THE TANGLE

Literary Review

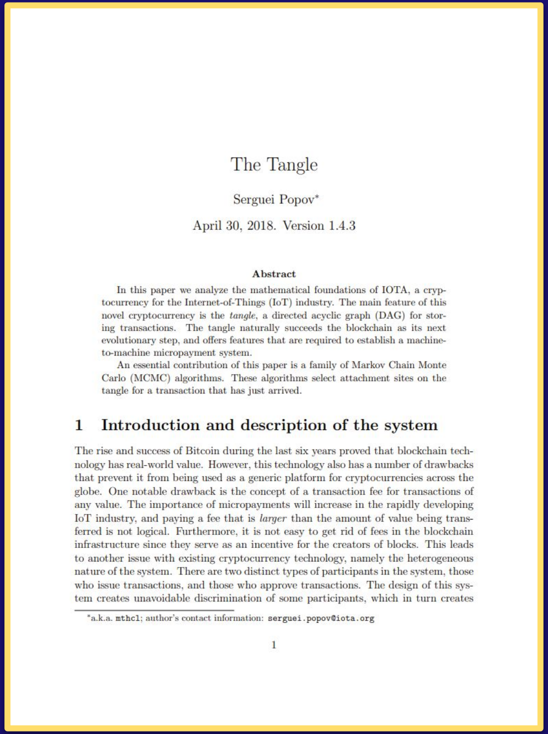
Valerio Di Stefano - Giuseppe Prisco

FIRST ANALYZED PAPER

The Tangle - White paper

2018 - Serguei Popov

http://cryptoverze.s3.us-east-2.amazonaws.com/wp-content/uploads/2018/11/10012054/IOTA-MIOTA-Whitepaper.pdf



The Tangle

Serguei Popov*

April 30, 2018. Version 1.4.3

Abstract

In this paper we analyze the mathematical foundations of IOTA, a cryptocurrency for the Internet-of-Things (IoT) industry. The main feature of this novel cryptocurrency is the *tangle*, a directed acyclic graph (DAG) for storing transactions. The tangle naturally succeeds the blockchain as its next evolutionary step, and offers features that are required to establish a machine-to-machine micropayment system.

An essential contribution of this paper is a family of Markov Chain Monte Carlo (MCMC) algorithms. These algorithms select attachment sites on the tangle for a transaction that has just arrived.

1 Introduction and description of the system

The rise and success of Bitcoin during the last six years proved that blockchain technology has real-world value. However, this technology also has a number of drawbacks that prevent it from being used as a generic platform for cryptocurrencies across the globe. One notable drawback is the concept of a transaction fee for transactions of any value. The importance of micropayments will increase in the rapidly developing IoT industry, and paying a fee that is *larger* than the amount of value being transferred is not logical. Furthermore, it is not easy to get rid of fees in the blockchain infrastructure since they serve as an incentive for the creators of blocks. This leads to another issue with existing cryptocurrency technology, namely the heterogeneous nature of the system. There are two distinct types of participants in the system, those who issue transactions, and those who approve transactions. The design of this system creates unavoidable discrimination of some participants, which in turn creates

*a.k.a. mthcl; author's contact information: serguei.popov@iota.org

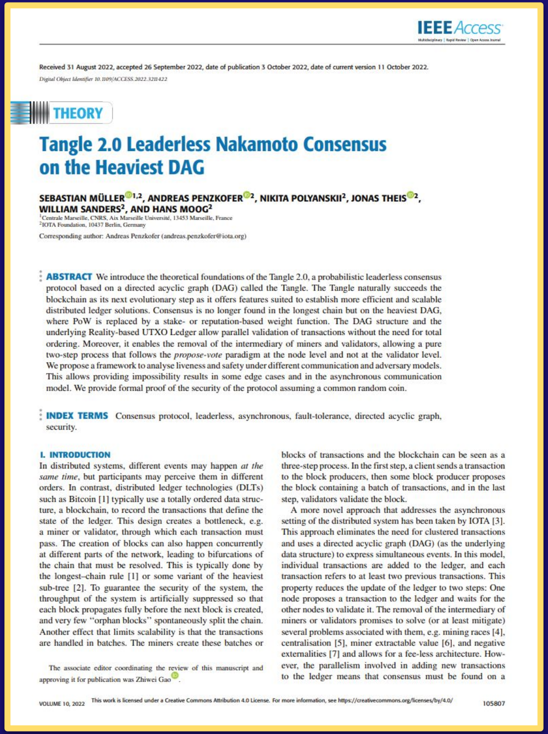
1

SECOND ANALYZED PAPER

Tangle 2.0 Leaderless Nakamoto Consensus on the Heaviest DAG

2022 - Sebastian Müller, Andreas Penzkofer, Nikita Polyanskii, Jonas Theis, William Sanders, Hans Moog

https://ieeexplore.ieee.org/document/9907014



IEEE Access

Multidisciplinary | Rapid Review | Open Access Journal

Received 31 August 2022, accepted 26 September 2022, date of publication 3 October 2022, date of current version 11 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3211422

THEORY

Tangle 2.0 Leaderless Nakamoto Consensus on the Heaviest DAG

SEBASTIAN MÜLLER[1,2], ANDREAS PENZKOFER[2], NIKITA POLYANSKII[2], JONAS THEIS[2], WILLIAM SANDERS[2], AND HANS MOOG[2]

[1]Centrale Marseille, CNRS, Aix Marseille Université, 13453 Marseille, France

[2]IOTA Foundation, 10437 Berlin, Germany

Corresponding author: Andreas Penzkofer (andreas.penzkofer@iota.org)

ABSTRACT We introduce the theoretical foundations of the Tangle 2.0, a probabilistic leaderless consensus protocol based on a directed acyclic graph (DAG) called the Tangle. The Tangle naturally succeeds the blockchain as its next evolutionary step as it offers features suited to establish more efficient and scalable distributed ledger solutions. Consensus is no longer found in the longest chain but on the heaviest DAG, where PoW is replaced by a stake- or reputation-based weight function. The DAG structure and the underlying Reality-based UTXO Ledger allow parallel validation of transactions without the need for total ordering. Moreover, it enables the removal of the intermediary of miners and validators, allowing a pure two-step process that follows the *propose-vote* paradigm at the node level and not at the validator level. We propose a framework to analyse liveness and safety under different communication and adversary models. This allows providing impossibility results in some edge cases and in the asynchronous communication model. We provide formal proof of the security of the protocol assuming a common random coin.

INDEX TERMS Consensus protocol, leaderless, asynchronous, fault-tolerance, directed acyclic graph, security.

I. INTRODUCTION

In distributed systems, different events may happen *at the same time*, but participants may perceive them in different orders. In contrast, distributed ledger technologies (DLTs) such as Bitcoin [1] typically use a totally ordered data structure, a blockchain, to record the transactions that define the state of the ledger. This design creates a bottleneck, e.g. a miner or validator, through which each transaction must pass. The creation of blocks can also happen concurrently at different parts of the network, leading to bifurcations of the chain that must be resolved. This is typically done by the longest–chain rule [1] or some variant of the heaviest sub-tree [2]. To guarantee the security of the system, the throughput of the system is artificially suppressed so that each block propagates fully before the next block is created, and very few "orphan blocks" spontaneously split the chain. Another effect that limits scalability is that the transactions are handled in batches. The miners create these batches or blocks of transactions and the blockchain can be seen as a three-step process. In the first step, a client sends a transaction to the block producers, then some block producer proposes the block containing a batch of transactions, and in the last step, validators validate the block.

A more novel approach that addresses the asynchronous setting of the distributed system has been taken by IOTA [3]. This approach eliminates the need for clustered transactions and uses a directed acyclic graph (DAG) (as the underlying data structure) to express simultaneous events. In this model, individual transactions are added to the ledger, and each transaction refers to at least two previous transactions. This property reduces the update of the ledger to two steps: One node proposes a transaction to the ledger and waits for the other nodes to validate it. The removal of the intermediary of miners or validators promises to solve (or at least mitigate) several problems associated with them, e.g. mining races [4], centralisation [5], miner extractable value [6], and negative externalities [7] and allows for a fee-less architecture. However, the parallelism involved in adding new transactions to the ledger means that consensus must be found on a

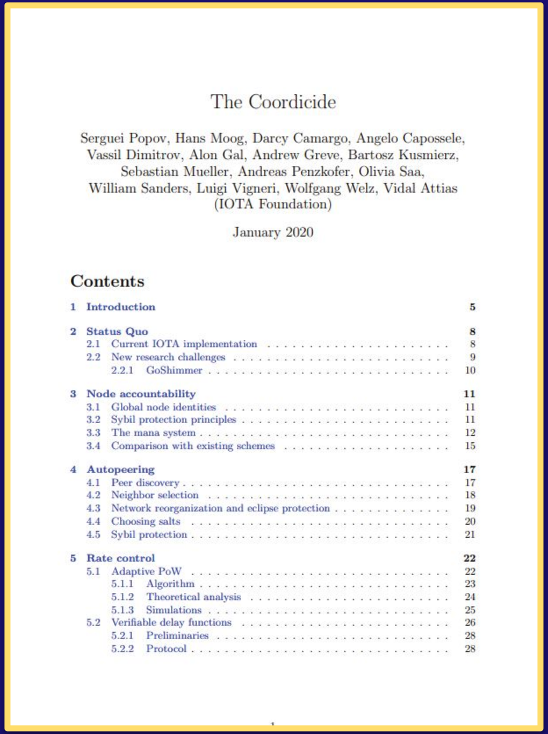
The associate editor coordinating the review of this manuscript and approving it for publication was Zhiwei Gao.

(EXTRA) THIRD ANALYZED PAPER

The Coordicide (WP)

2020 - S. Popov, H. Moog, D. Camargo, A. Capossele, V. Dimitrov, A. Gal, A. Greve, B. Kusmierz, S. Mueller, A. Penzkofer, O. Saa, W. Sanders, L. Vigneri, W. Welz, and V. Attias

https://files.iota.org/papers/20200120_Coordicide_WP.pdf



The Coordicide

Serguei Popov, Hans Moog, Darcy Camargo, Angelo Capossele, Vassil Dimitrov, Alon Gal, Andrew Greve, Bartosz Kusmierz, Sebastian Mueller, Andreas Penzkofer, Olivia Saa, William Sanders, Luigi Vigneri, Wolfgang Welz, Vidal Attias
(IOTA Foundation)

January 2020

Contents

STRUCTURE OF OUR REVIEW

1. **INTRODUCTION**
   Distributed Ledgers, Blockchain Technology, Advantages & Disadvantages

2. **THE TANGLE**
   Introduction to “The Tangle” as envisioned by the 1st and 2nd analyzed papers

3. **ALGORITHMS AND PROTOCOLS**
   Proposed algorithms and protocols of the 1st and 2nd analyzed papers

4. **PERFORMANCE ANALYSIS**
   Issued transactions per second and Time To Confirmation

5. **ATTACK SCENARIOS**
   Attack Scenarios and Security of the 1st and 2nd analyzed papers’ solutions

6. **IOTA, COORDICIDE & ADDITIONAL NOTES**
   Brief history of IOTA Protocol, The Coordicide, Additional Notes and Comments

1

INTRODUCTION

Distributed Ledgers,
Blockchain Technology,
Advantages & Disadvantages

DISTRIBUTED LEDGERS

Shared, replicated and synchronized data structure containing information distributed across many sites

Centralized Database

Central Administration

Single Point of Failure

Distributed Ledger

Peer-to-Peer Network

Replicated data structure

Consensus Mechanisms

DLTs CATEGORIZATION & CHARACTERISTICS

Data Structures

Linear (Blockchain)
Complex (DAG)

Consensus Algorithms

PoW, PoS, PoX,
DAG Consensus &
Voting Protocols

Permissions (on data access)

Permissionless
(Public),
Permissioned
(Private)

BLOCKCHAIN

A blockchain-based distributed ledger consists of **lists of records** securely linked together via **cryptographic hashes** of previously created records

Permissionless Blockchain

Public Network
(every node can read/write)

Consensus mechanism
for agreement (leader election)

PoW, PoS, PoX

Permissioned Blockchain

Private Network
(read/write access is controlled)

Network Administrators
manage data

ADVANTAGES OF BLOCKCHAIN

1. DECENTRALIZATION
   > No “single point of failure”
   > No central administration control

2. IMMUTABILITY
   > Blocks “finality”
   > Trust for stored data

3. TRANSPARENCY
   > Easy verification of stored data
   > Trust on the network

4. TRACEABILITY
   > Tracing of stored data
   > Tracing of changes on the network

DRAWBACKS OF BLOCKCHAIN

1. TRANSACTION FEES
   > Needed to incentivize block creation
   > Discourage micropayments

2. HETEROGENEOUS SYSTEM
   > Two roles (transactions issuers & validators)
   > Can lead to conflicts

3. SCALABILITY ISSUES
   > Heavy operations needed to issue a transaction
   > Lower throughput with a large network

4. BLOCKCHAIN TRILEMMA
   > Evolution of the the FLP impossibility for DLTs
   > Balance security, scalability and decentralization

2

THE TANGLE

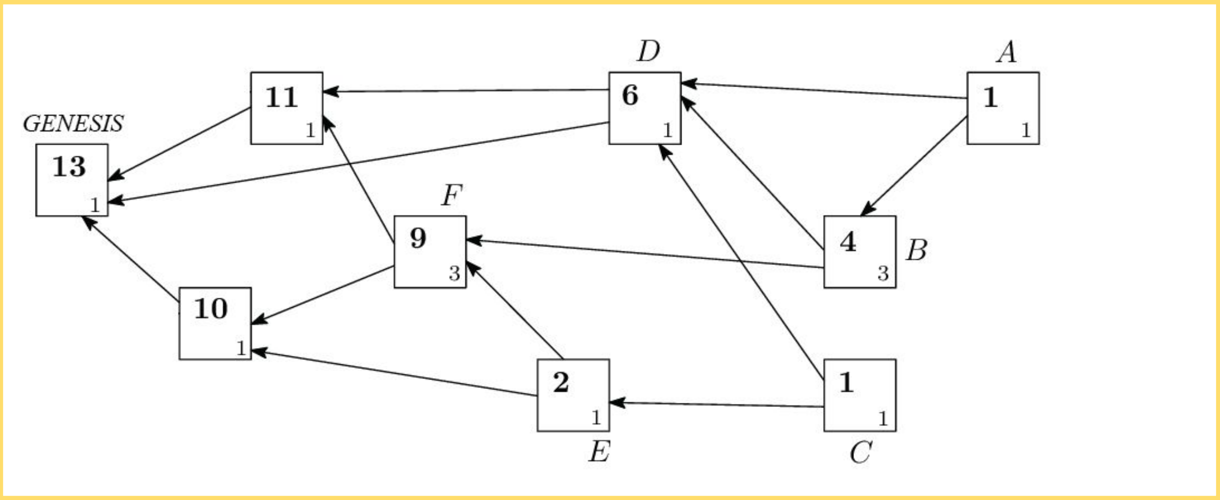
(PART 1)

Introduction to “The Tangle” as envisioned
by the **1st** analyzed papers

THE TANGLE

Directed Acyclic Graph (DAG) as a ledger

- > **Transactions** & Vertices (Sites)
- > **Approvals** & Edges (Direct / Indirect)
- > Tips & Genesis Site
- > Nodes "**work**" to validate other transactions



THE TANGLE

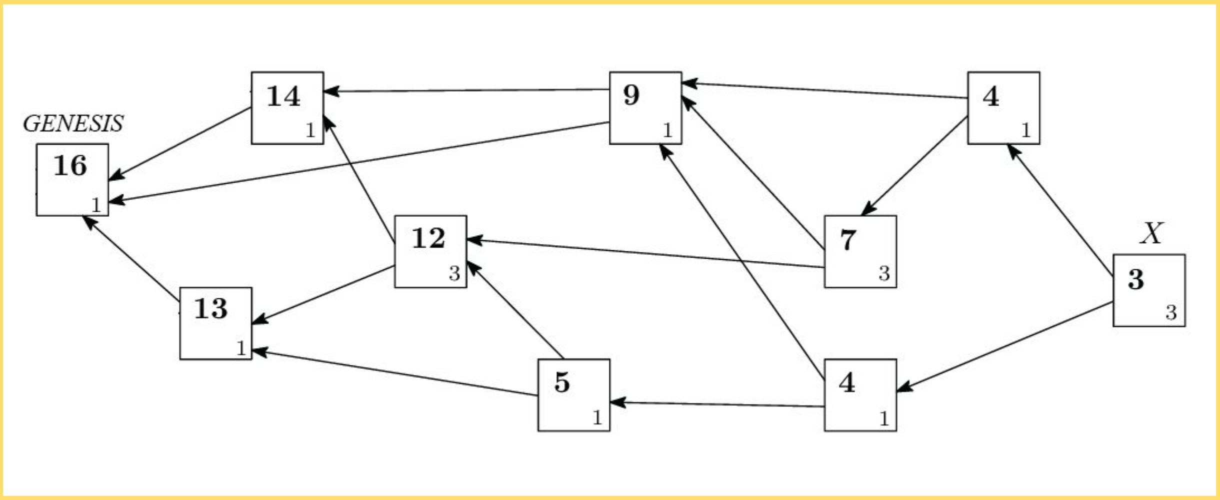
Directed Acyclic Graph (DAG) as a ledger

> **Transactions** & Vertices (Sites)

> **Approvals** & Edges (Direct / Indirect)

> Tips & Genesis Site

> Nodes “**work**” to validate other transactions



THE TANGLE

CONFLICTS

Conflicting transactions

Conflicts need to be **solved**

CONSENSUS

Mechanism to reach an **agreement** on confirmed transactions

WEIGHTS

"Own" weight

Cumulative weight

3

ALGORITHMS AND PROTOCOLS

(PART 1)

Proposed algorithms and protocols of the
1st analyzed papers

Tip Selection Algorithms

Consensus finding		Tip Selection Algorithms
Agreement on conflicting transactions	→	Mechanism for selecting tips to approve

Tip Selection Algorithms
Random selection

Uniform probability for tip selection

→

Not practical because of existence of “lazy” or malicious nodes

Tip Selection Algorithms
MCMC Algorithm

- Based on the use of **random variables** to construct possible events
- Probability of choosing a site on the Tangle proportional to its **cumulative weight**
- N independent "**walkers**" which move towards the tips
- Various strategies to choose tips reached by the walkers

2

THE TANGLE

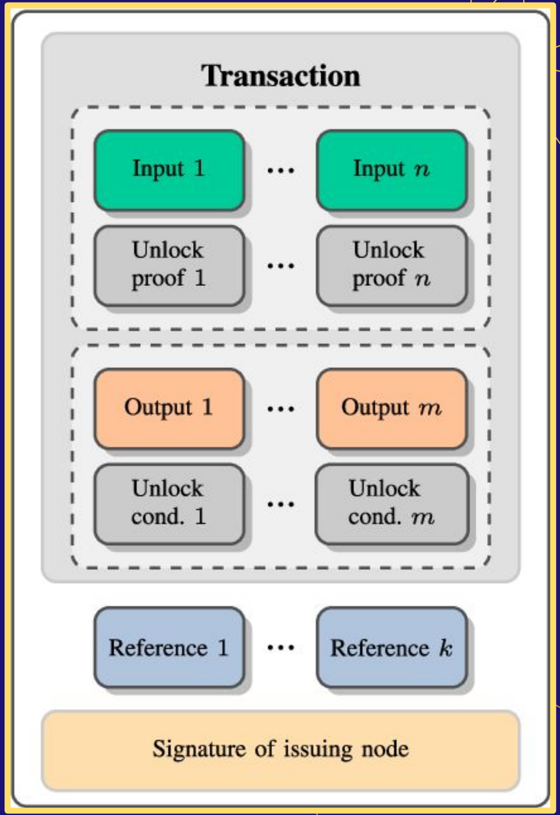
(PART 2)

Introduction to “The Tangle” as envisioned
by the **2nd** analyzed papers

The Tangle

Characteristics

- Generalization of first paper's proposal
- Unspent Transaction Output model (**UTXO**)



The Tangle

Problems of previous approach

1. Reliance on **PoW** in issuing transactions
2. **Vulnerability** to various attacks
3. **Liveness** problem

Proposed solutions

1. Reputation-based **Weight function**

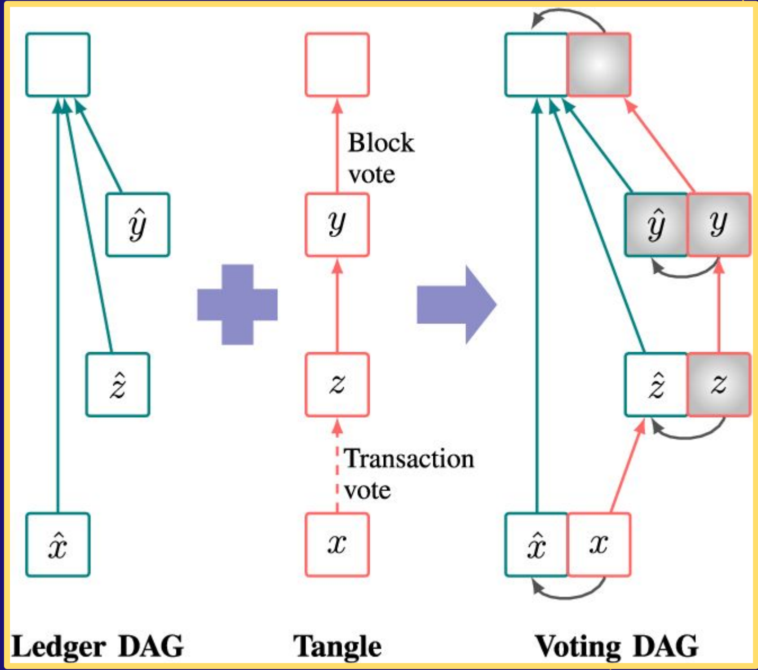
2. Probabilistic asynchronous protocol, with synchronization of nodes at certain time intervals

The Tangle - Proposed solutions

3.

- Separation of transactions from their containers
- Separation of the DAG structure in **two different graphs**

- Retrieve missing blocks (solidification process)
- Retrieve non-conflicting orphaned transactions



3

ALGORITHMS AND PROTOCOLS

(PART 2)

Proposed algorithms and protocols of the **2nd** analyzed papers

PROPOSED PROTOCOL

On Tangle Voting Protocol (OTV)

Asynchronous Layer

General solution for fully **asynchronous systems**

Guarantees **eventual consistency** for data of the shared ledger structure

Synchronized Layer

Allows for a **partial synchronization** of nodes at fixed time intervals

Based on a **dRNG** (common random coin)

NETWORK & DAGs

- Each **node** has a **weight** (a function of scarce resources)
- Each **block** of the Tangle DAG has a **Witness Weight** (percentage of nodes that witnessed the block's creation)
- Each **transaction** of the Ledger DAG has an **Approval Weight** (Percentage of nodes that approve the transaction)

REALITY SELECTION ALGORITHM

- > Nodes maintain their own **local** Ledger DAG (network delays)
- > **Reality**: sub-graph of the local Ledger DAG with no conflicts
- > In case of conflicts, nodes choose their own preferred reality using a **Reality Selection Algorithm**

Algorithm 1: Reality Selection in Conflict Graph

Data: Conflict Graph $G_{\mathcal{C}} = (\mathcal{C}, E)$

Result: reality $R \in \mathcal{B}$

$R \leftarrow \emptyset$

$U \leftarrow \mathcal{C}$

while $|U| \neq 0$ **do**

  $c^* \leftarrow \arg\max\{\mathbf{w}(c) : c \in \max_{\mathcal{C}}(U)\}$; `/* use min hash(c) for breaking ties */`

  $R \leftarrow R \cup \{c^*\}$

  $U \leftarrow U \setminus \{N_{\mathcal{C}}(c^*) \cup \{c^*\}\}$

end

TIP SELECTION ALGORITHM

- Nodes need to **let other nodes know** about their preferred reality
- Nodes express a preference on transactions in their preferred reality with a “**voting mechanism**” (**OTV**)
- Votes are expressed through the **reference of blocks** on the shared Tangle DAG
- Blocks to reference are chosen using a “**Tip Selection Algorithm**” (R-URTS)

Algorithm 3: Uniform Random Tip Selection Restricted on Reality R

Data: Tangle DAG $D_{\mathcal{T}}$, Ledger DAG $D_{\mathcal{L}}$, preferred reality $R \in \mathcal{B}$, number of references k

Result: tips $L_{\mathcal{T}} \cup L_{\mathcal{L}}$

$L_{\mathcal{T}} \leftarrow \emptyset$
$L_{\mathcal{L}} \leftarrow \emptyset$
$cnt \leftarrow 0$
while $cnt < k$ **do**
    Choose tip x uniformly at random in $D_{\mathcal{T}}$
    Set $Q_{\mathcal{V}}$ to be conflicts contained in $\text{cone}_{\mathcal{V}}^{(p)}(x)$
    Set $Q_{\mathcal{L}}$ to be conflicts in $\text{cone}_{\mathcal{L}}^{(p)}(\hat{x})$
    if $Q_{\mathcal{V}} \subseteq R$ **then**
        $cnt \leftarrow cnt + 1$
        $L_{\mathcal{T}} \leftarrow L_{\mathcal{T}} \cup \{x\}$
    else
        if $Q_{\mathcal{L}} \subseteq R$ **then**
            $cnt \leftarrow cnt + 1$
            $L_{\mathcal{L}} \leftarrow L_{\mathcal{L}} \cup \{x\}$
        end
    end
end

METASTABILITY & IMPOSSIBILITY RESULTS

- **Attacks** on the OTV protocol are still possible in some edge cases
- Attacks lead to **impossibility results** in an **asynchronous setting**
- A “second layer” of the OTV protocol introduces **partial synchrony** for nodes to avoid “**metastability**” situations

REALITY SELECTION ALGORITHM (2nd Version)

- Nodes choose their preferred reality with a **shared degree of randomness** (dRNG)
- Randomness is used to **interfere with possible attackers**
- The reality selection algorithm aims to quickly reach a “**pre-consensus**” state
- **F**rom there, consensus will be eventually reached for all nodes on a certain reality
- The **R-URTS algorithm** is then used for tip selection on the chosen reality

---

Algorithm 4: Reality Selection Algorithm With Common Coin

Data: Conflict Graph $G_{\mathcal{C}} = (\mathcal{C}, E)$, common randomness X distributed uniformly in $[0.5, \theta]$

Result: preferred reality $R \in \mathcal{B}$

$R \leftarrow \emptyset$

$U \leftarrow \mathcal{C}$

while $|U| \neq 0$ **do**

    $c^* \leftarrow \arg\max\{\mathbf{AW}(c) : c \in \max_{\mathcal{C}}(U)\}$; `/* use` $\max \mathrm{hash}(c)$ `for breaking ties */`

    if $\mathbf{AW}(c^*) > X$ **then**

        $R \leftarrow R \cup \{c^*\}$

        $U \leftarrow U \setminus \{N_{\mathcal{C}}(c^*) \cup \{c^*\}\}$

    else

        break the while-loop

    end

end

while $|U| \neq 0$ **do**

    $c^* \leftarrow \arg\max\{\mathrm{hash}(c||X) : c \in \max_{\mathcal{C}}(U)\}$

    $R \leftarrow R \cup \{c^*\}$

    $U \leftarrow U \setminus \{N_{\mathcal{C}}(c^*) \cup \{c^*\}\}$

end

4

PERFORMANCE ANALYSIS

Issued transactions per second,
Time To Confirmation

Most important metrics in DLTs

Number of issued Transactions per second

How many transactions are created and introduced in the system

Time to Confirmation

The time that a transaction needs in order to be confirmed

Number of issued Transactions

- Each node of the network can be modeled as an **independent entity** that issues transactions
- The times between two successive transactions are independent and exponentially distributed
- The process of incoming transactions is described by a Poisson distribution with rate λ

Regimes of load

Low Load Regime

Tip pool size is small, it is unlikely that different transactions reference the same tip

High Load Regime

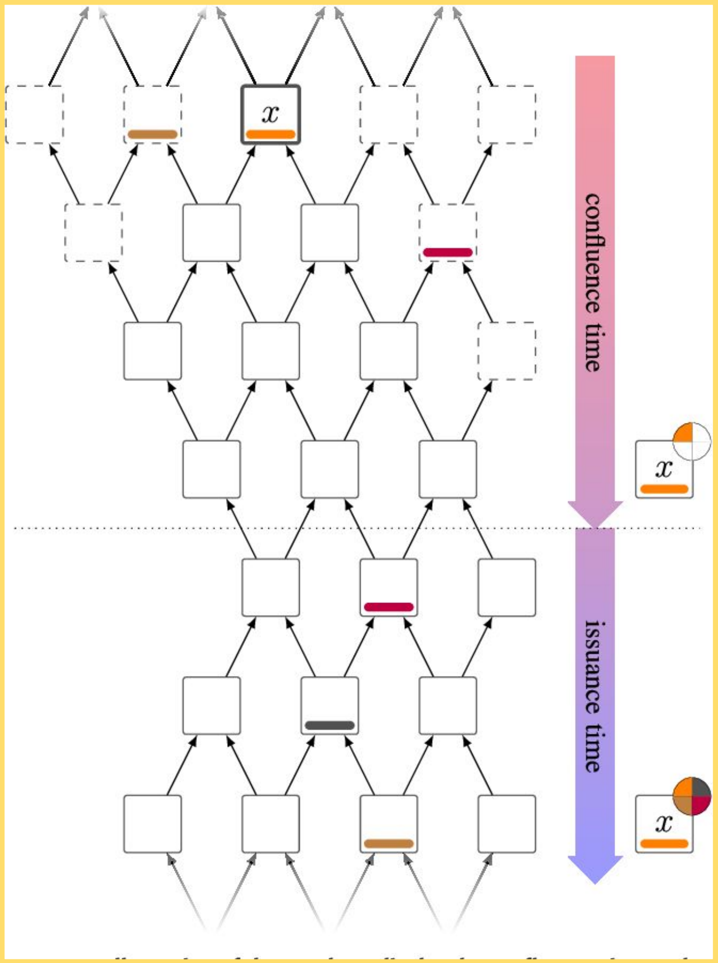
Large number of tips in the system, more likely to have different transactions approving the same tip

Time to Confirmation (high load)

The Time to Confirmation (TTC) can be divided in two phases:

- **Confluence time** t_c
- **Issuance time** t_{iss}

$$TTC \approx \frac{h}{log(k)} \cdot \ln(\lambda h) + \frac{N}{\lambda} \cdot (-\ln(1-\theta))$$



5

ATTACK SCENARIOS

Attack Scenarios and Security of the 1st and 2nd analyzed papers' solutions

Attacks presented by the first paper

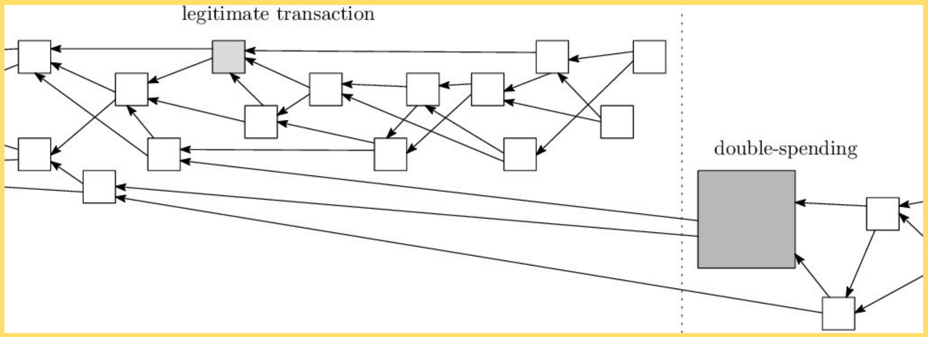
Double-spending attack

Parasite chain attack

Splitting attack

Double-spending attack

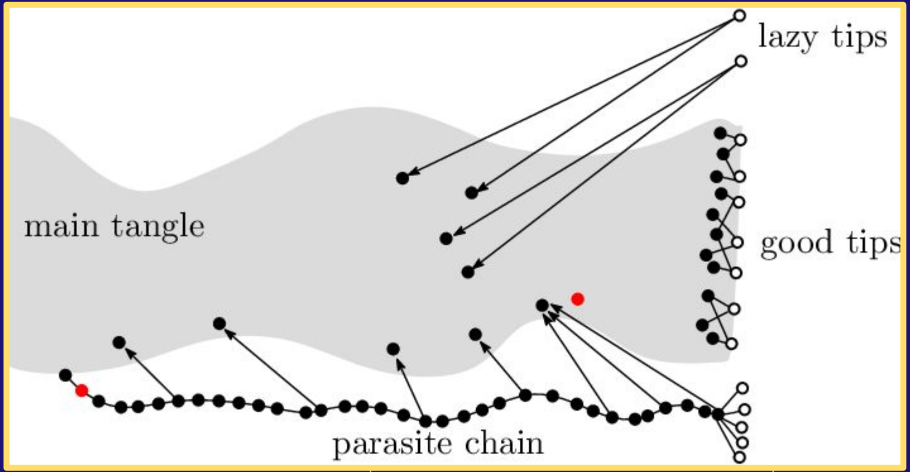
- The attacker issues a “honest” transaction and waits until it is confirmed
- The attacker uses his computing power to issue a double-spending transaction and:
  - Either produces many small transactions approving it
  - Or directly issues a single big double-spending transaction
- The attacker hopes that the network converges to this malicious branch



A possible solution would be to cap the own weight of issued transactions

Parasite chain attack

- The attacker issues a malicious transaction and starts building his own local subtangle
- The attacker grows his subtangle and occasionally refers to the main tangle, gaining a high score for his chain
- The attacker issues a “honest” transaction on the main tangle and waits for its confirmation
- The attacker then inflates the number of tips in his subtangle hoping that the network converges to this malicious branch



A mitigation would be to use a suitable tip selection algorithm

Splitting attack

The goal of the attacker is to split the main tangle in **two parallel branches** by issuing a double-spending transaction

If the attacker is able to keep the weight of the two parallel branches balanced, he will succeed in the attack by spending the same funds on both branches

To avoid this scenario, one can choose a rapidly decaying function for the transition probability of particles of the MCMC algorithm

With this type of transition rule for the tip selection algorithm the nodes of the network would converge into a single tip of one of the two parallel branches even if the difference in cumulative weight between them is very small

Attacks presented by the second paper

Attacks on Communication level

The attacker is able to delay honest nodes' packages

Attacks on Voting level

The attacker can issue blocks at a high rate in contrast to honest nodes

Attacks on Communication and Voting level

The attacker has a strong control over both levels

Attack on Communication level

Metastability attack

The goal of the attacker is to **delay** the confirmation of blocks voted by other honest nodes

Honest nodes will change their vote

The attacker repeatedly forces honest nodes in an **undecided state**

Attacks on Voting level

Metastability attack

The attacker as a high block issuance rate

The malicious nodes of the attacker continuously change their vote

Honest nodes will try to follow this change in votes

Honest nodes remain in an undecided state

Bait-and-switch attack

The attacker possesses a high weight

The attacker will alternate his vote between conflicting transactions

Honest nodes will switch their votes in an endless loop

Attack on Communication and Voting level

If the attacker has weight $q > \theta - 0.5$ the **safety** property is broken

The attacker is able to make honest nodes confirm two different conflicting transactions

Honest nodes are divided into two groups and vote for different transactions

SECURITY GUARANTEES

Theorem 1

Random block issuance and **random package delay** assumptions

The system **eventually converges** on a consensus state (with probability close to 1) if $\mathbf{W_{attacker} < 0.5\ W_{network}}$

Eventual consistency of the “asynchronous layer” of the OTV protocol

No **safety guarantees** nor conclusions about **finality of transactions** are given

Theorem 2

Stronger **probabilistic synchronicity assumptions** and **reliance on a dRNG**

The system **eventually converges** on a consensus state if the portion of the total weight of the network controlled by the attacker is both $\mathbf{q < 1 - \theta}$ and $\mathbf{q < \theta - 0.5}$

Gives guarantees about **liveness and safety** of the “synchronized layer” of the OTV protocol and allows to estimate consensus time

Does **not** require random block issuance and random package delay assumptions

6

IOTA, COORDICIDE & ADDITIONAL NOTES

Brief history of IOTA Protocol,
The Coordicide, Additional Notes and
Comments

IOTA

An open-source cryptocurrency designed for the IoT industry

Uses a DAG structure as a ledger, abandons the issuer/validator paradigm (removing blockchain's leader election bottleneck) and requires no fees

IOTA 1.0

Based on the original white paper proposal

RW-MCMC, **PoW** and **Quantum Computation**

IOTA 1.5

Temporary solution to 1.0 problems

Introduces a "**coordinator**" node

IOTA 2.0

Removes the "coordinator"

Fully decentralized

Work in progress

IOTA 1.0 - CHARACTERISTICS

1. **DAG -BASED**
   Based on a DAG ledger structure for transactions

2. **RW-MCMC**
   Cumulative weight biased random walks as a consensus mechanism

3. **FIXED PoW**
   Nodes need to provide a fixed amount of PoW to write on the Tangle

4. **QUANTUM COMPUTATION**
   Ternary Logic and “Winternitz One Time Signature” scheme (WOTS)

IOTA 1.0 - PROBLEMS

1. PoW RELIANCE

   Leads to a tradeoff between **security** (high PoW) and **access control** (low PoW)

2. CONFLICTS SPAMMING

   Lowers **transaction throughput** (because of higher convergence times)

3. PERFORMANCE ISSUES

   Slow TSA (reliance on cumulative weight), **slow ternary logic operations**, **scalability issues** (one time signatures)

4. ATTACKS & LEGAL ISSUES

   Real life phishing, scamming, and hacking attempts and vulnerability disclosure issues

IOTA 1.5 - CHARACTERISTICS

1. COORDINATOR NODE

Issues "**milestones**" which **totally order** transactions

2. UTXO

No "Account-Based" model

Easy **conflicts detection**

3. PERFORMANCE IMPROVEMENTS

No quantum computation (ternary logic & WOTS)

No RW-MCMC (coordinator takes care of conflicts)

4. LOW PoW REQUIREMENTS

Coordinator takes care of security (no need for high PoW)

Leds to easy network access

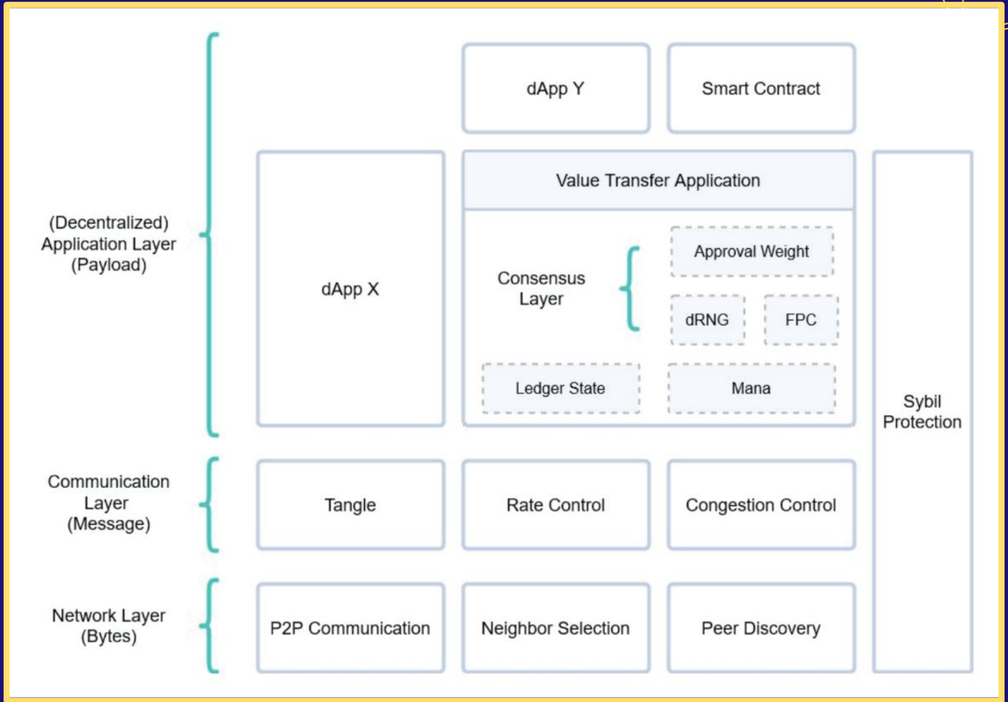
IOTA 2.0 - COORDICIDE

Aims to achieve the first **fully decentralized, secure and highly scalable** DLT

APPLICATION LAYER
Core applications ran by nodes (consensus applications or dRNGs)
Third party applications which can be integrated with the IOTA protocol (e.g. mart contracts)

COMMUNICATION LAYER
Maintains the **Tangle DAG structure**

NETWORK LAYER
Maintains **connections and direct communication** between nodes



COORDICIDE - CHARACTERISTICS

1. MANA
Access Mana (determines how many messages a node can issue)

   Consensus Mana (determines the AW of a transaction)

2. NEW MESSAGE STRUCTURE
Introduction of "**timestamps**"

   Introduction of message "**payloads**"

3. AUTOPEERING
Mechanism to allow nodes to choose their own **network neighbors** based on Consensus Mana

4. ADAPTIVE PoW
Small PoW needed to issue one transaction

   Increasing PoW to issue more transactions in a short time

5. TIP SELECTION ALGORITHM
Based on **R-URTS**

   Introduces "**approval switches**" (weak and strong approvals)

6. FAST PROBABILISTIC CONSENSUS
Based on a **dRNG**

   Randomized confirmation threshold for the AW of transactions

Additional Notes

1. Throughput

Increases as size of the network increase

Scalability directly proportional to throughput

2. Energy Consumption

PoS-like Weight function

No need of PoW as "Sybil Protection" mechanism

UTXO Ledger

No need to consider the entire ledger history

Less waste of Energy