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

University of Zurich^UZH

Analysis of the Tangle in the IoT Domain

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

Supervisor: Sina Rafati
Date of Submission: 23. Juni 2019

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

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...

Contents

Chapter 1

Introduction

1.1 Motivation

One of the biggest hurdles in distributed ledger technology (DLT) is the scalability issue. Bitcoin handles around 7 transactions per second (TPS), Ethereum 15 TPS [2], Litecoin 56 TPS and Ripple 1500 TPS [3]. Higher transaction throughput is achieved with larger block size. This is not a sustainable approach since the data that is stored in every node of the network grows linearly to the networks block size and effectively forcing people to leave the network with less storage and bandwidth capacity.

In order to compete with Visa's 1600 TPS [4], increasing the block size is not enough. The lightning network is one proposal to solve the scalability issue by creating off chain transactions. However, it is not flawless because the funds are locked in payment channels and the transactions to open and close such payment channels are still slow and expensive. Ethereum is working on a solution that works similar to database sharding, where every node is storing only a portion of all the transactions on the network. However, this requires additional mechanisms such that nodes must not trust other shards in order to verify transactions that are stored in other shards.

All the DLTs mentioned previously use a linked list as a core data structure. However, a DLT created by the IOTA foundation uses a directed acyclic graph (DAG) called Tangle. This fundamental difference brings several advantages compared to traditional DLTs. However, there are other hurdles to overcome using a DAG architecture. The arguments for and against a DAG architechture in a DLT is evaluated as part of this Master Basis Module (MBM).

1.2 Description of Work

This MBM will be conducted as a research assignment along with a 30 minutes presentation. The following topics are to be analyzed, evaluated and discussed.

- A detailed analysis of the strengths and weaknesses of the Tangle's architecture and how it reaches its consensus. This also includes the problem of the coordinator and the path to full decentralization. It includes a discussion about how suitable the IOTA network is for a future project with Internet-of-Things (IoT) devices.
- Analysis of possible attack scenarios including the double-spend attack, large weight attack and the parasite chain attack.
- Case studies in the mobility and smart energy industries.
- The current status, process and intentions of the Qubic project is to be analyzed which is IOTA's solution for oracle machines and smart contracts.

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 [10].

3.1 Network Partition

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 Chapter 4.

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.

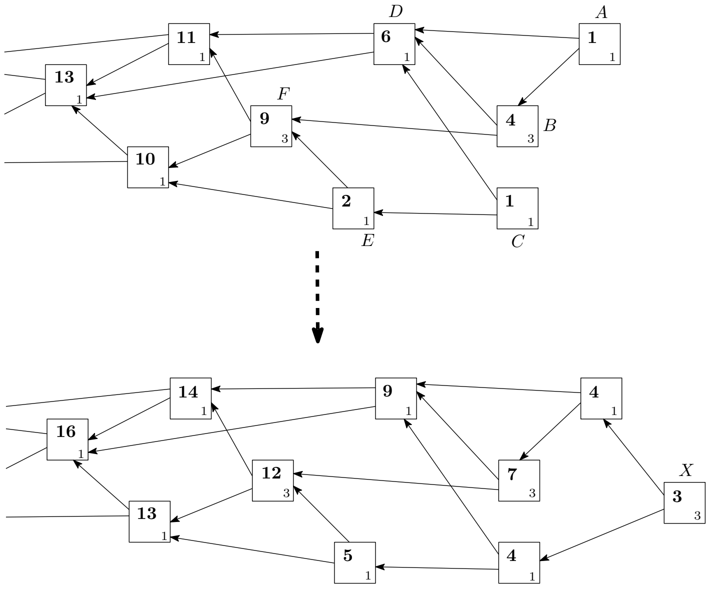

Figure 3.1: Cumulative Weight [10]

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.

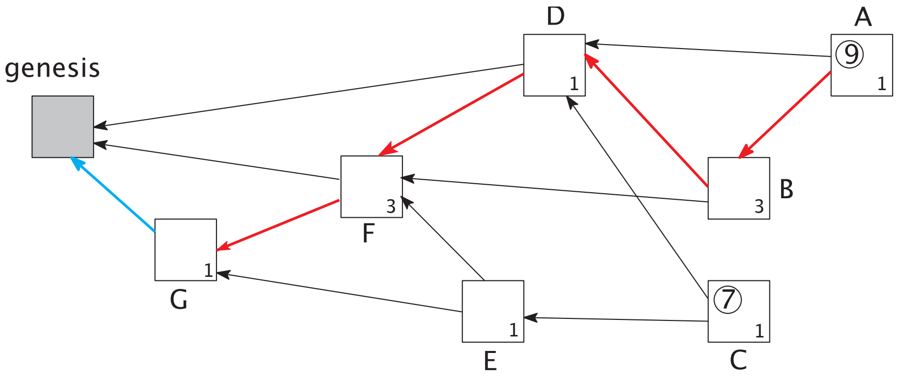

Figure 3.2: Height, Depth and Score [10]

These parameters will play an important role when discussing tip selection in Section 3.3 and attack scenarios 4.

3.2.1 Diffenrent Types of Transactions

There exist four types of transactions in the IOTA protocol.

Output transaction IOTA tokens are sent to another address.

Input transaction with 0 value These transactions facilitate a transfer to a new address in the sender's wallet.

Input transaction with a negative value These transactions completely spend the balance of that account.

Meta transactions These transactions have a value of 0 and make use of the *signatureMessageFragment* to store data on the Tangle.

IOTA represents data according to the trinary numeric system. In comparison to binary, a trinary system is more efficient in terms of computation and memory as it can represent data in three states rather than just two. Hence, a transaction is also tryte-encoded where each tryte consists of one of 27 characters.

A transaction in the IOTA protocol consists of 2673 tryte-encoded characters which are equivalent to 1.59 kBytes. Every transaction object contains the same fields. The most interesting fields are the following two.

Message (2187 trytes) This field can be used for two purposes. In case of an input or output transaction, this field contains the signature. In case of a meta transaction, these trytes can be used for storing data on the tangle. If the data is larger than 2187 trytes, the data is split among multiple transactions and grouped together in a transaction bundle. The tangle treats a transaction bundle just like a transaction. It either accepts the bundle or rejects all transactions within the bundle. The tangle explorer cannot search for a specific message. Instead, tags can be used to localize desired messages.

Tag (27 trytes) This field is used to search for a transaction with a specific tag value.

Due to the design choice of quantum-resistant signature scheme, the protocol does not allow spending multiple times from the same address. Thus, a simple transaction from Alice's wallet to Bob results in the best case in a transaction bundle of three transactions. One for sending IOTA tokens to Bob, the second one for spending all the remaining IOTAs in Alice's wallet and a third one that is a meta transaction and holds the second part of the signature. Thus, the actual size of a basic Alice-to-Bob transaction results in 8019 trytes (4.77 kBytes).

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 $\boldsymbol{\lambda}$ 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 λ. The higher λ is set in the simulation, the more transactions arrive within one time-unit. If λ 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 $\lambda = 5$. 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.

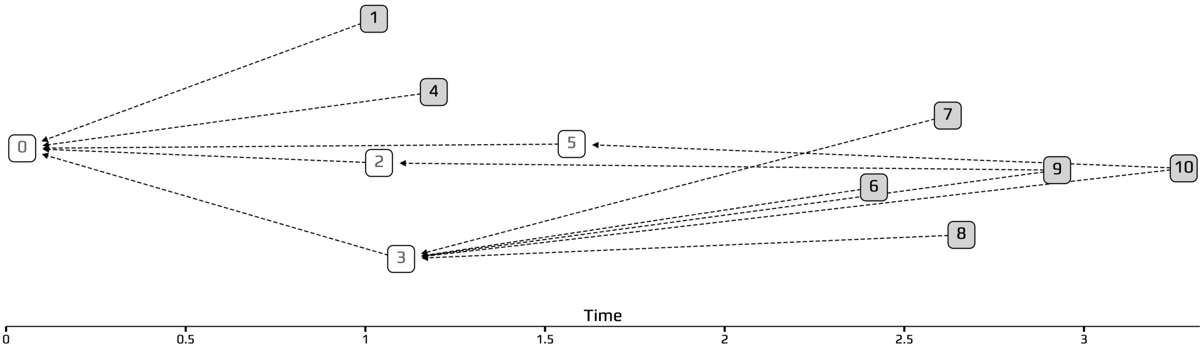

Figure 3.3: Simulation with $\lambda = 5$ and $d = 1$ [13]

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.

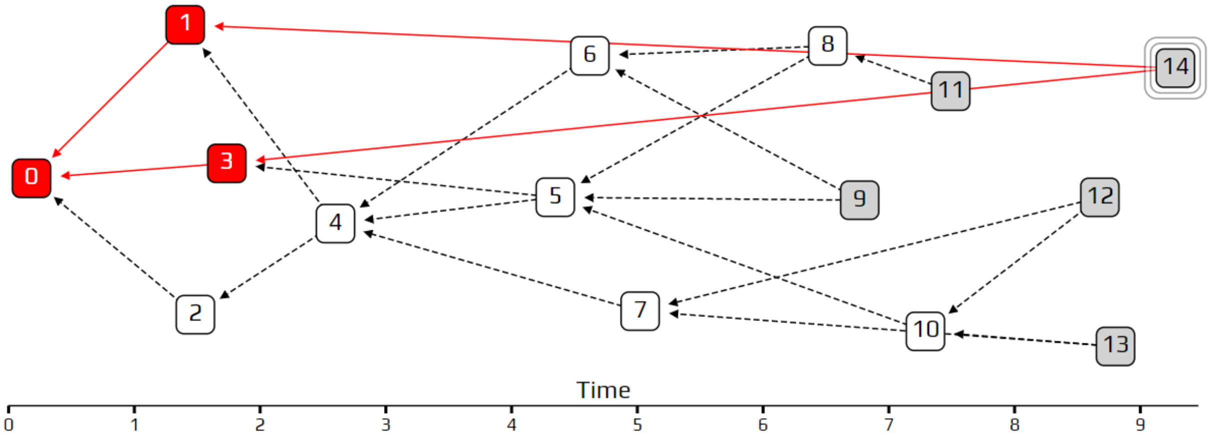

Figure 3.4: Lazy Tip [13]

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) [16]. 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 [10]. 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.

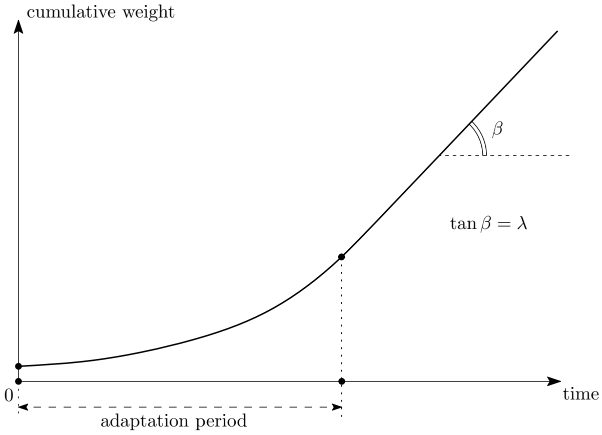

Figure 3.5: Confirmation Confidence and Cumulative Weight Growth [10]

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.

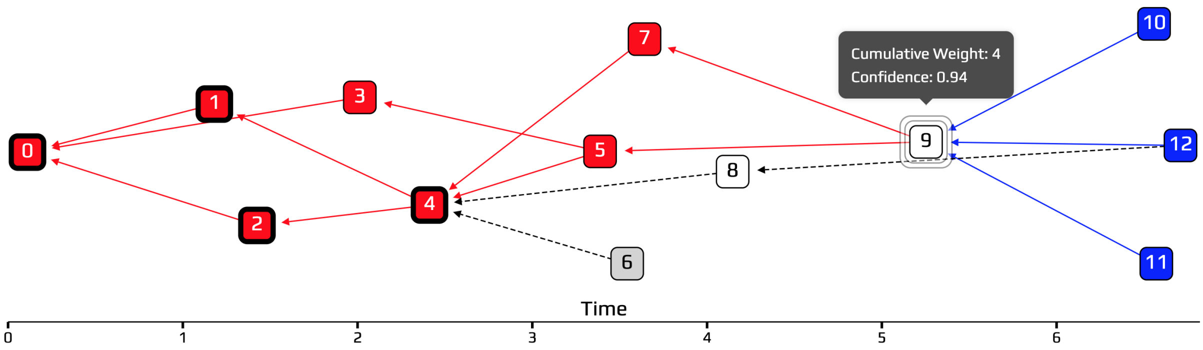

Figure 3.6: Graph Representation of the Confirmation Confidence

3.5 Permanodes, Common Nodes and Local Snapshots

A common node only stores transactions back to the last milestone set by the Coo or with the introduction of IRI 1.6.0 [20], a node can manually adjust the time period for making local snapshots. Whenever a milestone is reached, only the account balances are stored in order to minimize the necessary storage requirements for a common node. Thus, the message field and the tag field are also dropped whenever a snapshot is made. In order to access data from a previous checkpoint, a common node relies on permanodes. An alternative is to subscribe to a specific tag. Whenever an incoming transaction features this tag, the node stores it locally. This requires additional hardware with higher storage capacity that monitor the tangle and store the desired data.

Permanodes are computers with large storage and bandwidth capacity that store the entire history of the Tangle. At the time of writing this thesis, the Tangle grows nearly 3GB in size every week with a total size of 300GB since day one. A permanodes also has the ability to make snapshots in order to reduce the size.

There are permanodes that offer a pay-per-request business model, where data from previous snapshots can be retrieved.

3.6 Coordinator

In the early stages of the Tanlge, the network is susceptible to several attacks. Some of these attacks are discussed in Chapter 4. In summary, a user that controls a majority of the hashing power can double-spend coins. Unlike in Bitcoin where a miner competes with all other miners, in IOTA an attacker only competes with nodes that actively issue transactions. Thus, in times where not many transactions are issued, an attack becomes more feasible to execute.

In order to protect itself against such attacks, the IOTA foundation operates a special node called the Coordinator (Coo). This node has a checkpoint function. By issuing periodically zero-value transactions, the Coo creates milestones. Every transaction that is directly or indirectly confirmed by this milestone is considered as valid. The Coo is a central entity in the tangle and manifests a single point of failure. The Coo is not able to invalidate transactions from previous milestones. However, the node has several privileges compared to a regular node.

1. The foundation can prioritize transactions.
2. The Coo has the ability to censor transactions by continuously not approving certain transactions.
3. If Coo is attacked and no longer works, the entire network halts.

The coordinator-free Tangle is developed by a dedicated research team of the IOTA foundation. There are three concepts proposed for a more decentralized network.

1. **Node accountability** is a reputation system built into the protocol, similar to object reputation systems used in p2p file-sharing such as the Gnutella network [9]. Such a protocol allows making judgments about the authenticity of incoming transactions. The reputation of a node is lowered whenever a node attempts to make a double-spend transaction or issues many re-attachments. Re-attachments are used in IOTA when a transaction is left behind on a branch that is likely to be abandoned.
2. As mentioned in Section 3.3, the **tip selection algorithm** is one of the main difficulties in the network. Without the Coo, there are no milestones from which the MCMC random walk algorithm derives. Thus, a new heuristic algorithm is developed using backtracking from a recent tip until it reaches a transaction with a high cumulative weight.
3. The **Stars Concept** is an idea that works with well-known, public and trusted entities such as governments and corporations. These entities issue reference transactions which are similar to the milestones issued by the Coo.

These approaches are implemented and tested on the so-called zero-value testnet (znet).

3.7 Qubic

The Qubic protocol addresses the integration of smart contracts (SC), oracles and outsourced computation within the IOTA network. The following terminology helps to understand the aim of the protocol.

Qubic (QBC) The protocol receives its name from quorum-based (distributed) computation.

quorum A quorum is the minimum number of votes that a transaction/data must obtain, such that is is considered as valid. The introduction of quorum-based computation makes it more difficult for malicious nodes to falsify data as well as reduce noisy data from faulty sensors.

qubic Besides the protocols name, a qubic is also referred to a packaged quorum-based computation that occurs according to the Qubic protocol. One can think of a qubic as a data/computation request or task on the tangle.

qubic owner The qubic owner is the node that issues the request (qubic). For every qubic, a reward is defined. This reward is split among all nodes that enforce the quorum result. This promotes honest behavior, as a node is not rewarded when publishing a defective result.

(deliberative) assembly A group of oracles forms an assembly where all of its members will process the same set of qubics. Each oracle will post its results for every qubic on the Tangle. The assembly will decide on the true value of the requested data. The threshold of the acceptance rate is usually set to 2/3.

Abra The IOTA foundation develops a functional programming language called Abra. It uses the trinary number system in order to save disk space and computational power.

The Qubic protocol is still in development and is not deployed on any testnet by date of writing this paper.

3.7.1 Oracles

Oracles bring real-world data into the ledger. Difficulties that must be considered are the Sybil attack and the classroom attack, where oracles copy the result of other oracles without measuring the requested data.

Sybil Attack A single oracle could impersonate multiple oracles at the same time in order to receive a larger cut of the reward. Such a Sybil attack is most likely mitigated in the Qubic protocol by weighted voting. An oracle has a voting weight according to the resources it used to solve a crpytographic puzzle (PoW) or according to its stake in the network. These voting weights are set initially when a assembly is formed. However, it can be adjusted when new nodes join the assembly or when the majority of the assembly agrees on a new resource test phase. During this phase, the computational resources of each oracle are examined and the weights are updated accordingly.

Classroom attack Results must be published in a commit-reveal schema, such that oracles cannot copy the results from others without verifying the data.

3.7.2 Outsourced computing

Outsourced computing addresses the problem that not every IoT device is able to execute computationally complex tasks due to memory, computational power and energy availability limitations. As with oracle machines, outsourced computation is handled in a decentralized way, with the Qubic protocol ensuring that the results can be trusted to a high degree of certainty. The protocol allows anyone to request to run a computational task without permission. On the other hand, any node can become part of an assembly which will eventually be assigned to solve computational tasks.

3.7.3 Smart Contracts

Smart contracts facilitate, verify and enforce transactions on the underlying ledger technology without the need of a third party.

3.7.4 Qubic in Action

The following example illustrates how the three building blocks complete the Qubic protocol.

1. The car insurance and the driver establish a **smart contract** that contains variable rates for different driving conditions. The cost depends on multiple factors. This data can be retrieved in a distributed manner by issuing qubics such as a temperature qubic, traffic jam qubic, etc.

2. Autonomous cars could act as a group of **oracles** (assembly) when deciding on traffic congestion. If the quorum is set to 2/3 and 2/3 of the cars in a specific area register a high degree of traffic the tangle registers this information.

3. Analyzing the data from the tangle might be **computationally** expensive. Thus, a new qubic is issued for analyzing the different factors from the tangle and is **outsourced** to a assembly that can compute this task efficiently.

4. When the result of the analyzing qubic is received, the **smart contract** automatically pays the necessary amount for the car insurance according to the driving conditions.

3.8 Economic Clustering

Economic Clustering is a theoretical approach for more scalability. It is inspired by global real-world economies. The global economy is divided up into regional economies. These local economies coexist but do not influence each other.

When this concept of global economies is applied to IOTA, each of these local economies represents a cluster. Clusters run their own tangle. Clusters are not compatible and one can only transact within a cluster. Thus, there is a need for inter-cluster exchanges, which are users or institutions that have funds in multiple clusters.

Chapter 4

Attack Scenarios

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

4.1 Double Spend

A double spend situation occurs when a user tries to exceed his account balance by issuing two or more conflicting transactions. Figure 4.1 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.

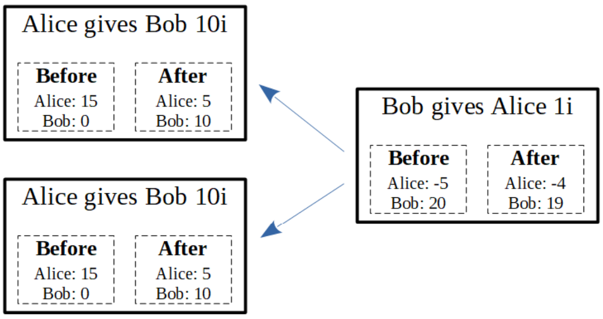

Figure 4.1: Double Spend Attack [15]

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.

4.2 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.

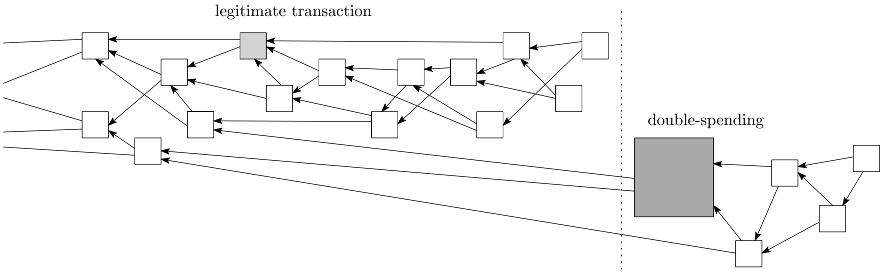

Figure 4.2: Large Weight Attack [10]

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 ?.

4.3 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 4.3.

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.

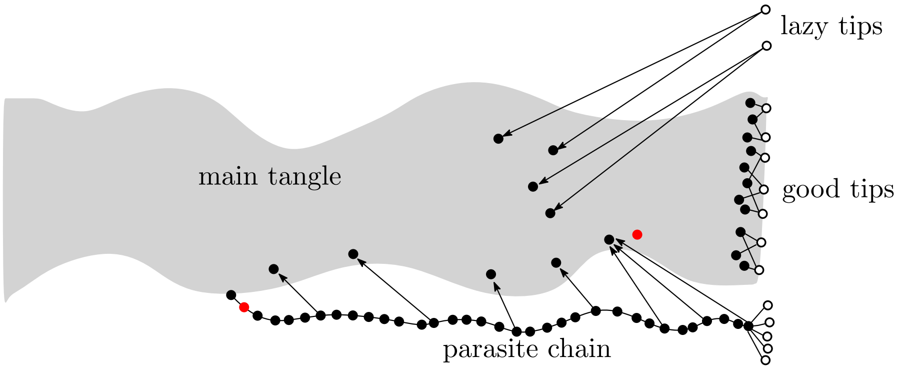

Figure 4.3: Parasite Chain Attack [10]

Chapter 5

Case Studies

5.1 Mobility

5.2 Smart Energy

Chapter 6

Evaluation

As this paper is about the analysis and the evaluation of the potential of the tangle's architecture, this section focuses on the advantages of the protocol as well as the negative points.

6.1 Coordinator

As described in Section 3.6, the current implementation of the tangle uses a special centralized node that marks milestones in order to protect the network against several attacks. However, it creates a single point of failure which is not desired in a P2P network and defeats the purpose of a P2P currency. The IOTA foundation is working on a coordinator-free network. However, as long as the network runs in the current form, the entire network is susceptible to attacks on the coordinator which can result in a halt of the entire system.

6.2 Anti-Spam Mechanism

The current implementation of the tangle requires a PoW for every issued transaction as an anti-spam mechanism and to prevent the attacks described in Section 4.

Most of the time, nodes in the network only forward transactions and update the tangle accordingly and thus, they do not provide security benefits by doing so. A node only makes the network more secure when issuing new transactions. Thus, an attacker in IOTA must only have a greater hash-power than all the incoming transactions within a certain time period. This would allow him to create an alternative subtangle with higher cumulative weight forcing other parts of the tangle to be abandoned.

This is a big difference to PoW blockchains where a miner tries to solve the cryptographic puzzle before every other miner in the network. In order to have the same level of security as Bitcoin, the IOTA network has to exhibit a similarly high network hashing power.

However, one must note that the hashing power is much more distributed compared to the Bitcoin network and make such an attack less likely, whereas in Bitcoin a 51

Furthermore, as IOTA claims to become the network of IoT devices, PoW does not seem to be a sustainable solution to the anti-spam mechanism. Lots of IoT devices run on limited power and computation capacity. Outsourcing the PoW could be a solution but then there is an overhead in maintaining additional hardware that solely computes PoW in order to facilitate the transactions of IoT devices.

A solution based on the acquired stake in the network to prevent fraudulent behavior would make more sense and might be included in the next big system update when the coordinator-less protocol will be developed.

6.3 Permanodes

As explained in Section 3.2.1, a transaction consist of 2673 trytes. However, due to the signature schema in the protocol, a transaction from Alice to Bob is in the best case a bundle of three transactions. This results in a minimum of 4.77 kBytes, whereas a Bitcoin transaction is in average 0.6 kBytes [21] and a basic Alice-to-Bob Ether transaction is about 0.1 kBytes [22]. Thus, storing the history of the tangle is more resource intensive.

The protocol supports local pruning called snapshots. This allows a node to discard all transaction and only store the balances as well as the transactions from within a time period. This period is set by default to 30 days [20]. When making local snapshots the field for the signatureMessageFragment and the tag is no longer stored. Thus, whenever a node wants to access data transactions from before the last snapshot, it relies on permanodes. When receiving data from permanodes, there is no other way to check if the data was not manipulated than asking several permanodes for the same data and compare their response.

Furthermore, the protocol allows storing data on the tangle as described in Section 3.2.1. Storing data on the tangle is relatively cheap as one only pays for the PoW for every transaction. Retrieving data from a permanode on the other hand will most likely not be free since maintaining the entire history of the tangle requires a datacenter. In Ethereum it is the other way around. One has to pay for storing data on the blockchain but reading from the blockchain is free.

6.4 Economic Clustering

IOTA's approach to solving the storage problem is economic clustering as described in Section 3.8. This approach essentially deploys a new tangle for every cluster. The security benfit of having many nodes confirming transaction does not exceed the borders of a cluster. A new cluster can be compared to a hard fork in Bitcoin. When a network splits and both networks coexist, the hash power is split among both chains and thus, they become less secure than the original chain. The same applies to economic clustering.

Chapter 7

Summary and Conclusions

Bibliography

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

[2] Alyssa Hertig: How Will Ethereum Scale?, `https://www.coindesk.com/information/will-ethereum-scale`, accessed 20th of Mai 2019.

[3] Transactions Per Second (TPS): Cryptocurrency And Blockchain Importance Examined, `https://bitcoinexchangeguide.com/transactions-per-second-tps/`, 2nd of September 2018, accessed 20th of Mai 2019.

[4] Jan Vermeulen: VisaNet - handling 100,000 transactions per minute, `https://mybroadband.co.za/news/security/190348-visanet-handling-100000-transactions-per-minute.html`, 17th of December 2016, accessed 20th of Mai 2019.

[5] IOTA Foundation: Coordinator. Part 1: The Path to Coordicide, `https://blog.iota.org/coordinator-part-1-the-path-to-coordicide-ee4148a8db08`, 20th of November 2018, accessed 20th of Mai 2019.

[6] IOTA Foundation: Coordinator. Part 2: IOTA is a DAG, not a Blockchain, `https://blog.iota.org/coordinator-part-2-iota-is-a-dag-not-a-blockchain-2df8ec85200f`, 20th of November 2018, accessed 20th of Mai 2019.

[7] IOTA Foundation: Coordinator. Part 3: Approaches to Coordicide, `https://blog.iota.org/coordinator-part-3-approaches-to-coordicide-583fb82382bc`, 20th of November 2018, accessed 20th of Mai 2019.

[8] IOTA Foundation: Coordinator. Part 4: An Open Source Coordinator, `https://blog.iota.org/coordinator-part-4-an-open-source-coordinator-7d3804931058`, 20th of November 2018, accessed 20th of Mai 2019.

[9] Kevin Walsh, Emin Guen Sirer: Experience with an Object Reputation System for Peer-to-Peer Filesharing, `https://www.usenix.org/legacy/event/nsdi06/tech/walsh/walsh.pdf`, accessed 20th of Mai 2019.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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 December 2017, accessed 20th of Mai 2019.

[17] 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.

[18] 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.

[19] Jordan Tuwiner: Bitcoin Mining Pools, https://www.buybitcoinworldwide.com/mining/pools/, 29th of January, accessed 20th of Mai 2019.

[20] Jakub Cech: IRI 1.6.0 with local snapshots out now!, https://blog.iota.org/iri-1-6-0-with-local-snapshots-out-now-fc4d991faba8, 11th of January 2019, accessed 20th of Mai 2019.

[21] Analysis of Bitcoin Transaction Size Trends, TradeBlock, https://tradeblock.com/blog/analysis-of-bitcoin-transaction-size-trends, 15th of October 2015, accessed 20th of Mai 2019.

[22] Gavin Wood: Ethereum: A Secure Decentralized Generalised Transaction Ledger Byzantium Version, `https://ethereum.github.io/yellowpaper/paper.pdf`, 19th of June 2019, accessed 24th of June 2019.

Abbreviations

MBM	Master Basis Module
SC	Smart Contract
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)
MT	Main Tangle
PC	Parasite Chain
TPS	Transactions Per Second
Coo	short for Coordicile
IRI	IOTA Reference Implementation
CLIRI	Cooridinator-Less IOTA Reference Implementation

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