IOTA WORKSHOP

IOTA Basics/101

Speaker(s): Dharmen Dhulla

Event Organizers

dcentrum Community

Connect | Collaborate | Create

Venue Support

Tech Mahindra

Agenda

- What is IOTA and How it Works?
- IOTA Features
- Key Components of IOTA
- IOTA Transactions
- Transactions in Bundle
- IOTA Transaction Attributes & Types
- Other Features

What is IOTA?

IOTA is a distributed ledger technology that allows computers in an IOTA network to transfer immutable data and value among each other.

IOTA aims to improve efficiency, increase production, and ensure data integrity in a machine-to-machine economy.

How IOTA works?

- Clients send data and IOTA tokens to each other through nodes
- Package transactions into bundles
- Transactions in a bundle instruct transfer of IOTA tokens from one address to another
- Addresses are generated by using a client's secret password called a seed
- When the bundle is confirmed in the Tangle, the IOTA tokens are transferred

Background of IOTA

- Setup in 2015 in Germany as non profit foundation
- Raised 1337 BTC as Initial Coin Offering
- Total IOTA's is $(3^{33} - 1)/2 = 2779530283277761$
- All Tokens are mined and sold to ICO Investors
- Mainnet is live since July 11, 2016

IOTA Features

- Scalability – Network becomes stronger as number of transactions increases, the confirmation rates get better
- Decentralization – Every transaction maker is also a transaction validator
- No transaction Fees – IOTA can be used for micropayments
- Quantum computing protection – Quantum computing is still in early stages, estimated to arrive by 2030. It will “crack” current data encryption methods much faster. IOTA uses Winternitz One-Time signature scheme which is quantum-resistant

Key Components of IOTA

- Trits – A trit is a digit in a base 3 number system: either -1, 0, 1 (balanced ternary computing)
- Trytes – A tryte consists of 3 trits
- Seed - A seed is a secret password that's used to create addresses and to sign bundles whose transactions withdraw IOTA tokens. An IOTA seed consists 81 characters which is same as 81 trytes
- Addresses – With the seed the IOTA wallet can generate corresponding Private and Public addresses
- Bundle – IOTA uses bundles which consists of multiple transactions

Key Components of IOTA

- Transaction – Transaction is an object containing fields such as address, signature, value and tag. Each transaction always validates 2 previous non validated transactions
- TIPS – Unconfirmed transactions in the tangle graph
- Tangle – A Tangle is a data structure based on Directed Acyclic Graph (DAG)
- IRI Node

TRINARY NUMERAL SYSTEM

- Trit means Trinary Digit, analogous to bit and has the following values: -1, 0, 1 **0**
- Trytes means Trinary Byte, analogous to byte. A tryte consists of 3 trits **-1** **0** **1**
- Convert tryte -1, 1, 0 to integer:
  $-1 \times 3^0 + 0 \times 3^1 + 1 \times 3^2 = 7$
- IOTA development team has created the tryte alphabet to make it more human readable:
  9ABCDEFGHIJKLMNOPQRSTUVWXYZ
- Since 1 tryte has $3^3 = 27$ combinations, each tryte can be represented by a character in the tryte alphabet

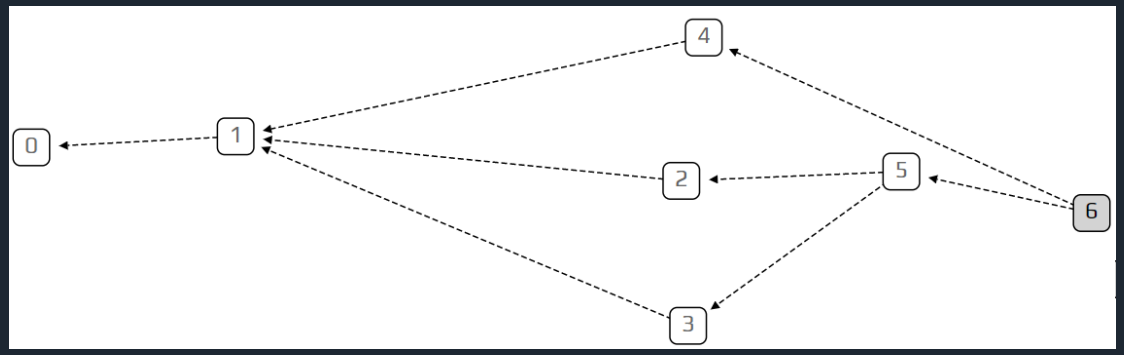
Trinary Numeral System

Tryte	Dec	Char	Tryte	Dec	Char
0, 0, 0	0	9			
1, 0, 0	1	A	-1,-1,-1	-13	N
-1, 1, 0	2	B	0,-1,-1	-12	O
0, 1, 0	3	C	1,-1,-1	-11	P
1, 1, 0	4	D	-1, 0,-1	-10	Q
-1,-1, 1	5	E	0, 0,-1	-9	R
0,-1, 1	6	F	1, 0,-1	-8	S
1,-1, 1	7	G	-1, 1,-1	-7	T
-1, 0, 1	8	H	0, 1,-1	-6	U
0, 0, 1	9	I	1, 1,-1	-5	V
1, 0, 1	10	J	-1,-1, 0	-4	W
-1, 1, 1	11	K	0,-1, 0	-3	Z
0, 1, 1	12	L	1,-1, 0	-2	Y
1, 1, 1	13	M	-1, 0, 0	-1	Z

- IOTA seeds, addresses, hashes etc. are trytes

Directed Acyclic Graph - Tangle

- Directed Graph: Collection of vertices and Edges
- Tangle: Graph which holds transactions. Data Structure behind IOTA
  - Vertex: Transactions
  - Edges: Approvals
  - Tips: Unapproved Transactions
- When new transaction joins Tangle, it chooses 2 previous transactions (tips) to approve
- Strategy of choosing these 2 tips to approve is very important and key to IOTA's new technology



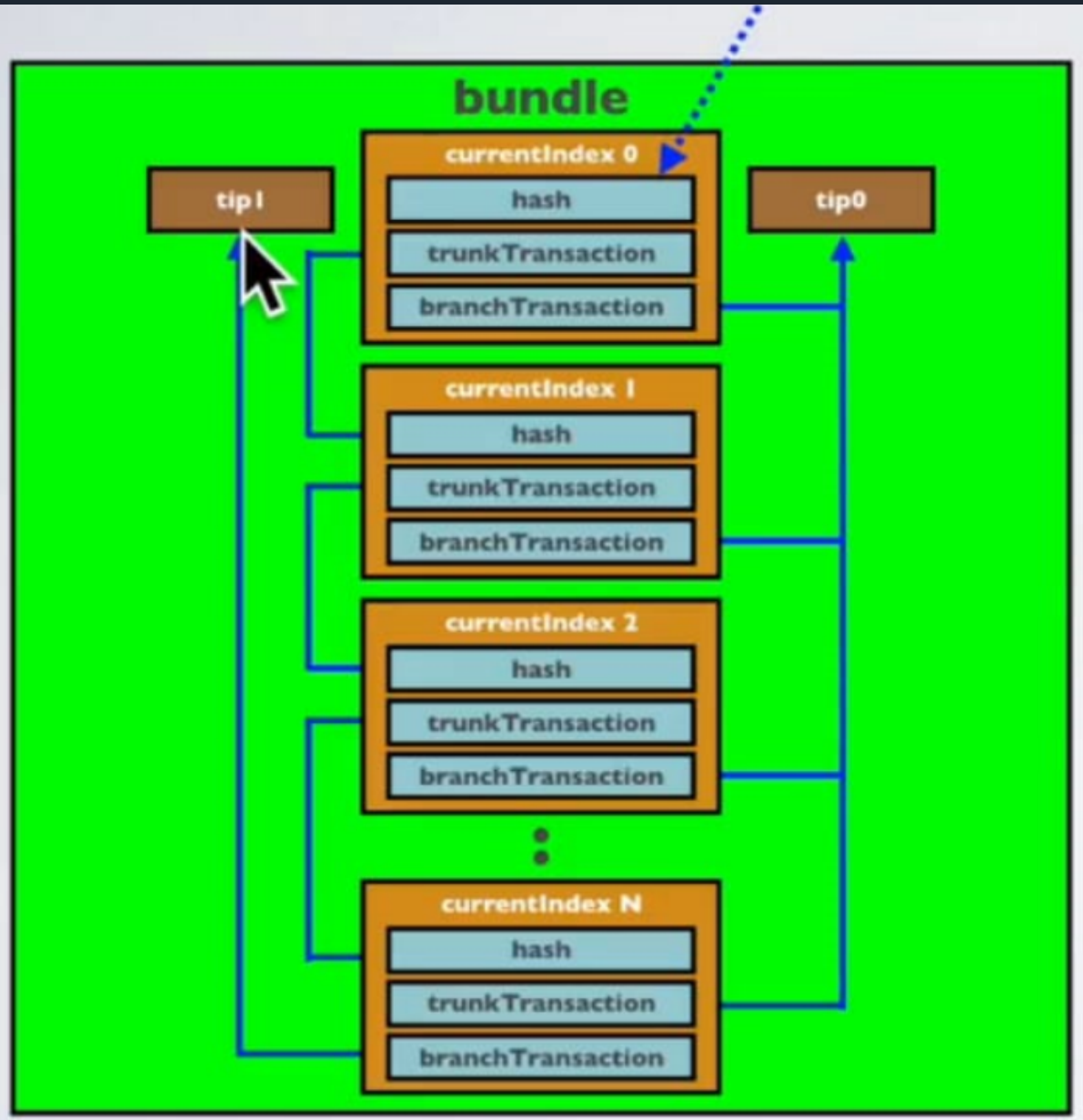
Directed Acyclic Graph

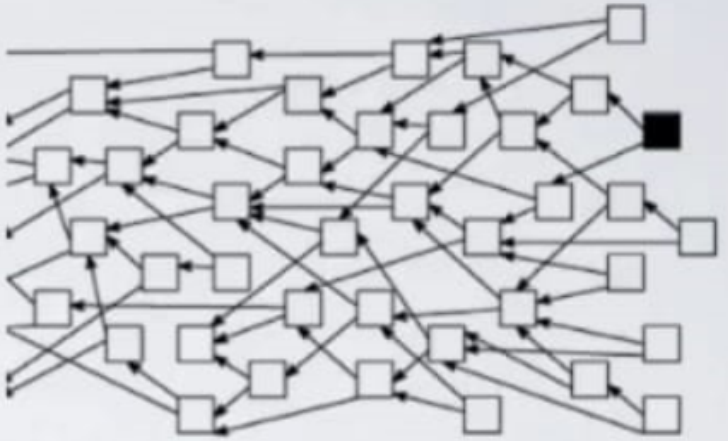
How a Transaction is created?

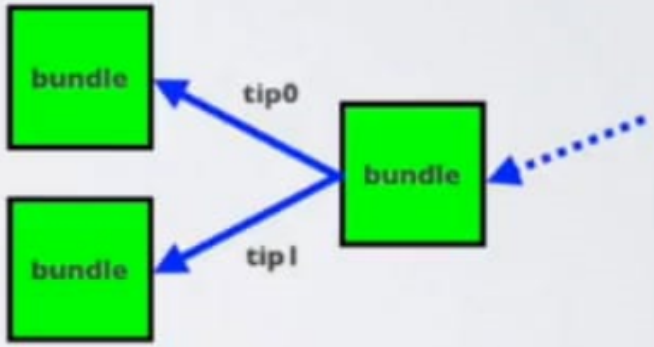
Making a transaction is 3 step process:

- Signing – Your node (Computer/mobile) creates a transaction and sign it with your private key
- Tip Selection – Your node chooses two other unconfirmed transactions (tips) using the Random Walk Monte Carlo (RMWC) algorithm
- Proof of Work – Your node checks if the two transactions are not conflicting. Next, the node must do POW by solving cryptographic puzzle (hashcash) by repeatedly hashing same data until a hash is found with a certain number of leading zero bits.

Transactions in Bundle



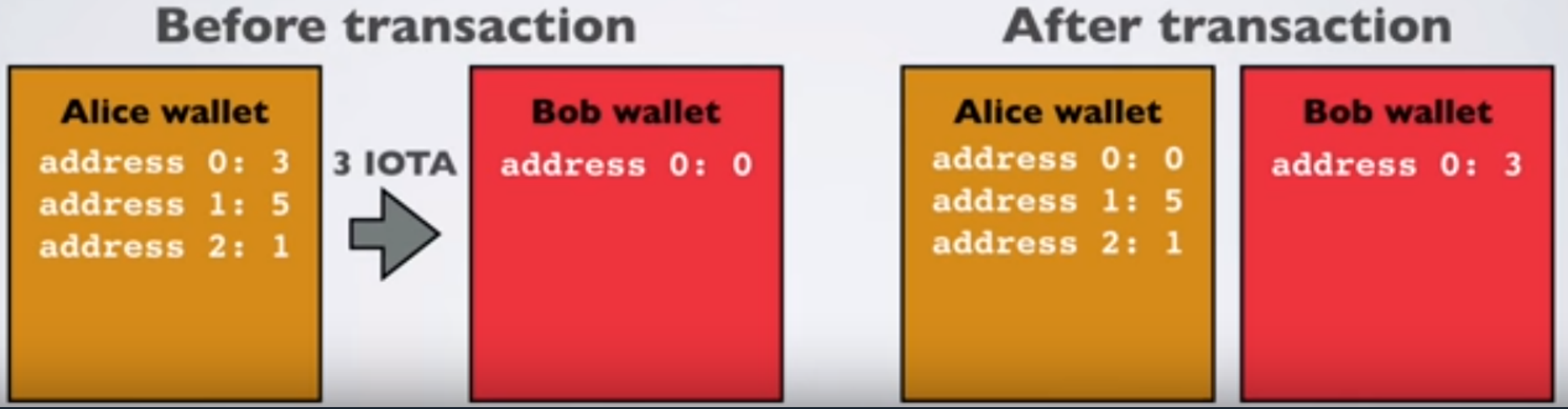




currentIndex 0 = tail transaction
currentIndex N = head transaction

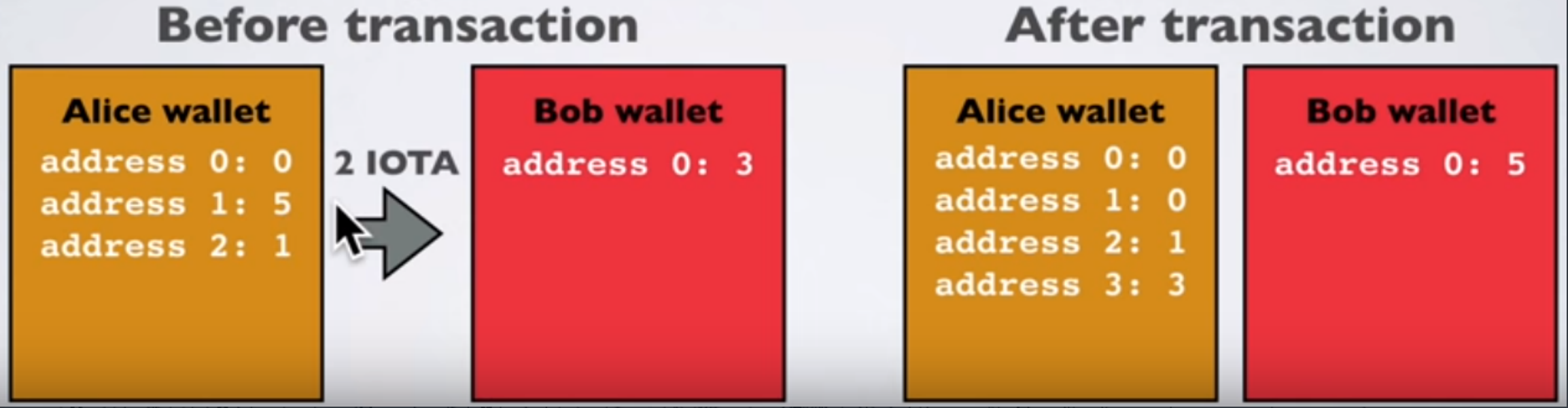
Transaction Example 1

- Alice wants to send 3 IOTAs to Bob. Alice's wallet starts with address 0 and adds the balances of the consecutively addresses until 3 IOTAs are reached or exceeded. Any extra amount over the payment amount will be sent to a new address called the change address, which means you will not have to worry about address reuse in typical cases.



Transaction Example 2

- Alice wants to send 2 IOTA's to Bob. Alice's wallet starts with address 0 and adds the balances of the consecutively addresses until 2 IOTA's are reached or exceeded. Any extra amount over the payment amount will be sent to a new address called the change address, which means you will not have to worry about address reuse in typical cases.



IOTA Transaction Attributes & Types

A Transaction is a unit of multiple attributes (mentioned below). It may be for depositing IOTAs or withdrawal of IOTAs or sending message

Transaction Attributes (2673 trytes)

- ❑ hash
- ❑ signatureMessageFragment
- ❑ address
- ❑ value
- ❑ timestamp
- ❑ currentIndex
- ❑ lastIndex
- ❑ bundle
- ❑ trunkTransaction
- ❑ branchTransaction
- ❑ tag
- ❑ Nonce

Transaction Types

- ❑ Input transaction (withdraw)
- ❑ Output transaction (Deposit / message)
- ❑ Meta Transaction (for signature fragments)

Prepare transactions

- Prepare one or more transaction with senders or receivers addresses and values sign all input transaction with sender's keys using key generator
- Prepare Bundle and generate Bundle Hash
- Assign Bundle Hash to the transactions

Output

Transaction

```
Address   : QQQQQQ......QQQ
Value     : 80
Tag       : VISUALTRANSAC
Timestamp: CurrentTime()

Index     : 0
LastIndex: 3
Bundle    :
Nonce     :

Message   : WELCOME9TO9IOTA
```

Input

Transaction

```
Address   : AAAAAA......AAA
Value     : -100
Tag       : VISUALTRANSAC
Timestamp: CurrentTime()

Index     : 1
LastIndex: 3
Bundle    :
Nonce     :

Message   :
```

Remainder

Transaction

```
Address   : EEEEEE......EEE
Value     : 20
Tag       : VISUALTRANSAC
Timestamp: CurrentTime()

Index     : 3
LastIndex: 3
Bundle    :
Nonce     :

Message   :
```

Transaction Bundle structure

A bundle is an atomic unit of one or more transactions

index	Contents	Transaction type
0	Recipient's address, positive value and sign	output / Deposit
1	Sender's address and the first part of its signature Negative value	Input / withdrawal
2	Sender's address and the rest of its signature	Meta transaction
3	Sender's address and positive value	Output / deposit

Tip Selection Algorithm

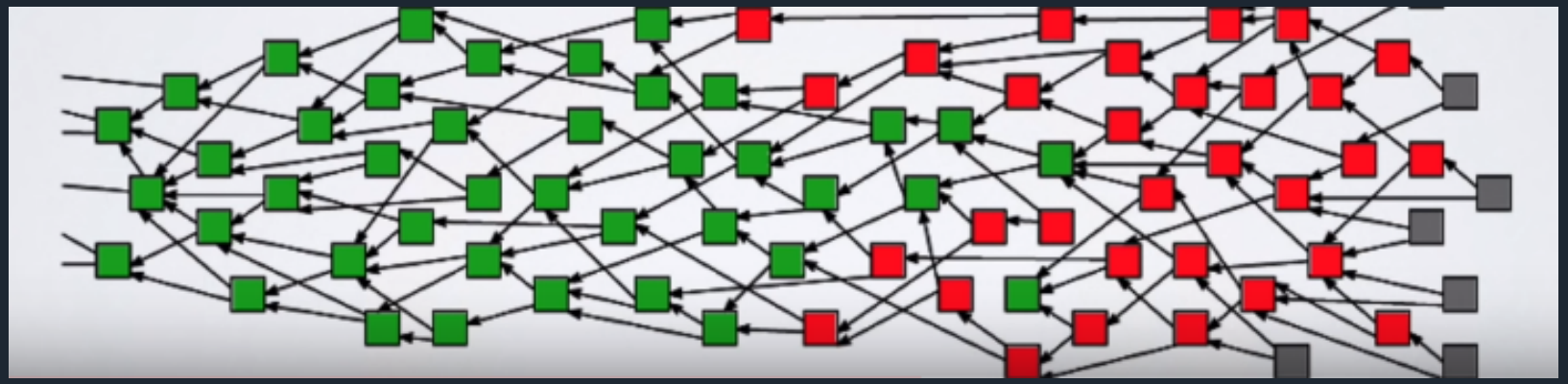
RANDOM WALK MONTE CARLO (RWMC)

- ❑ The goal is to generate fair samples from some difficult distribution
- ❑ RWMC is used in 2 ways:
  - To choose two unconfirmed transactions (tips)
  - And to determine if a transaction is confirmed

Green Blocks: Confirmed

Red Blocks: Still uncertain

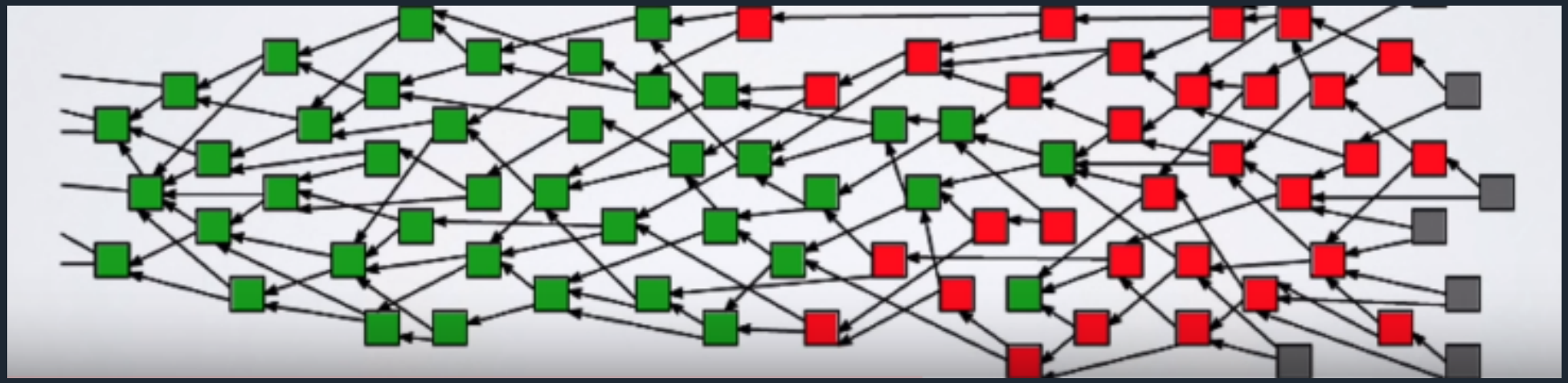
Grey Blocks: Unconfirmed



Tip Selection Algorithm

RANDOM WALK MONTE CARLO (RWMC)

- The goal of every transaction is to be Green. (Grey->Red->Green)
- Identify Depth to start from and execute RMWC N times, probability of your transaction being accepted is therefore M of N. M being the number of times you land on the tip that has a path to transaction
- Transactions with bigger depths takes longer to be validated



Proof Of Work (POW)

- Get the Minimum Weight Magnitude (MWM) – difficulty of POW (9 for test net and 14 for mainnet)
- IOTA transaction data is encoded and stored in a string of 2673 trytes
- Last 81 trytes are reserved for the nonce
- Execute POW using transaction trytes and MWM to generate nonce
- Insert nonce in the transaction obj trytes and convert object into trits
- Execute curl hash with the new transaction trtyes. This returns the transaction hash
- Number of 0's at the end of the CheckHash value must be at least the MWM, then the nonce is valid
- A valid nonce is required for transaction to accepted by the tangle

Transaction Validation

- Each IRI node is responsible for validating transactions to make sure that counterfeit transactions are never confirmed.
- To protect the integrity of the ledger, IRI nodes append only valid transaction to their ledgers.
- IRI nodes validate transactions during the following stages:
  - On receipt of new transactions
  - During the tip selection process

Transaction Validation – On Receipt

When an IRI node receives a new transaction, the transaction validator checks it for the following:

- The proof of work was done.
- The value of any transaction in the bundle doesn't exceed the total global supply.
- The transaction is not older than the last snapshot and not newer than two hours ahead of the node's current time.

Transaction Validation – Tip Selection

Bundle validator

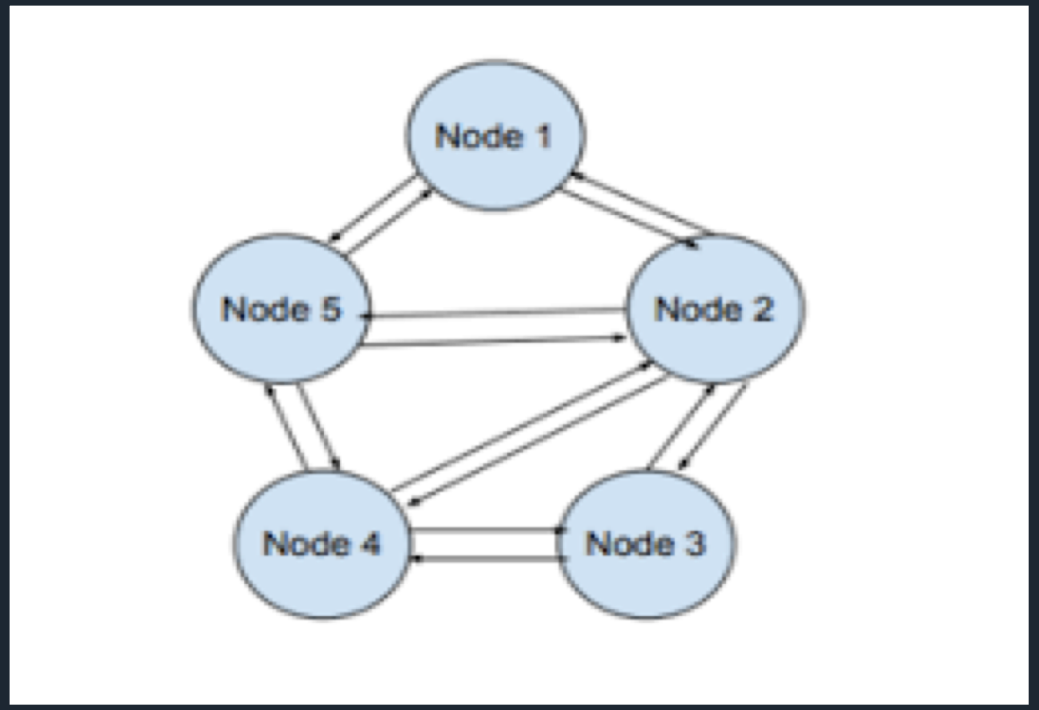
- The value of any transaction in the bundle doesn't exceed the total global supply
- The total value of all transactions in the bundle is 0
- Any signatures are valid in value transactions

Ledger validator

- Checks that each bundle does not lead to a double-spend by checking the values of all addresses in a bundle
- If a double-spend is found, the weighted random walk steps back one transaction and finds another route to a tip transaction

Gossip protocol

- IOTA uses the gossip protocol to propagate messages through the network.
- Any new transaction is immediately broadcast to all neighbors.
- When a node detect that a transaction is missing (i.e. the branch or trunk of a new transaction): the node will ask for the missing txn to it's neighbors.



IOTA Address Usage

- Once you have sent a transaction from an address, never use this address again
- With each transaction part of the private key is revealed
- IOTA uses the Winterwitz one-time signature. This makes it easier for attackers to steal that address's balance with brute force
- You can receive as many transactions you want to an address
- The seed is not compromised if you receive funds at an address that has already been spent from, but the funds at that address is.

Coordinator (COO)

- COO are several full nodes scattered across the world run by IOTA foundation
- It creates zero value transactions called milestones
- Main purpose is to temporary protect the network from large attacks
- Once the amount of organic activity on IOTA ledger is sufficient to where it can evolve unassisted, COO will permanently shut-off

Snapshot

- A snapshot is a method which keeps ledger database small in size
- Groups several transfers to the same address into 1 record, saves only non-zero balances and removes transaction history
- The addresses with balances acts like new genesis addresses
- There are permanodes which stores the entire tangle history and data permanently and securely

IOTA Github

- An IOTA Reference Implementation (IRI), wallet and libraries are available at: https://github.com/iotaledger
- IOTA Reference Implementation is written in Java
- IOTA libraries are available in different programming languages such as javascript, python and Go

Meetups Sponsor

nagarro

THINKING BREAKTHROUGHS

Supporters

GOVERNMENT OF TELANGANA

bradsol

SIMPLIFYING BUSINESS

iDEALABS


WHERE?
when?
which?
WHO?
why?
WHEN?
How?
WHERE?
WHAT?