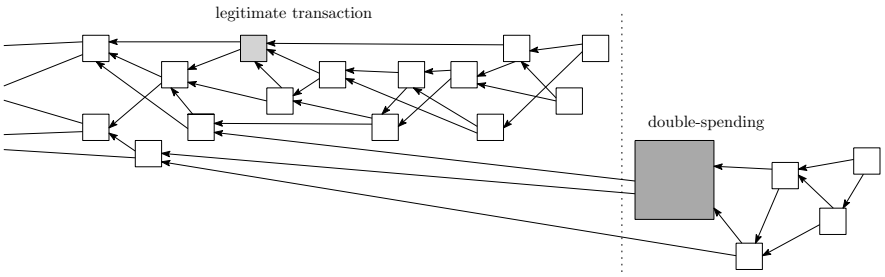

Figure 3.8: Large Weight Attack [2]

This attack can only be carried out if the attacker has more computing power than all the nodes that actively issue new transactions. In a well-established network with many nodes issuing transactions, this is less of an issue. In the early stages, however, there are not enough transactions passing through the network in order to be safe from such an attack. Due to this reason, the IOTA foundation has put a coordinator in place which is discussed in more detail in Section ?.

3.5.3 Parasite Chain Attack

The parasite chain attack also tries to convince the network to abandon a previously confirmed transaction by biasing the tip selection algorithm.The attack works as follows:

1. The attacker creates a transaction branching off from the main tangle (MT). He does not broadcast this transaction. This transaction is the red dot furthest to left in Figure 3.9.
2. Instead, he keeps adding new transactions to this local chain called parasite chain (PC).
3. He makes sure, that he references the MT within the PC.
4. The malicious user creates a transaction on the MT which he hopes to get abandoned by the network when he publishes the parasite chain. This transaction is the red dot furthest to right.
5. The user waits until the transaction on the MT is considered as validated. During this time he keeps building on the PC but can only reference transactions before the double-spend transaction on the MT.
6. At this point, the bad actor broadcasts the parasite chain.
7. Furthermore, he might try to artificially inflate the number of tips on the PC.

The attacker's intention is that new transactions reference the parasite chain such that the MT will be orphaned. However, the tips on the parasite chain have a smaller amount of cumulative weight, assumed the attacker has less computational power than the rest

of the network. Thus, in order to mitigate such an attack, it is important for the MCMC selection algorithm to be biased towards transactions with a high cumulative weight. The tradeoff for setting the bias is discussed in Section 3.3.

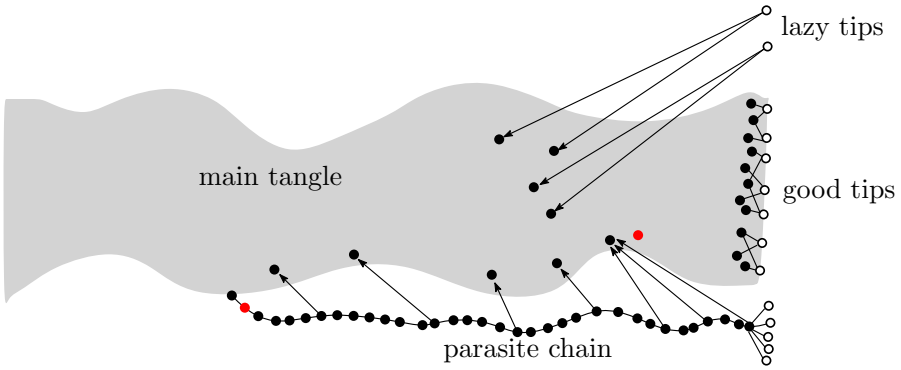

Figure 3.9: Parasite Chain Attack [2]

Chapter 4

Evaluation

Chapter 5

Summary and Conclusions

Bibliography

[1] Autoren: Titel, Verlag, http://..., Datum.

[2] Serguei Popov: The Tangle - Version 1.4.3, https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf, 30th of April 2018, accessed 20th of Mai 2019..

[3] Alon Gal: The Tangle: an illustrated introduction, https://blog.iota.org/the-tangle-an-illustrated-introduction-f359b8b2ec80, 31st of January 2018, accessed 20th of Mai 2019.

[4] Alon Gal: The Tangle: an illustrated introduction - Part 2: transaction rates, latency, and random walks, https://blog.iota.org/the-tangle-an-illustrated-introduction-f359b8b2ec80, 7th of February 2018, accessed 20th of Mai 2019.

[5] Alon Gal: The Tangle: an illustrated introduction - Part 3: Cumulative weights and weighted random walks, https://blog.iota.org/the-tangle-an-illustrated-introduction-f359b8b2ec80, 14th of February 2018, accessed 20th of Mai 2019.

[6] Alon Gal: The Tangle: an illustrated introduction - Part 4: Approvers, balances, and double-spends, https://blog.iota.org/the-tangle-an-illustrated-introduction-f359b8b2ec80, 21st of February 2018, accessed 20th of Mai 2019.

[7] Alon Gal: The Tangle: an illustrated introduction - Part 5: Consensus, confirmation confidence, and the coordinator, https://blog.iota.org/the-tangle-an-illustrated-introduction-f359b8b2ec80, 28th of February 2018, accessed 20th of Mai 2019.

[8] Ben Shaver: A Zero-Math Introduction to Markov Chain Monte Carlo Methods, https://towardsdatascience.com/a-zero-math-introduction-to-markov-chain-monte-carlo-methods-dcba889e0c50, 22nd of Dezember 2017, accessed 20th of Mai 2019.

[9] Will Koehrsen: The Poisson Distribution and Poisson Process Explained, https://towardsdatascience.com/the-poisson-distribution-and-poisson-process-explained-4e2cb17d459, 21st of January 2019, accessed 20th of Mai 2019.

[10] Bartosz Kusmierz: Attack analysis - the simple parasite chain `https://blog.iota.org/attack-analysis-the-simple-parasite-chain-42a34bfeaf23`, 10th of October 2019, accessed 20th of Mai 2019.

Abbreviations

MCMC	Markov Chain Monte Carlo
DAG	Directed Acyclic Graph
IF	IOTA Foundation
IoT	Internet of Things
UTXO	Unspent Transaction Output
MIOTA	Mega IOTA (Equivalent to 1 Million IOTA)

Glossary

IOTA IOTA is the name of the smallest unit in the crypto currency created by the IOTA Foundation. Most exchanges use MIOTA which is equivalent to 1 Million IOTA.

Lazy Tips A node in the network which does not confirm the most recent transactions creates lazy tips.

Tangle The underlying graph data sturcture in IOTA is called Tangle.

List of Figures

List of Tables

Appendix A

Installation Guidelines

Appendix B

Contents of the CD