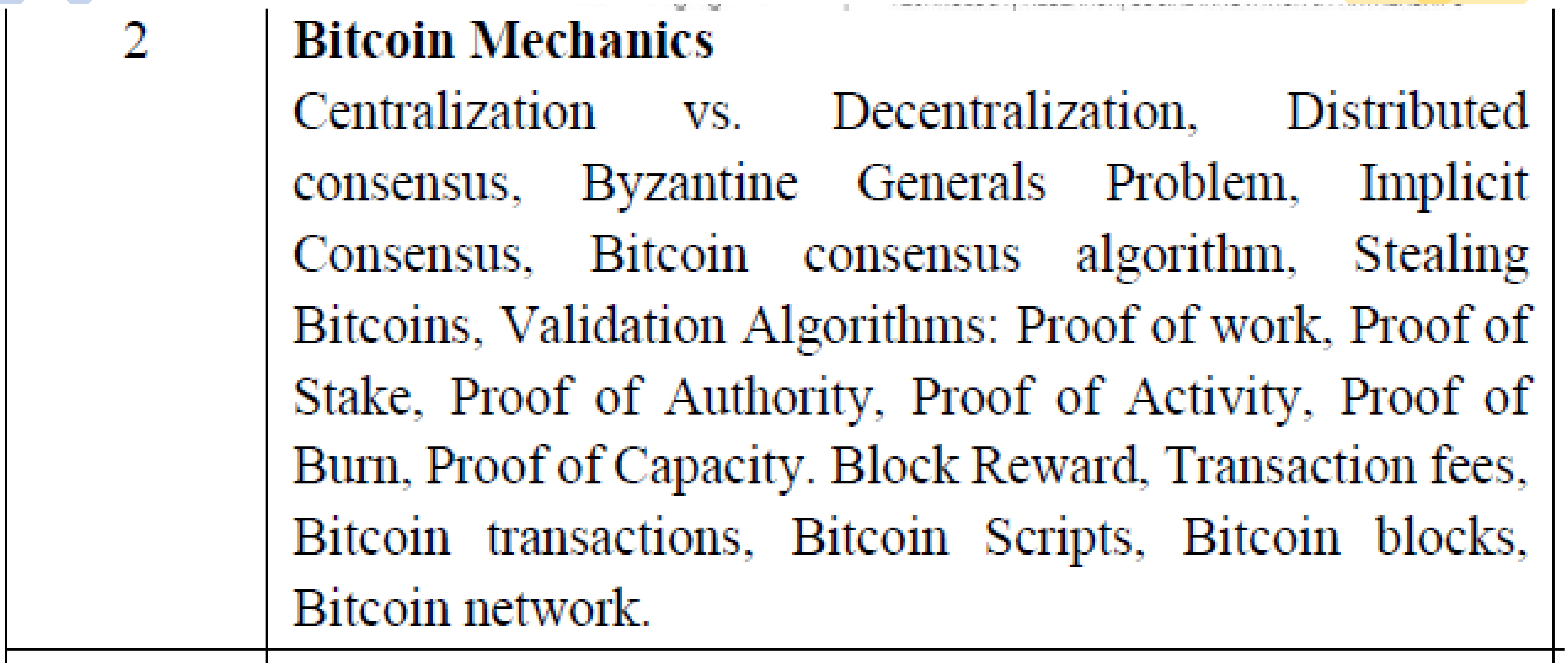
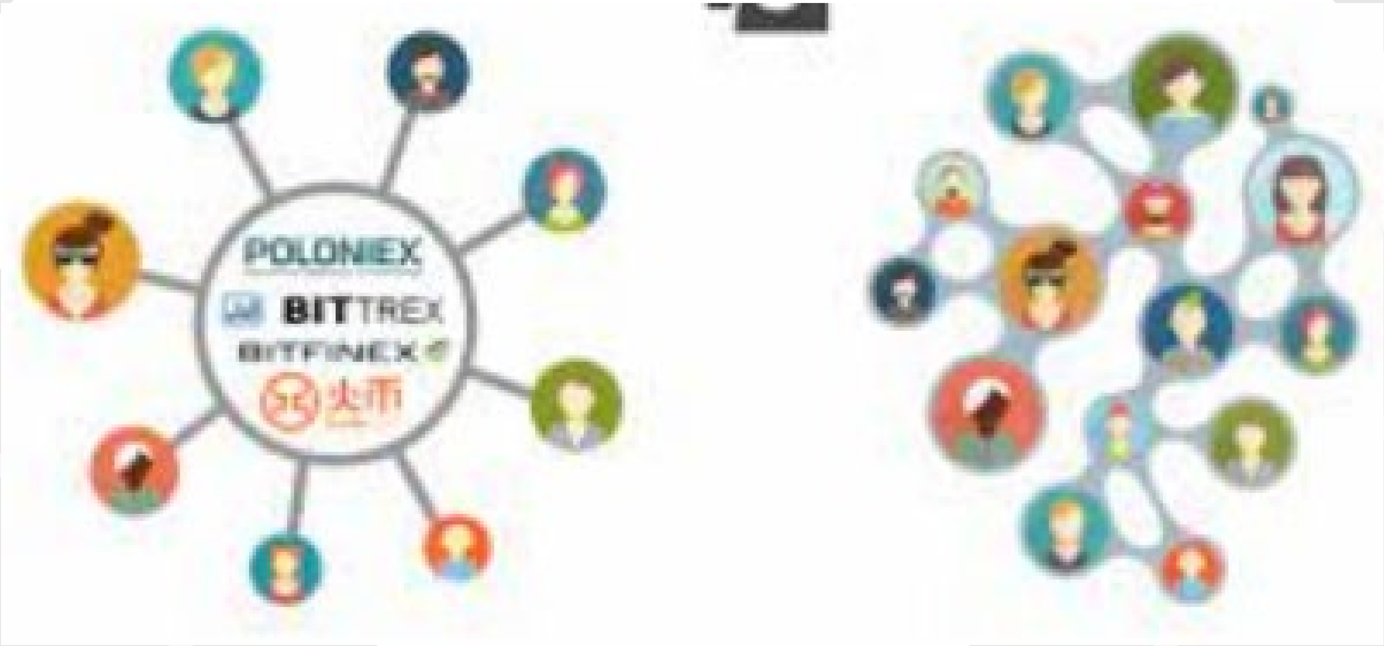


2	**Bitcoin Mechanics** Centralization vs. Decentralization, Distributed consensus, Byzantine Generals Problem, Implicit Consensus, Bitcoin consensus algorithm, Stealing Bitcoins, Validation Algorithms: Proof of work, Proof of Stake, Proof of Authority, Proof of Activity, Proof of Burn, Proof of Capacity. Block Reward, Transaction fees, Bitcoin transactions, Bitcoin Scripts, Bitcoin blocks, Bitcoin network.

Centralization & Decentralization in BitCoin



Centralized vs decentralized cryptocurrency exchanges

- They refer to places where you can buy or sell crypto. Every crypto exchange has its unique rules and regulations, but they all provide you access to the most prevalent cryptocurrencies.
- These exchanges are mainly of two kinds: **Centralized & Decentralized**
- A **centralized cryptocurrency exchange** is a platform where you can buy or sell digital assets. Here, you have to trust a third party to monitor the transaction and secure the assets on behalf of the buyer and the seller.
- Such exchanges require you to submit your personal information for verification.

Centralized vs decentralized cryptocurrency exchanges

- In most cases, centralized crypto exchanges provide their users with flat pairs at stable prices.
- These exchanges are widely popular among cryptocurrency users, and you can easily find one of these platforms online.
- Some examples of centralized cryptocurrency exchanges include Binance, Coinbase, LocalBitcoins, and others.

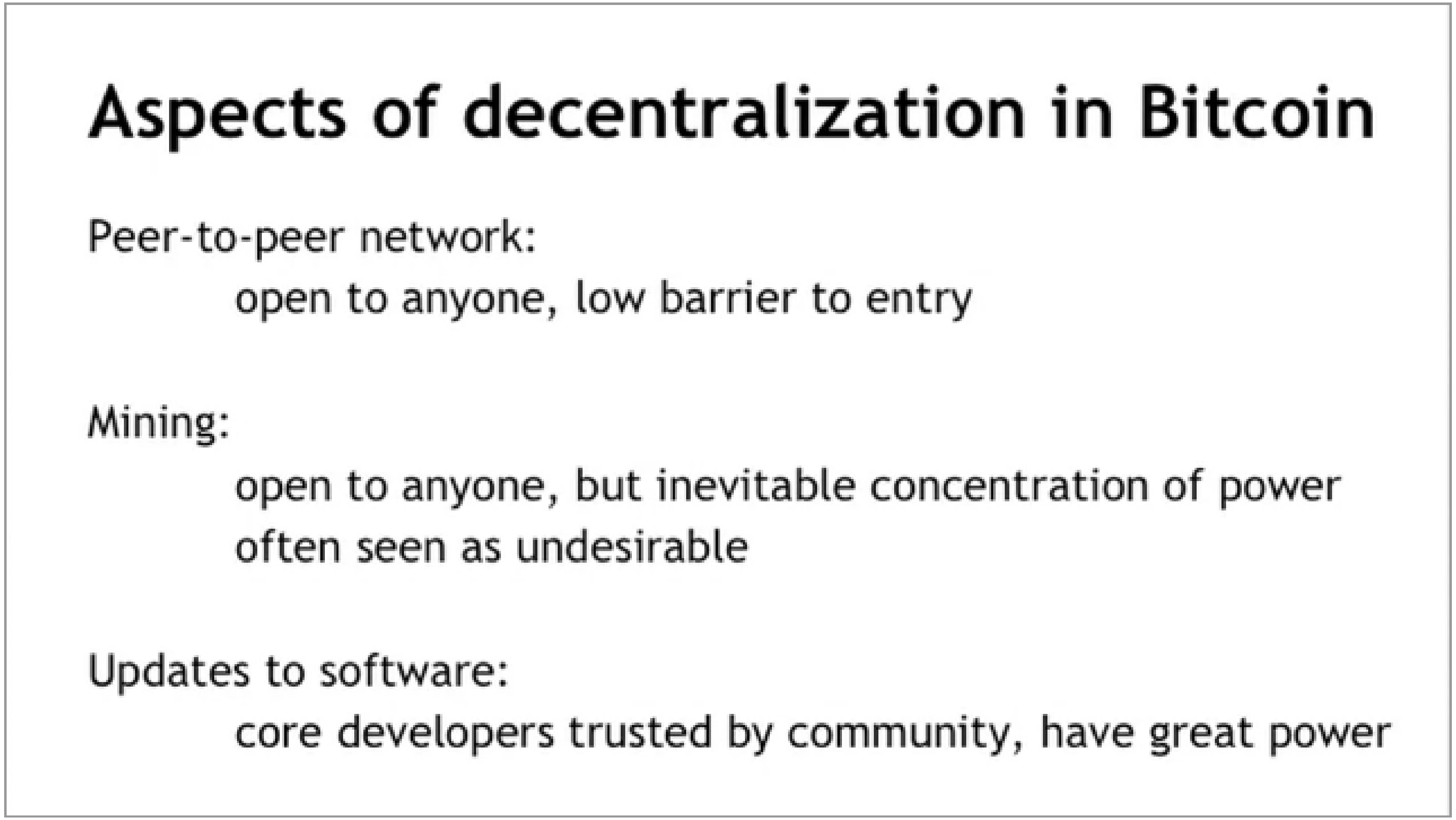
Decentralized cryptocurrency exchanges

- A DEX or a **decentralized cryptocurrency exchange** is similar to a centralized one, except it doesn’t have a third party on which you can rely.
- All of the funds in this exchange remain stored on the blockchain.
- These platforms allow peer-to-peer (P2P) trading for which it uses assets, proxy tokens, or an escrow system, unlike the IOU-based system a centralized crypto exchange uses.
- In a DEx, the client (you) brings his/her cryptocurrency to the gate, which stores the same and gives the client proxy tokens in their place.
- The client can now use these tokens within the blockchain of this exchange. The real cryptocurrency present in the gates collateralizes these tokens.

Decentralized cryptocurrency exchanges

- You can order to sell your current tokens for another kind of tokens in exchange.
- Your order, its matching process, and all the consequent processes remain stored on the blockchain of the exchange, which is the first highlight of these places.
- When you receive any tokens through a transaction, you can convert them into real cryptocurrency as well.

Bitcoin Distributed Aspects



Aspects of decentralization in Bitcoin

Peer-to-peer network:

open to anyone, low barrier to entry

Mining:

open to anyone, but inevitable concentration of power
often seen as undesirable

Updates to software:

core developers trusted by community, have great power

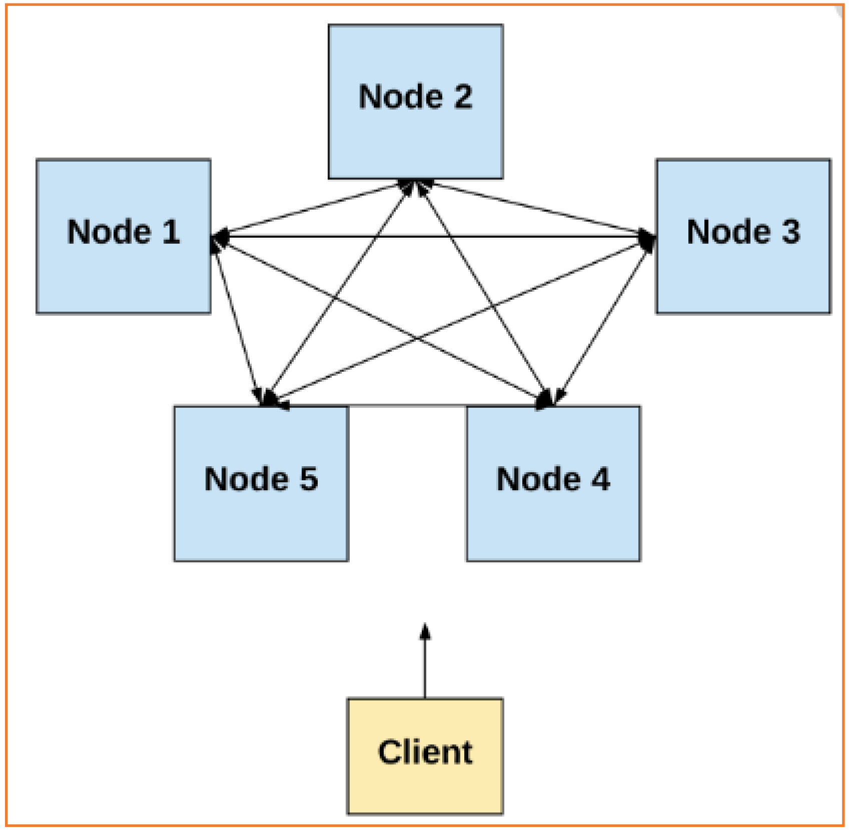
How the Bitcoin protocol achieves decentralization into five more specific questions:

1. Who maintains the ledger of transactions?
2. Who has authority over which transactions are valid?
3. Who creates new bitcoins?
4. Who determines how the rules of the system change?
5. How do bitcoins acquire exchange value?

Centralization and Decentralization in Bitcoin

Parameter	Centralized Crypto Exchange	Decentralized Crypto Exchange
Control	The platform has the most control	User has the most control
Security	Risk of hackers	No chance of hacking or other dangers
Popularity	Highly popular	Not very popular
Fees	Charges fees for using the platform	Charges zero or very minimal fees
Features	Provides a variety of features	Very few features available
Regulation	Easy to regulate, requires license from authorities	Complicated to regulate; doesn’t require a license
Liquidity	High Liquidity	Low Liquidity
Speed	Executes orders in milliseconds	Can take up to a minute to execute orders

Distributed Consensus



A distributed consensus ensures a consensus of data among nodes in a distributed system or reaches an agreement on a proposal.

A consensus algorithm may be defined as the mechanism through which a blockchain network reach consensus.

Public (decentralized) blockchains are built as distributed systems and, since they do not rely on a central authority, the distributed nodes need to agree on the validity of transactions. This is where consensus algorithms come into play.

They assure that the protocol rules are being followed and guarantee that all transactions occur in a trustless way, so the coins are only able to be spent once.

Failures in Consensus

Crash Faults- where a node may suddenly fail during consensus

Network Partitioned Fault Where a node failure creates network partitions thus affecting consensus

Latency that is the asynchronous delay

Byzantines where a node starts behaving maliciously

Byzantine Fault Tolerance

Practical Byzantine Fault Tolerance is a consensus algorithm introduced in the late 90s by Barbara Liskov and Miguel Castro.

pBFT was designed to work efficiently in asynchronous(no upper bound on when the response to the request will be received) systems.

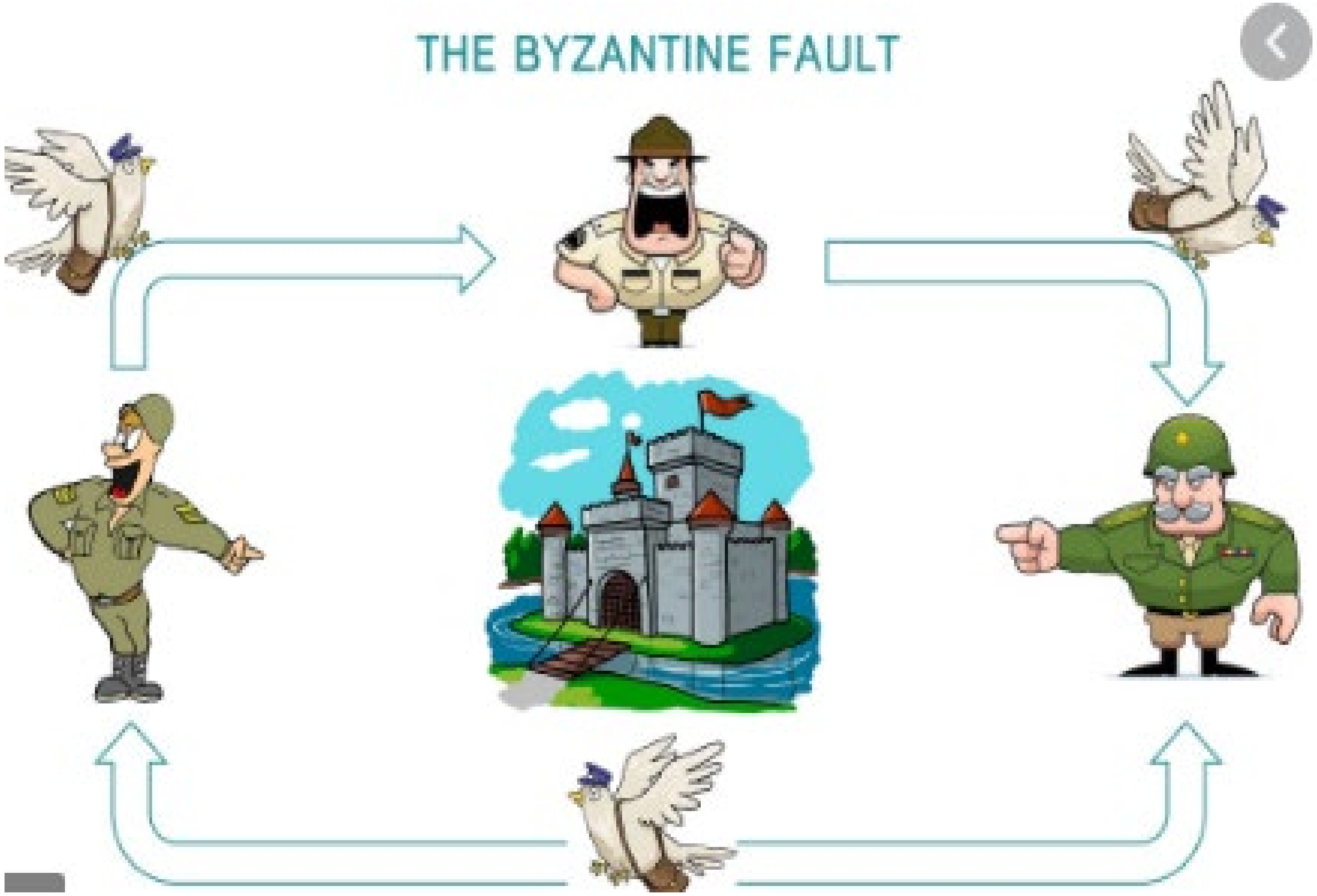
It is optimized for low overhead time.

Byzantine Fault Tolerance solutions. Application areas include distributed computing and blockchain..

- **Byzantine** Fault Tolerance(BFT) is the feature of a distributed network to reach **consensus**(agreement on the same value) even when some of the nodes in the network fail to respond or respond with incorrect information
- This means that a BFT system is able to continue operating even if some of the nodes fail or act maliciously. There is more than one possible solution to the **Byzantine** Generals' Problem and, therefore, multiple ways of building a BFT system.

Byzantine Generals Problem

- Imagine divisions of a Byzantine army, attacking a completely encircled city.
- To proceed, the generals of each division, who are dispersed around the city's periphery, must agree on a battle plan.
- However, while some generals want to attack, others may want to retreat.

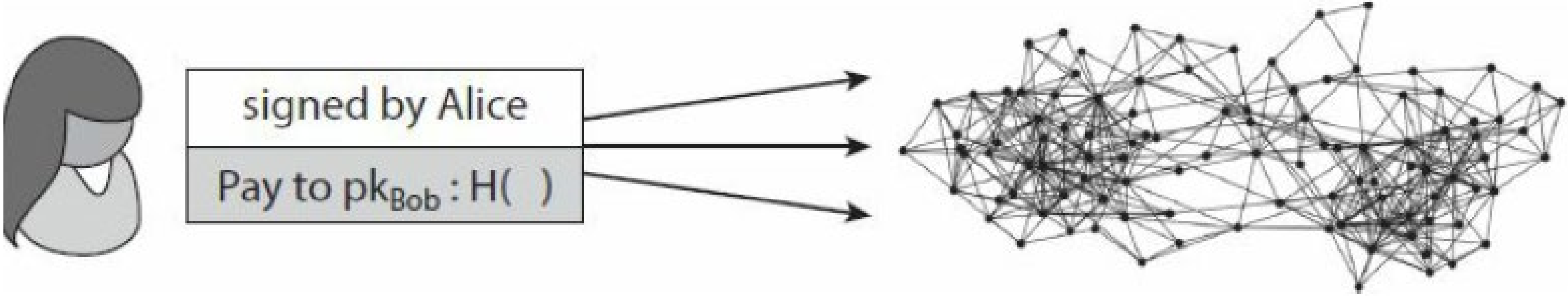


Byzantine Generals Solution Strategy

The solution to the problem relies on an algorithm that can guarantee that:

1. **All loyal lieutenant generals decide upon the same plan of action, and**
2. **A small number of traitors cannot cause the loyal lieutenants to adopt a bad plan.**

- **The loyal lieutenants will all do what the algorithm says they should, but the traitors may do anything they wish.**
- **The algorithm must guarantee the first condition regardless of what the traitors do. The loyal lieutenants should not only reach an agreement but should agree upon a reasonable plan.**



Distributed consensus protocol.

- There are *n* nodes that each have an input value.
- Some of these nodes are faulty or malicious.

<u>A Distributed consensus protocol has the following two properties:</u>

- It must terminate with all honest nodes in agreement on the value.
- The value must have been generated by an honest node.

Consensus in bitcoin Blocks I

- ❑ Given that a variety of users are broadcasting these transactions to the network, the nodes must agree on exactly which transactions were broadcast and the order in which these transactions occurred.
- ❑ So at any given point, all nodes in the peer-to-peer network have a ledger consisting of a sequence of blocks, each containing a list of transactions that they have reached consensus on.
- ❑ Additionally, each node has a pool of outstanding transactions that it has heard about but that have not yet been included in the block chain.
- ❑ For these transactions, consensus has not yet happened, and so by definition, each node might have a slightly different version of the outstanding transaction pool.

Consensus in bitcoin Blocks - II

- ❑ **How exactly do nodes come to consensus on a block?**
- ❑ **One way to do this is as follows. At regular intervals (e.g., every 10 minutes), every node in the**

  system proposes its own outstanding transaction pool to be included in the next block.
- ❑ **Then the nodes execute some consensus protocol, where each node's input is its own proposed block.**
- ❑ **Now, some nodes may be malicious and put invalid transactions into their blocks, but we can assume that other nodes are honest.**

Consensus in bitcoin Blocks - III

- **If the consensus protocol succeeds, a valid block will be selected as the output.**
- **Even if the selected block was proposed by only one node, it’s a valid output as long as the block is valid.**
- **Now there may be some valid outstanding transaction that did not get included in the block,**

  but this is not a problem.
- **If some transaction somehow didn’t make it into this particular block, it could just wait and get into the next block.**

Consensus in Bitcoin, Can be hard !

At any given time:

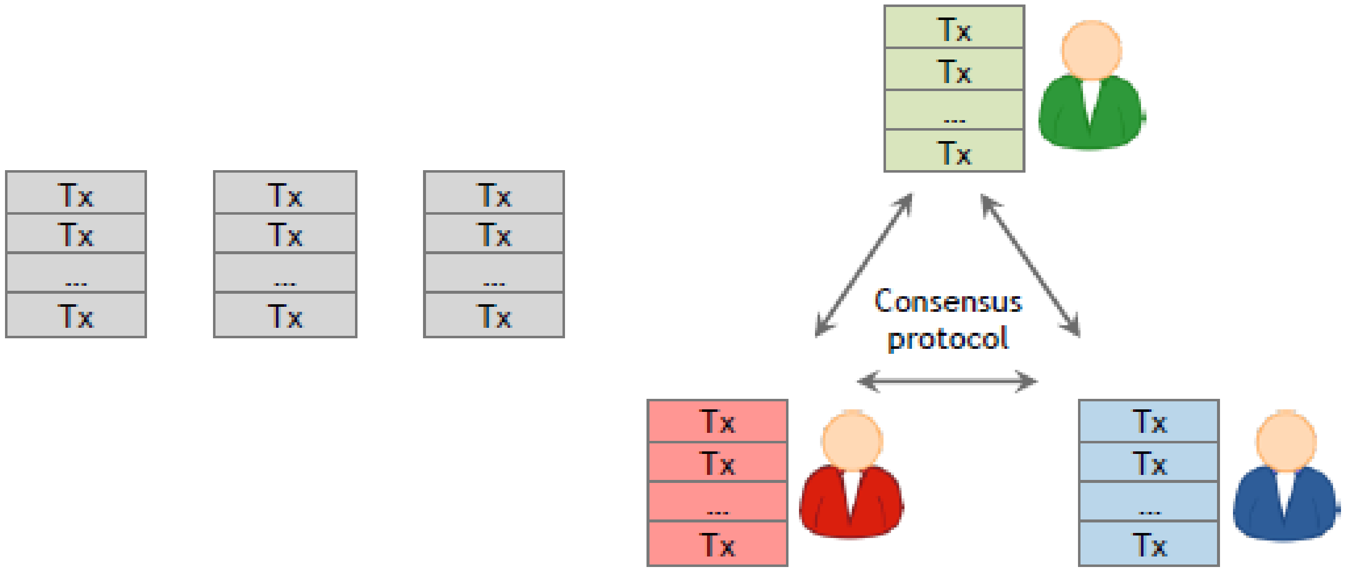
- ❑ All nodes have a sequence of blocks of transactions they have reached consensus on
- ❑ Each node has a set of outstanding transactions it has heard about

❖Nodes may crash,   Nodes may be malicious (Byzantine behaviour)

❖Network is imperfect

Not all pairs of nodes connected , Faults in network , Latency; no global time

How Consensus could work in Bitcoin



OK to select any valid block, even if proposed by only one node

Bitcoin Consensus: Theory & Practice

Bitcoin consensus works better in practice than in theory.

Theory is still catching up.

BUT theory is important, can help predict unforeseen attacks.

Things Bitcoin does differently

Introduces incentives

- Possible only because it's a currency!

Embraces randomness

- Does away with the notion of a specific end-point
- Consensus happens over long time scales — about 1 hour

Consensus without Identities

Why **identity**?

- Pragmatic: some protocols need node IDs

- Security: assume less than 50% malicious

Why don’t Bitcoin nodes have identities?

- Identities are hard in P2P systems – Sybil attacks

- Pseudonymity is a goal of Bitcoin

Key idea: implicit consensus

In each round, random node is picked

This node proposes the next block in the chain

Other nodes implicitly accept/reject this block

- by either extending it
- or ignoring it and extending chain from earlier block

Every block contains hash of the block it extends

Consensus Algorithm (simplified)

1. New transactions are broadcast to all nodes

2. Each node collects new transactions into a block

3. In each round a random node gets to broadcast its block

4. Other nodes accept the block only if all transactions in it are valid (unspent, valid signatures)

5. Nodes express their acceptance of the block by including its hash in the next block they create

What can a Malicious Node do?

Stealing Bitcoins:

– Stealing another user’s coins would require to forge the owner’s signature

Denial-of-Service:

– Alice wants to prevent Bob’s transactions from being included in block chain.

– Alice may prevent for one or more rounds.

– Eventually, honest node will be picked, who will include Bob’s transaction in proposed block.

Double-Spend Attack:

– Alice purchases service from Bob and pays in coins.

– Alice creates transaction and broadcasts it to the network.

– Later, Alice attempts to pay same coin to one of her accounts.

Stealing Bitcoins

- Can Alice simply steal bitcoins belonging to another user at an address she
- doesn’t control?
- No. Even if it is Alice’s turn to propose the next block in the chain, she cannot steal other users’ bitcoins.
- Doing so would require Alice to create a valid transaction that spends that coin.
- This would require Alice to forge the owners’ signatures, which she cannot do if a secure digital signature scheme is used.
- So as long as the underlying cryptography is solid, she’s not able to simply steal bitcoins.

Denial of Service Attack

- Let’s consider another attack. Suppose that Alice really dislikes some other user Bob.
- Alice can then decide that she will not include any transactions originating from Bob’s address in any block that she proposes to put in the block chain.
- In other words, she’s denying service to Bob. Even though this is a valid attack that Alice can try to mount, luckily it’s nothing more than a minor annoyance.
- If Bob’s transaction doesn’t make it into the next block that Alice proposes, he will just wait until an honest node has the chance to propose a block, and then his transaction will get into that block. So that’s not really a good attack either.

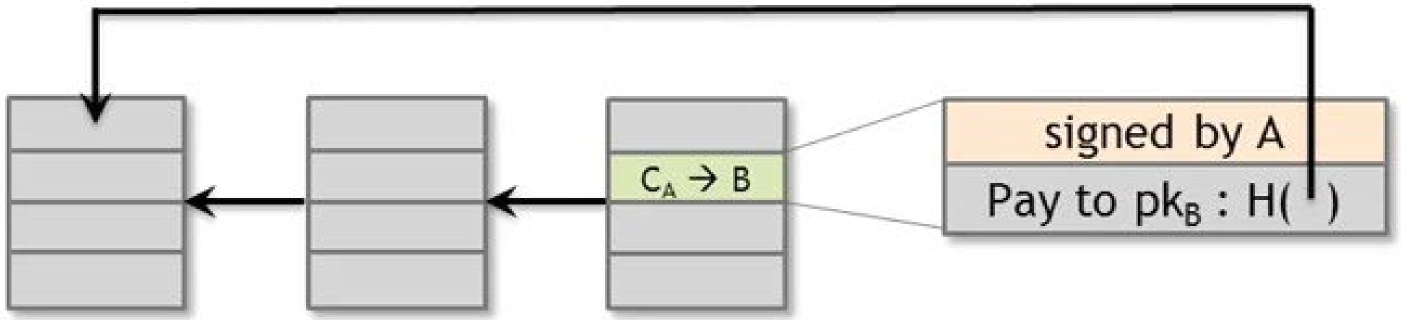
Double Spending attack

Alice may try to launch a double-spend attack. To understand how that works, let’s assume that Alice is a customer of some online merchant or website run by Bob, who provides some online service in exchange for payment in bitcoins. Let’s say Bob’s service allows the download of some software. So here’s how a double-spend attack might work. Alice adds an item to her shopping cart on Bob’s website, and the server requests payment. Then Alice creates a Bitcoin transaction from her address to Bob’s and broadcasts it to the network. Let’s say that some honest node creates the next block, and includes this transaction in that block. So there is now a block that was created by an honest node that contains a transaction that represents a payment from Alice to the merchant Bob.

Double Spending attack

Let’s return to how Alice can launch a double-spend attack. The latest block was generated by an honest node and includes a transaction in which Alice pays Bob for the software download. On seeing this transaction included in the block chain, Bob concludes that Alice has paid him and allows Alice to download the software. Suppose the next random node that is selected in the next round happens to be controlled by Alice. Since Alice gets to propose the next block, she could propose one that ignores the block that contains the payment to Bob and instead contains a pointer to the previous block. Furthermore, in the block that she proposes, Alice includes a transaction that transfers the very coins that she was sending to Bob to a different address that she herself controls. This is a classic double-spend pattern.

What can a malicious node do?

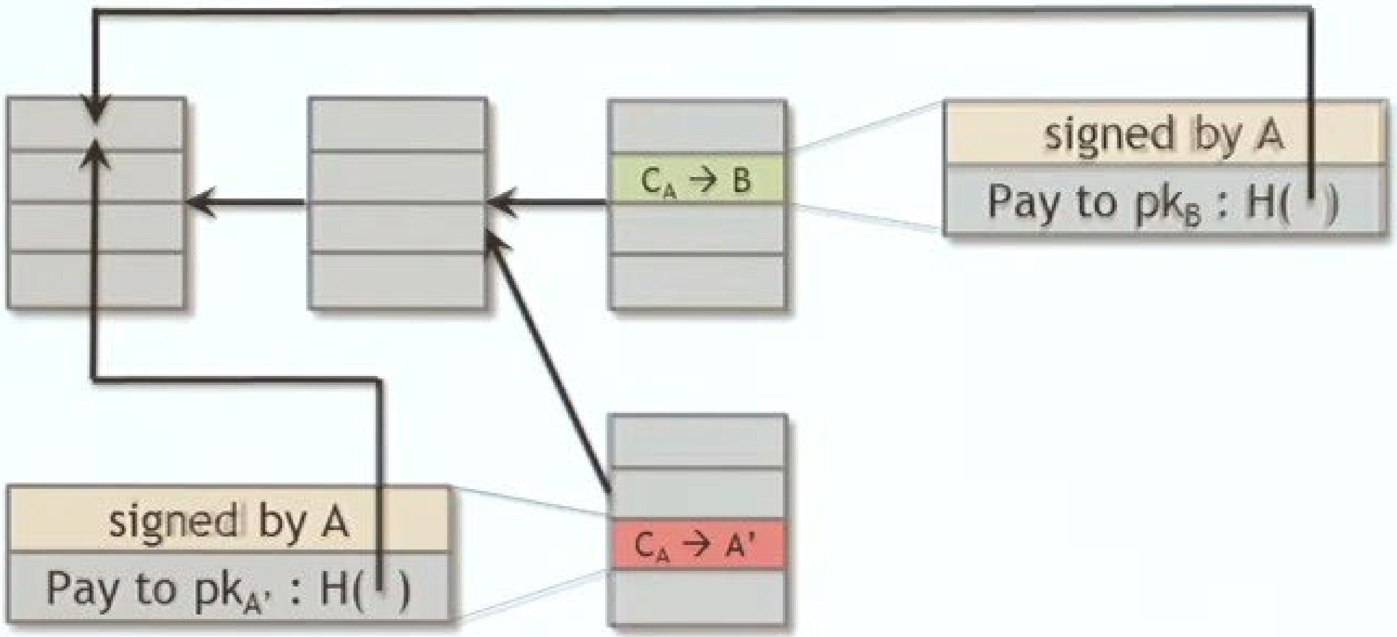


Double Spending attack

Alice creates two transactions: one in which she sends Bob bitcoins, and a second in which she double spends those bitcoins by sending them to a different address, which she controls. As they spend the same bitcoins, only one of these transactions can be included in the block chain. The arrows between blocks are pointers from one block to the previous block that it extends by including a hash of that previous block within its own contents. CA is used to denote a coin owned by Alice.

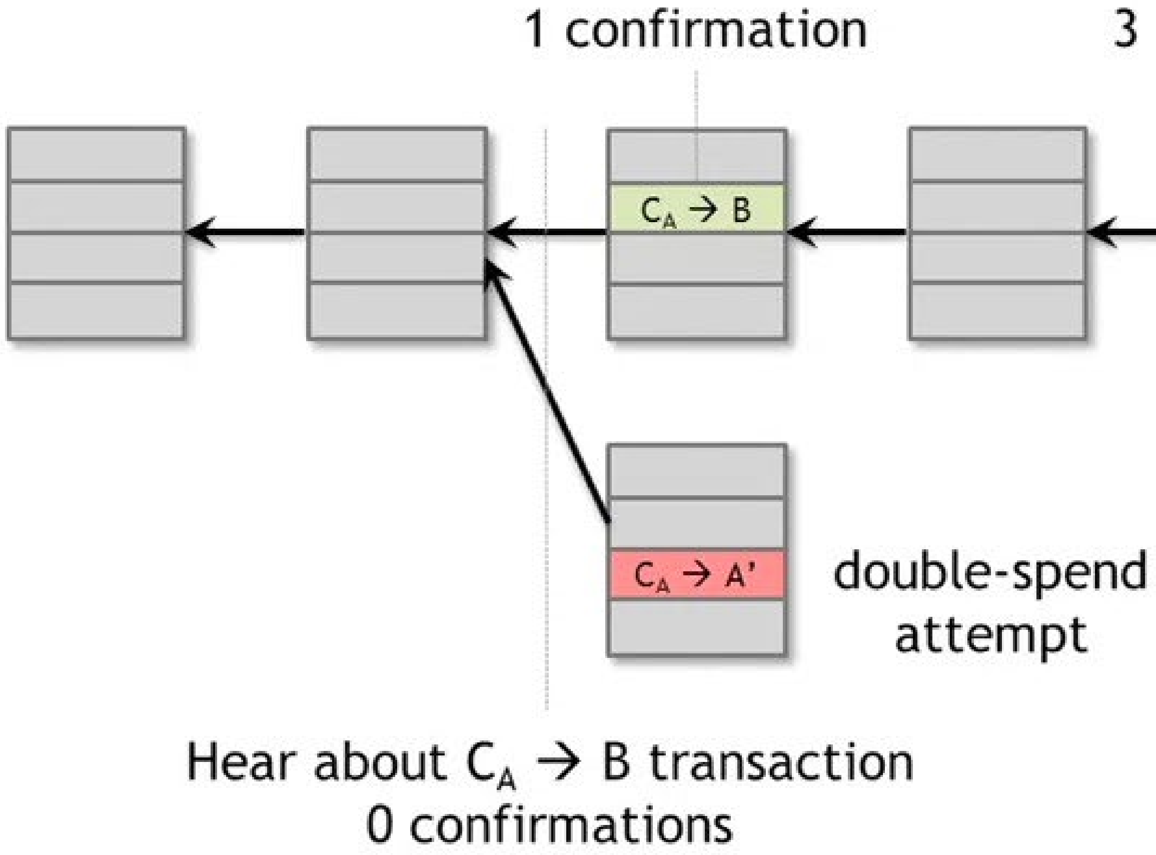
How do we know whether this double-spend attempt is going to succeed or not? Well, that depends on which block will ultimately end up on the long-term consensus chain—the one with the Alice → Bob transaction or the one with the Alice → Alice transaction. What determines which block will be included? Honest nodes follow the policy of extending the longest valid branch, so which branch will they extend? There is no right answer!

What can a malicious node do?



Double-spending attack

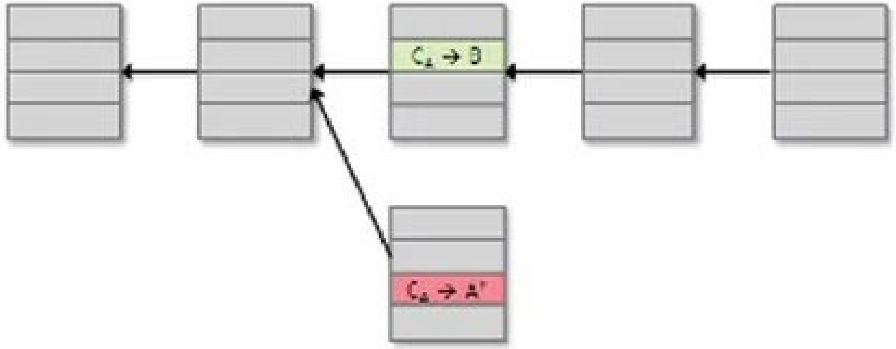
From Bob the merchant’s point of view



Double-spend probability <u>decreases exponentially</u> with # of confirmations

Most common heuristic:
6 confirmations

Recap



Protection against invalid transactions is cryptographic, but enforced by consensus

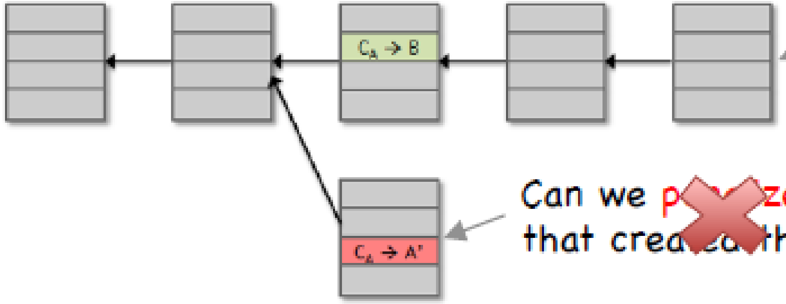
Protection against double-spending is purely by consensus

You're never 100% sure a transaction is in consensus branch. Guarantee is probabilistic

Block Validation Algorithms

Assumption of Honesty is problematic

Q: Can we give nodes incentives for behaving honestly?



Can we reward nodes that created these blocks?

Can we penalize the node that created this block?

Everything so far is just a distributed consensus protocol.

But now we utilize the fact that the currency has value.

Two Types of Incentives

Incentive Type 1: **Block Reward**

Incentive Type 2: **Transaction Fees**

Incentive 1: Block Reward

Creator of block gets to

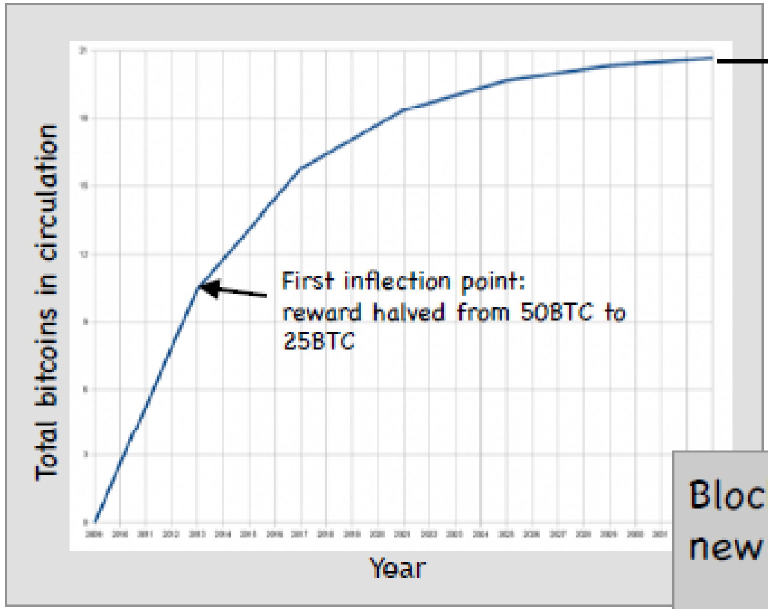
1. include special coin-creation transaction in the block
2. choose recipient address of this transaction (typically creator)

Value is fixed: currently 25 BTC, halves every 4 years

The Catch:
Block creator gets to "collect" the reward only if the block ends up on long-term consensus branch!

Note: This is the only way to create new Bitcoins!

There is a finite Supply of Bitcoins



First inflection point: reward halved from 50BTC to 25BTC

Total supply: 21 million

Block reward is how new bitcoins are created.

Runs out in 2040. No new bitcoins unless rules change.

Incentive 2: Transaction Fees

Creator of transaction can choose to make output value less than input value.

Remainder is a transaction fee and goes to block creator.

Purely voluntary, like a tip.

Transaction fees become increasingly important, as block rewards start running out.

It is a bit unclear how this all will work out. Ongoing research!

Three Remaining Problems

1. How to pick a random node?
2. How to avoid a free-for-all due to rewards?
3. How to prevent Sybil attacks?

Selecting a Random Node: Proof of Work

To approximate selecting a random node:

Select nodes in proportion to a resource
that no one can monopolize (we hope)

- In proportion to computing power: **proof-of-work**

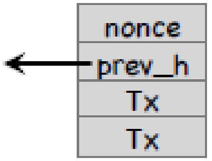
- In proportion to ownership: **proof-of-stake**

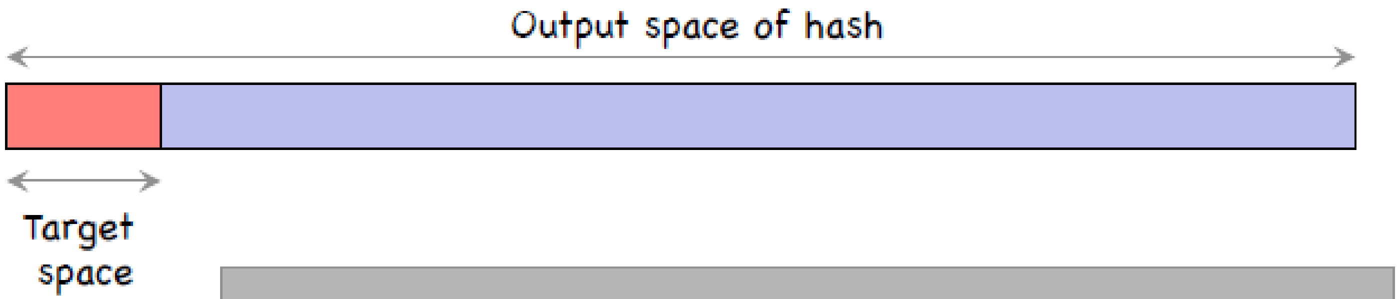
Proof-of-Work: Hash Puzzles

To create block, find **nonce** such that

H(nonce || prev_hash || tx || ... || tx)

is very small.





If hash function is secure:

only way to succeed is to try enough nonces until you get lucky

The 3 necessary Properties of Proof-of-Work

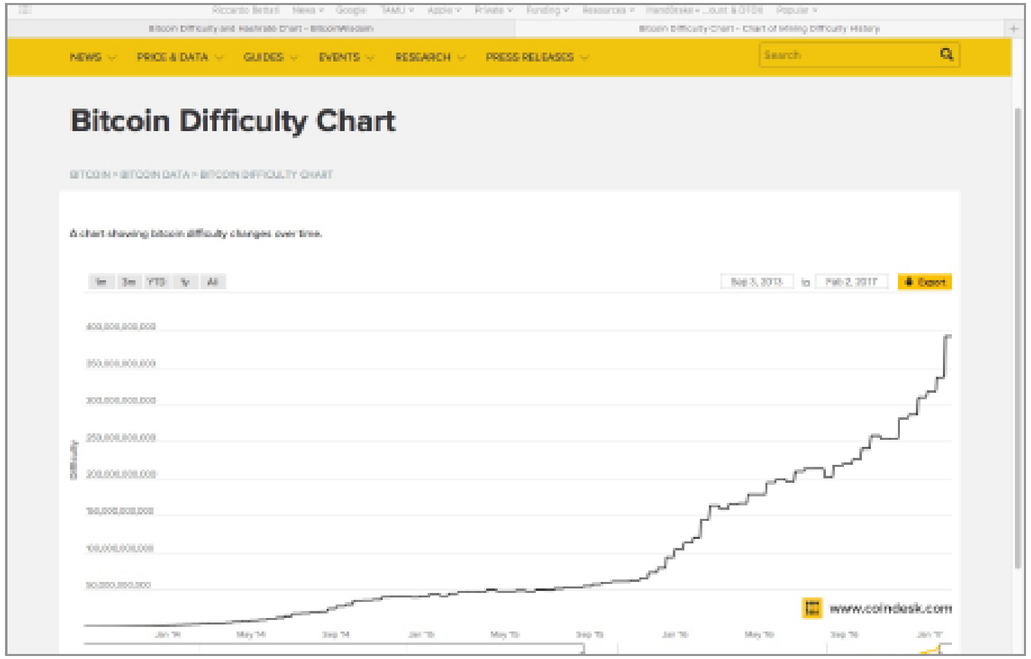
Property 1: Must be (moderately) difficult to compute

Property 2: The Cost must be "parameterizable"

Property 3: Must be trivial to verify

Property 1: Difficult to compute

It takes about *2^32 * Difficulty* to find a block.



Bitcoin Difficulty Chart

BITCOIN > BITCOIN DATA > BITCOIN DIFFICULTY CHART

A chart showing bitcoin difficulty changes over time.

Only some nodes bother to compete: Miners

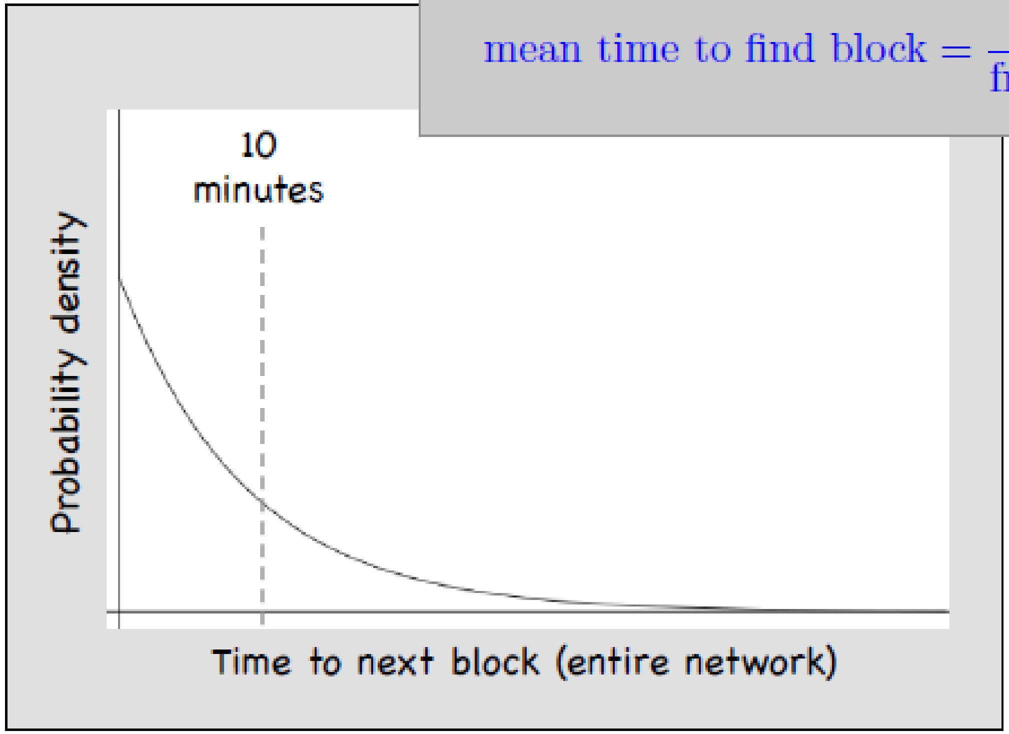
Property 2: Parameterizable Cost

Nodes automatically re-calculate the target every 2016 blocks (about every two weeks).

Goal: average time between blocks = 10 minutes

Adjust difficulty to meet 10-minute goal.

When will I get my Bitcoins?



For individual miner:

$$\text{mean time to find block} = \frac{\text{10 minutes}}{\text{fraction of hash power}}$$

Property 3: Trivial to Verify

Nonce is published as part of block.

Other miners simply verify that

$$H(nonce \parallel prev\_hash \parallel tx \parallel \ldots \parallel tx) < target$$

Economics of Mining

If

mining reward > mining cost

then miner makes a **profit**

where

mining reward = block reward + tx fees

mining cost = hardware cost + operating costs (electricity, cooling, etc.)

Complications:

- fixed vs. variable costs
- reward depends on global hash rate
- Cost in US$ vs. reward in Bitcoins
- Being an honest miner is not provably optimal!

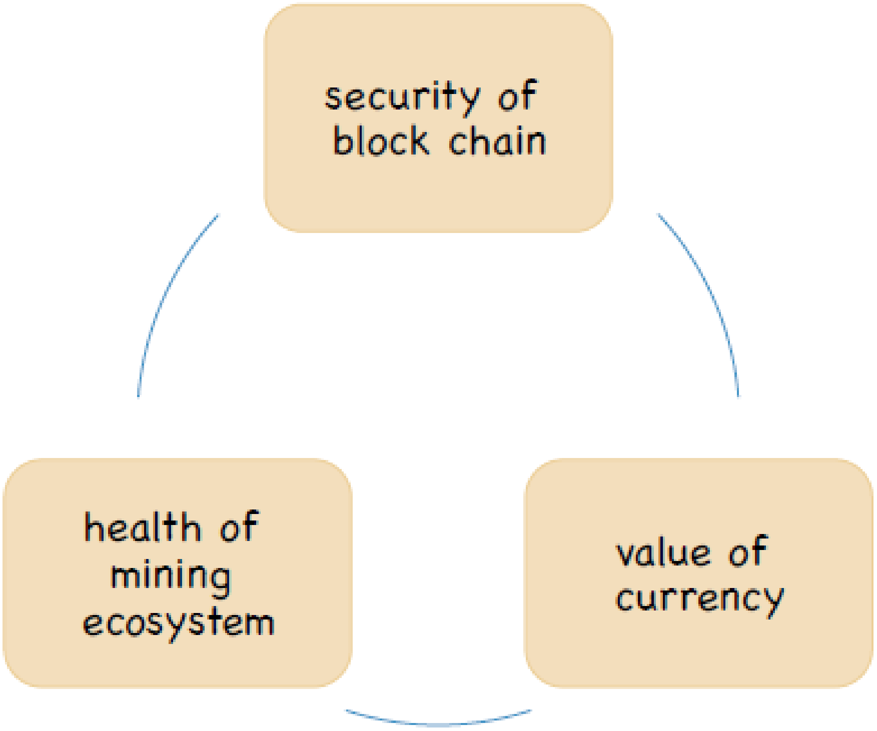
We need Three Types of Consensus

1. Consensus on **Value**

2. Consensus on **State**

3. Consensus on **Rules**

Bootstrapping a Cryptocurrency



What about the "51% Attacker" Scenario?!

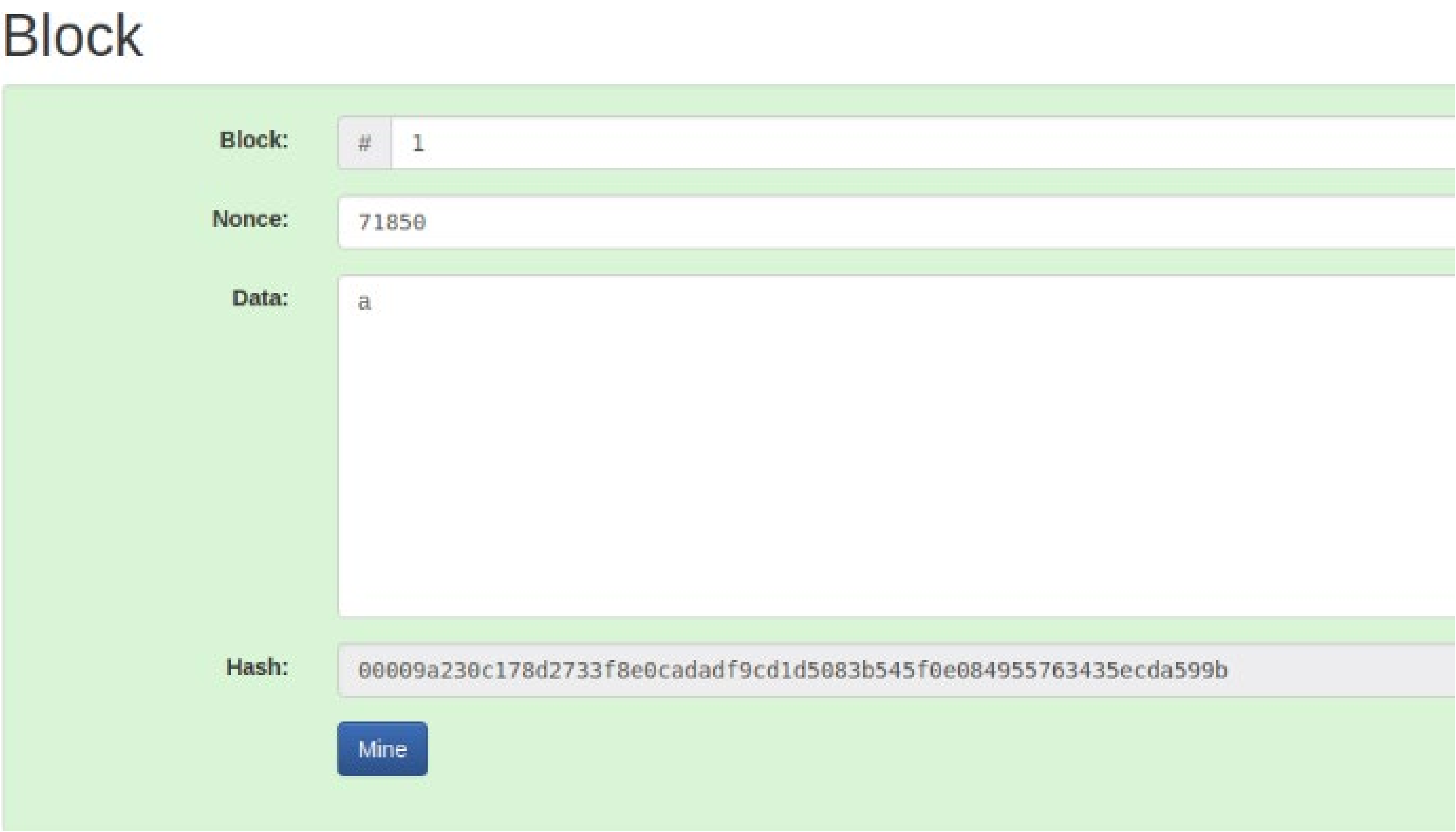
Steal coins from existing address?	✗
Suppress some transactions?	
• From the block chain	✓
• From the P2P network	✗
Change the block reward?	✗
Destroy confidence in Bitcoin?	✓✓

Proof of Work

- Proof of Work(PoW) is the **original consensus algorithm** in a blockchain network.
- The algorithm is used to confirm the transaction and creates a new block to the chain.
- In this algorithm, **minors** (a group of people) compete against each other to complete the transaction on the network.
- The process of competing against each other is called **mining**. As soon as miners successfully created a valid block, he gets **rewarded**.
- The most famous application of Proof of Work(PoW) is Bitcoin.

Proof of Work

- Producing proof of work can be a random process with low probability.
- In this, a lot of **trial and error** is required before a valid proof of work is generated.
- The main working principle of proof of work is a mathematical puzzle which can easily prove the solution.
- Proof of work can be implemented in a blockchain by the Hashcash proof of work system.
- In the below image, you can see that this block is composed of a block number, data field, cryptographic hash associated with it and a nonce. The nonce is responsible for making the block valid.



Block

Block: # 1

Nonce: 71850

Data: a

Hash: 00009a230c178d2733f8e0cadadf9cd1d5083b545f0e084955763435ecda599b

Proof of Work

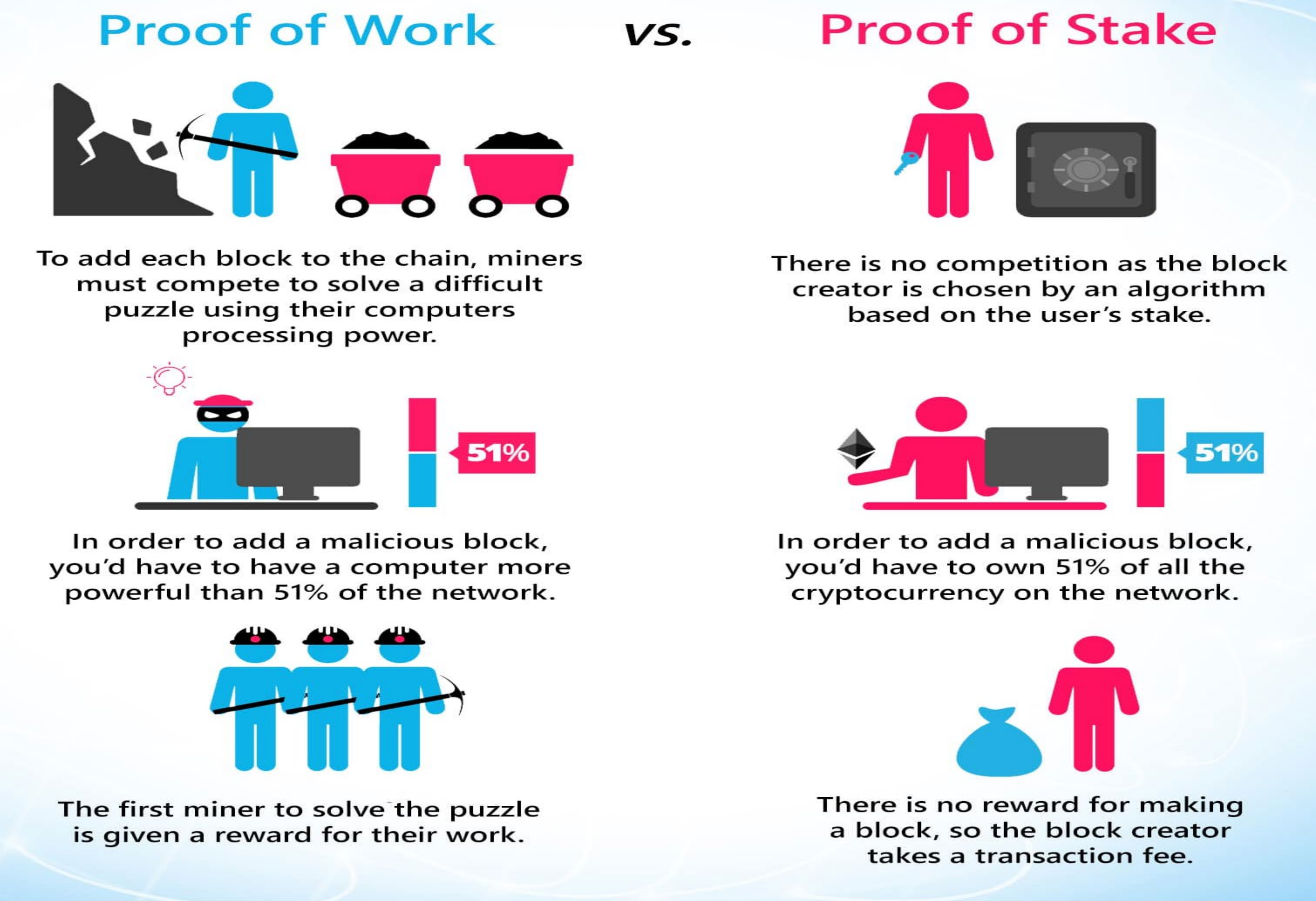
- In the puzzle game, bitcoin software creates a challenge, and there is a game begins.
- This game involves all miners competing against each other to solve the challenges, and this challenge will take approximately 10 minutes to be completed.
- Every single miner starts trying to find the solution to that one Nonce that will satisfy the hash for the block.
- At some specific point, one of those miners in the global community with higher speed and great hardware specs will solve the cryptography challenge and be the winner of the game.
- Now, the rest of the community will start verifying that block which is mined by the winner.
- If the nonce is correct, it will end up with the new block that will be added to the blockchain. The concept of generating a block provides a clear explanation of proof of work(PoW).

Proof of Stake

- The Proof of Stake (PoS) concept states that a person can mine or validate block transactions according to how many coins they hold. This means that the more coins owned by a miner, the more mining power they have.
- Ethereum is moving to a consensus mechanism called proof-of-stake (PoS) from proof-of-work (PoW). This was always the plan as it's a key part in the community's strategy to scale Ethereum via the Eth2 upgrades. However getting PoS right is a big technical challenge and not as straightforward as using PoW to reach consensus across the network

Proof of stake- Key takeaways

- With Proof of Stake (POS), cryptocurrency miners can mine or validate block transactions based on the amount of coins a miner holds.
- Proof of Stake (POS) was created as an alternative to Proof of Work (POW), which is the original consensus algorithm in Blockchain technology, used to confirm transactions and add new blocks to the chain.
- Proof of Work (POW) requires huge amounts of energy, with miners needing to sell their coins to ultimately foot the bill; Proof of Stake (PoS) gives mining power based on the percentage of coins held by a miner.
- Proof of Stake (POS) is seen as less risky in terms of the potential for miners to attack the network, as it structures compensation in a way that makes an attack less advantageous for the miner.
- Bitcoin, the largest cryptocurrency, runs on proof of work rather than proof of stake.



Proof of Work vs. Proof of Stake

To add each block to the chain, miners must compete to solve a difficult puzzle using their computers processing power.

There is no competition as the block creator is chosen by an algorithm based on the user's stake.

In order to add a malicious block, you'd have to have a computer more powerful than 51% of the network.

In order to add a malicious block, you'd have to own 51% of all the cryptocurrency on the network.

The first miner to solve the puzzle is given a reward for their work.

There is no reward for making a block, so the block creator takes a transaction fee.

Proof of Authority

- In PoA-based networks, transactions and blocks are validated by approved accounts, known as validators.
- Validators run software allowing them to put transactions in blocks. The process is automated and does not require validators to be constantly monitoring their computers.
- Proof of Authority is currently being implemented as a more efficient alternative because it is able to perform much more transactions per second.
- It, however, does require maintaining the computer (the authority node) uncompromised. The term was coined by Gavin Wood, co-founder of Ethereum and Parity Technologies.
- With PoA, individuals earn the right to become validators, so there is an incentive to retain the position that they have gained.

Proof of Authority

- By attaching a reputation to identity, validators are incentivized to uphold the transaction process, as they do not wish to have their identities attached to a negative reputation.
- This is considered more robust than PoS (proof-of-stake) - PoS, while a stake between two parties may be even, it does not take into account each party's total holdings.
- This means that incentives can be unbalanced. On the other hand, PoA only allows non-consecutive block approval from any one validator, meaning that the risk of serious damage is centralized to the authority node.
- PoA is suited to private blockchains in which authority node identity may be established and disclosed within the private network. Some consider PoA ill-suited to public blockchains.

Proof of Activity

- **Proof-of-activity** (PoA) is a blockchain consensus algorithm. It is used to ensure that all transactions occurring on the blockchain are genuine, as well as to ensure that all miners arrive at a consensus.
- PoA is a combination of two other blockchain consensus algorithms: **proof**-of-work (PoW) and **proof**-of-stake (PoS).

Proof of Activity

KEY TAKEAWAYS (REF. HTTPS://WWW.INVESTOPEDIA.COM/)

- Proof-of-activity (PoA) is a blockchain consensus algorithm that is a combination of two other blockchain consensus algorithms: proof-of-work (PoW) and proof-of-stake (PoS).
- The PoA system is an attempt to combine the best aspects of both the PoW and the PoS systems; the mining process begins like a PoW system, but after a new block has been successfully mined, the system switches to resemble a PoS system.
- Decred (DCR) is the most well-known cryptocurrency that uses the PoA consensus mechanism.

Proof of Burn

- Proof of burn (POB) is an alternative consensus algorithm that tries to address the high energy consumption issue of a POW system.
- POB is often called a POW system without energy waste. It operates on the principle of allowing miners to “burn” virtual currency tokens
- Proof of burn is one of the several consensus mechanism algorithms implemented by a blockchain network to ensure that all participating nodes come to an agreement about the true and valid state of the blockchain network.
- This algorithm is implementing in order to avoid the possibility of any cryptocurrency coin double-spending.

Proof of Burn

KEY TAKEAWAYS

- Cryptocurrencies use several methods to validate the data stored on their blockchains, including a method called "proof of burn."
- Proof of burn is the third attempt at creating a system to deter fraudulent activity on a blockchain, while also improving the functioning of the blockchain as a tool for transactions.
- Proof of work and proof of stake are also methods for preventing fraudulent activity on a blockchain; proof of work is the system employed by the original and most popular cryptocurrency, Bitcoin.

Proof of Capacity

- **Proof of capacity** (PoC) is a consensus mechanism algorithm used in blockchains that allows for mining devices in the network to use their available hard drive space to decide mining rights and validate transactions.
- This is in contrast to using the mining device's computational power (as in the proof of work algorithm) or the miner's stake in the cryptocurrencies (as in the proof of stake algorithm).

Proof of Capacity

- **Key Takeaways**

- Proof of capacity (PoC) authentication systems employ spare space on a device's hard drive to store solutions to a cryptocurrency hashing problem.
- The main benefit of a PoC system is its efficiency compared to proof-of-work (PoW) and proof-of-stake (PoS) systems.
- Blockchains that run on proof of capacity include Storj, Burst, Chia, and SpaceMint.

Block Reward

What Is a Block Reward

Bitcoin block reward refers to the new bitcoins that are awarded by the blockchain network to eligible cryptocurrency miners for each block they mine successfully.

<u>KEY TAKEAWAYS</u>

❑ *A block reward refers to the number of bitcoins you get if you successfully mine a block of the currency.*

❑ *The amount of the reward halves every 210,000 blocks, or roughly every four years.*

❑ *The amount is expected to hit zero around 2140.*

Understanding Block Reward

- ❑Each bitcoin block is around 1 MB in size and is used to store the bitcoin transaction information. For example, when A sends money to B, this transaction information is stored on a block.
- ❑Miners who use mining devices to find new blocks are rewarded for their efforts through block rewards. Other cryptocurrencies have a similar mechanism for rewarding miners with blocks of the respective blockchain. The winning miner claims a block reward by adding it as a first transaction on the block.
- ❑At inception, each bitcoin block reward was worth 50 BTC. The block reward is halved after the discovery of every 210,000 blocks, which takes around four years to complete. As of February 2019, one block reward was worth 12.5 BTC.
- ❑Working on the principle of a standard cryptocurrency economy with declining bitcoins awarded as block rewards, fewer new bitcoins will be available over time, and that will keep bitcoin prices high. After 64 iterations of halving the block reward, it will eventually become zero.

Transaction Fees in Block Chain

❑Average **Bitcoin transaction fees** can spike during periods of congestion on the network, as they did during the 2017 **Crypto** boom where they reached nearly 60 USD.

❑**Bitcoin** Average **Transaction Fee** is at a current level of 26.86, down from 29.69 yesterday and up from 0.6994 one year ago.

❑Before going into different for **transaction** speeds, **Bitcoin transactions** generally **take** anywhere from 10 minutes to 1 hour. **The** reason for **the** range in time **is** that different situations require different amounts of confirmations (1 confirmation **takes** ~10 minutes) for a **transaction**

Transaction fees

- ❑Bitcoin miners get paid all the transaction fees in the block they mine. So as such, it is in their interest to maximize the amount of money they make when they create a block. So what they do is pick the 1,000,000 bytes of transactions that results them getting paid the most money.
- ❑From a bitcoin miner perspective, they don't care of the *value* of a transaction, but just the size (amount of bytes), because they are only allowed to create blocks of 1,000,000 bytes or less. So miners don't consider the absolute fee a transaction has, but rather, the **fee per byte**.

Why fees high and changes

- ❑Sometimes fees are high when there is a lot of demand for blockspace due to new investors coming in.
- ❑Remember that there can be only so many transactions per block. And there is a sort of auction that occurs to determine who's transactions make it in and who's don’t.
- ❑If there are a lot of people who really need to get into the next block, they will pay for the privilege.
- ❑Wait for demand to die down and fees will be almost 0.
- ❑Fees have been coming down since large exchanges like Coinbase have been batching payments.

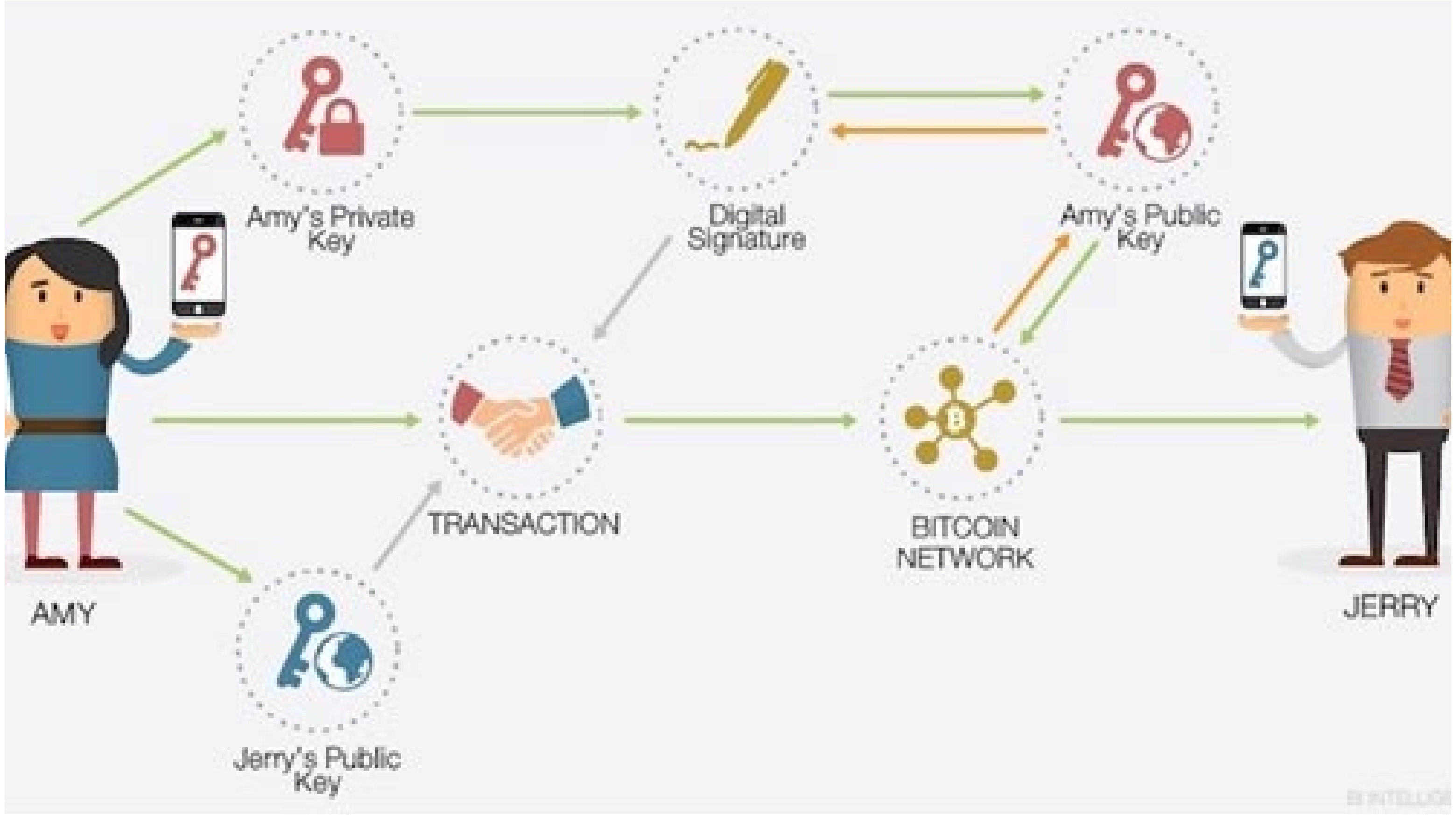
Economics behind tx fees in block chains

- ❑A Bitcoin transaction has to be added to the Blockchain in order to be successfully completed. However, for a transaction to be added to the Blockchain, it first needs to be validated by miners who solve a complex mathematical problem to verify the transaction. These miners spend a lot of computing power and energy when verifying a block of transactions from the Bitcoin Mempool (short for memory pool), which contains unconfirmed transactions waiting to be added to a block for confirmation.
- ❑Now, miners need to be incentivized for the time, effort, and resources that they are putting in to validate the unconfirmed transactions. As a result, they are given a fee of 12.5 BTC to successfully mine a block, but this is just one of the incentives on offer. Miners also earn a transaction fee that’s selected by the sender in a Bitcoin transaction for their effort as they play a critical role in keeping the network secure.

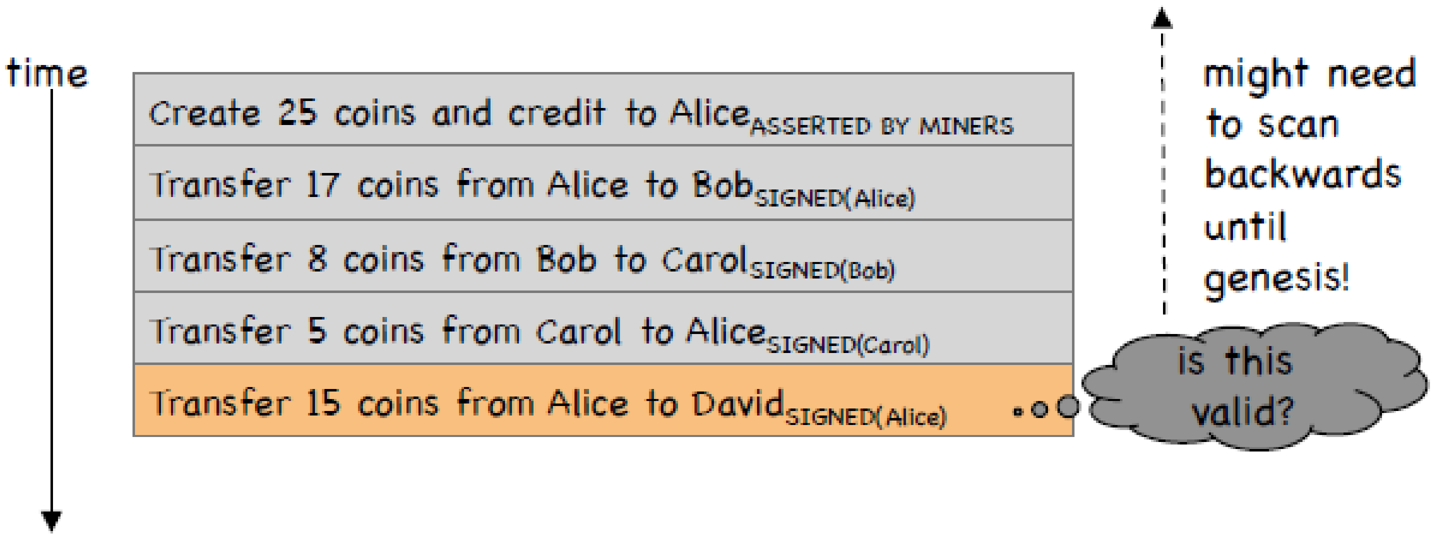
Bitcoin transactions

- ❑A transaction is the fundamental building block of the bitcoin blockchain. It’s the operation in which we transfer value from one party to another in a secure way.
- ❑**Bitcoin units**
- ❑The general unit structure of bitcoins has 1 **bitcoin (BTC**) equivalent to 1,000 millibitcoins (**mBTC**), 1,000,000 microbitcoins (µBTC), or 100,000,000 **satoshis**.
- ❑While the exact figure is unknown, it is estimated that **Satoshi** Nakamoto may possess 1 **million** bitcoins, equivalent to 100,000,000,000,000 **satoshis**.

Bitcoin transactions



An account-based Ledger (not Bitcoin)



time

Create 25 coins and credit to Alice$_{\text{ASSERTED BY MINERS}}$

Transfer 17 coins from Alice to Bob$_{\text{SIGNED(Alice)}}$

Transfer 8 coins from Bob to Carol$_{\text{SIGNED(Bob)}}$

Transfer 5 coins from Carol to Alice$_{\text{SIGNED(Carol)}}$

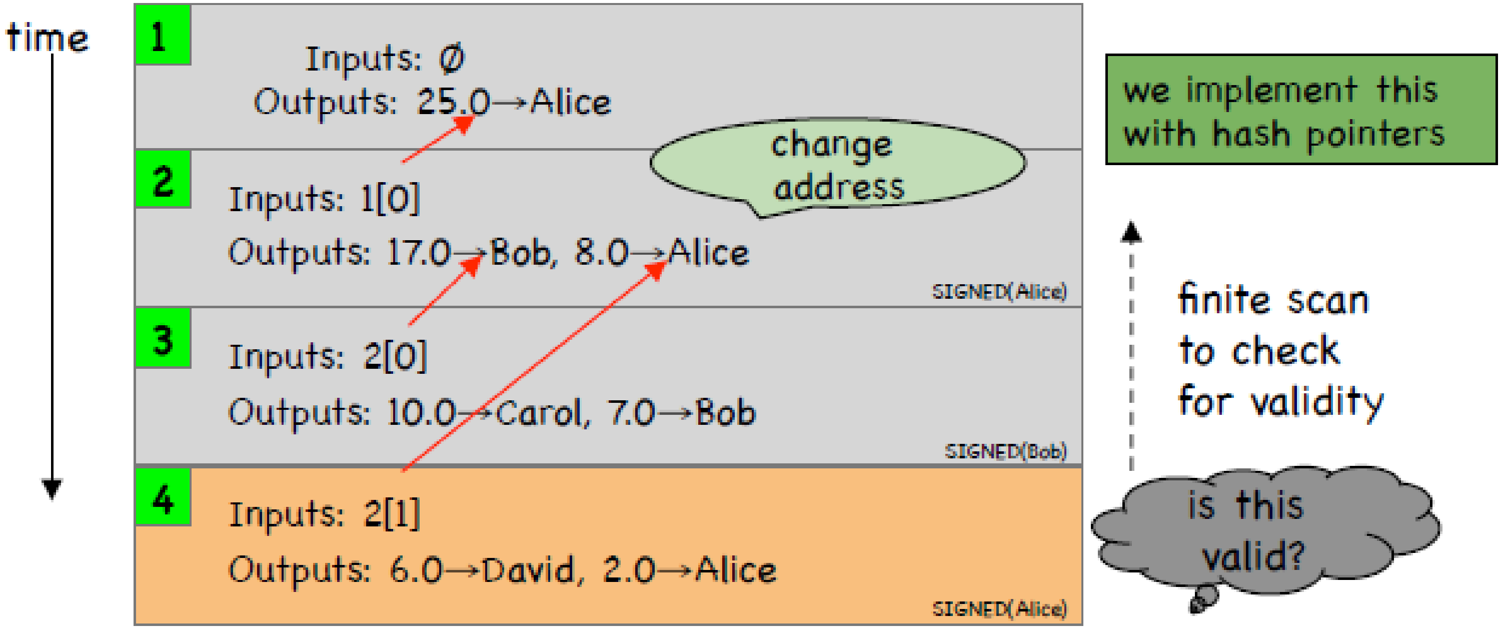
Transfer 15 coins from Alice to David$_{\text{SIGNED(Alice)}}$

is this valid?

might need to scan backwards until genesis!

SIMPLIFICATION: only one transaction per block

A transaction-based Ledger (Bitcoin)



time

1 Inputs: Ø
Outputs: 25.0→Alice

2 Inputs: 1[0]
Outputs: 17.0→Bob, 8.0→Alice
SIGNED(Alice)

3 Inputs: 2[0]
Outputs: 10.0→Carol, 7.0→Bob
SIGNED(Bob)

4 Inputs: 2[1]
Outputs: 6.0→David, 2.0→Alice
SIGNED(Alice)

change address

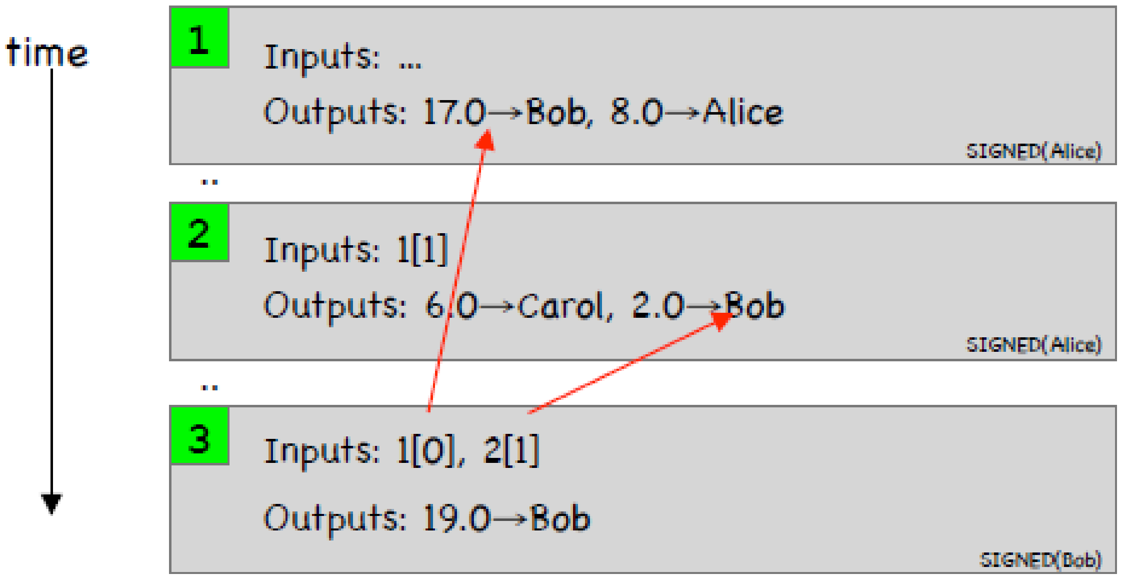
we implement this with hash pointers

finite scan to check for validity

is this valid?

SIMPLIFICATION: only one transaction per block

Merging Value



time

1

Inputs: ...

Outputs: 17.0→Bob, 8.0→Alice

SIGNED(Alice)

..

2

Inputs: 1[1]

Outputs: 6.0→Carol, 2.0→Bob

SIGNED(Alice)

..

3

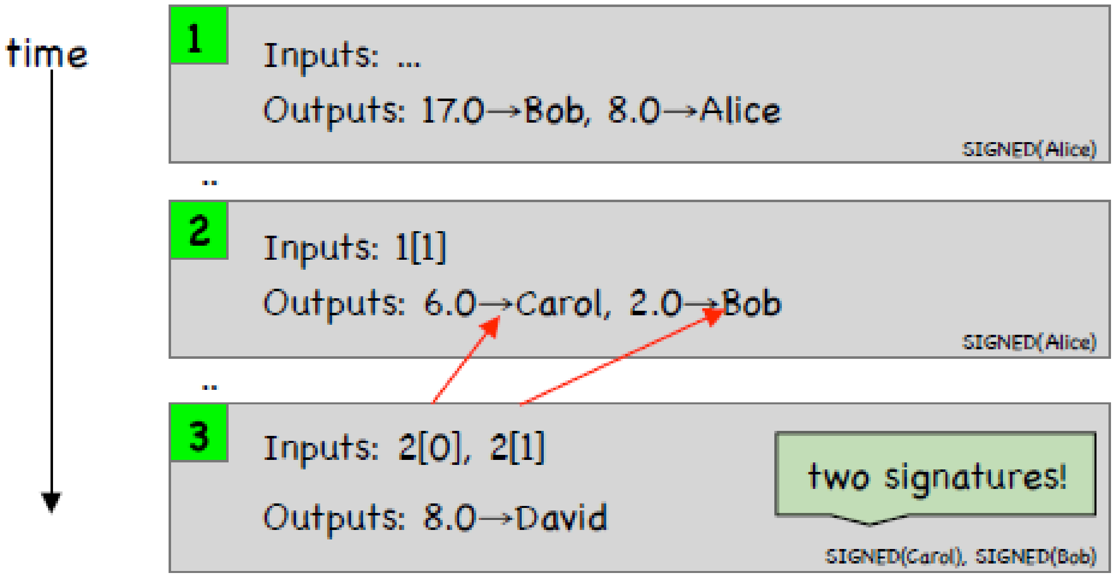
Inputs: 1[0], 2[1]

Outputs: 19.0→Bob

SIGNED(Bob)

SIMPLIFICATION: only one transaction per block

Joint Payments



time

1

Inputs: ...

Outputs: 17.0→Bob, 8.0→Alice

SIGNED(Alice)

..

2

Inputs: 1[1]

Outputs: 6.0→Carol, 2.0→Bob

SIGNED(Alice)

..

3

Inputs: 2[0], 2[1]

Outputs: 8.0→David

two signatures!

SIGNED(Carol), SIGNED(Bob)

SIMPLIFICATION: only one transaction per block

Bitcoin Script

- Bitcoin Script is a simple, stack-based programming language that enables the processing of transactions on the Bitcoin blockchain.

Bitcoin Scripting Language ("Script")

Design "goals":

- Built for Bitcoin (inspired by Forth)
- Simple, compact
- Stack-based
- No looping
- Support for cryptography
- Limits on time/memory
- Not Turing complete!

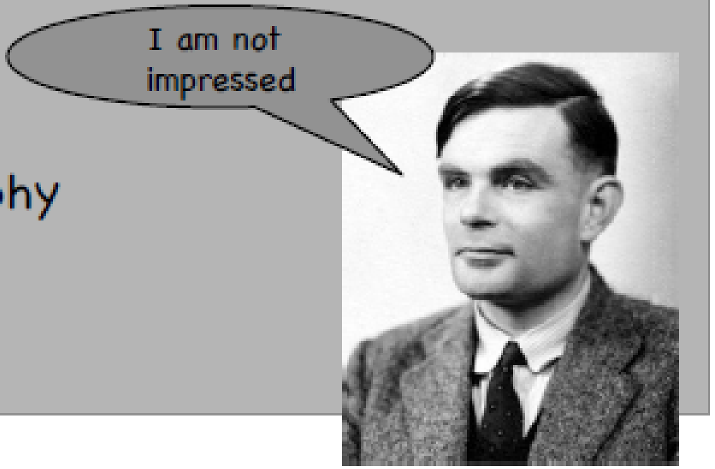
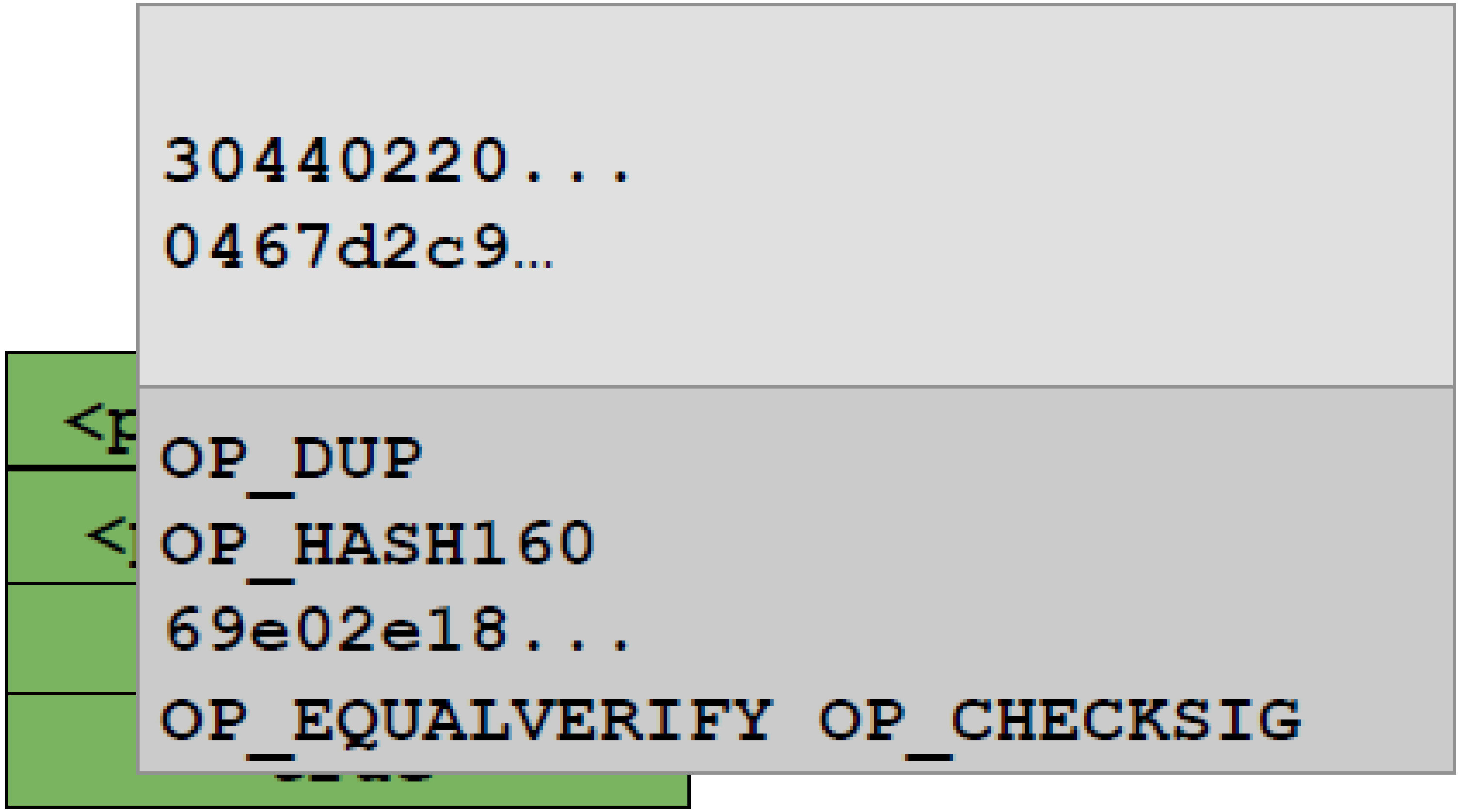


image via Jessie St. Amand

Bitcoin Script Execution Example

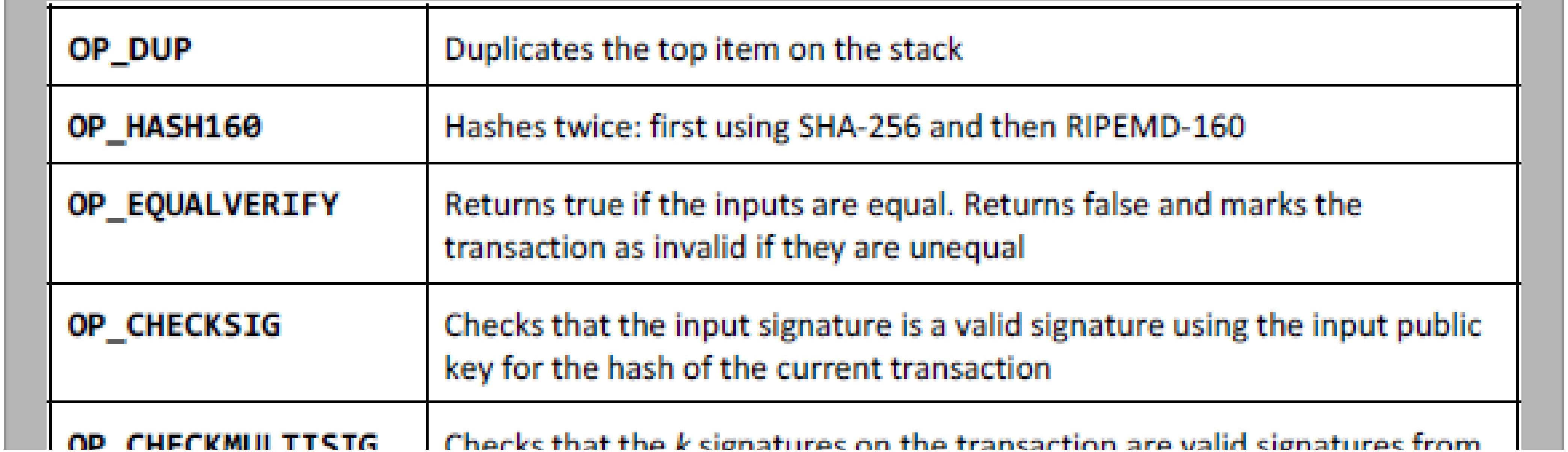


```
30440220...
0467d2c9…
```

```
OP_DUP
OP_HASH160
69e02e18...
OP_EQUALVERIFY OP_CHECKSIG
```

256 opcodes total (15 disabled, 75 reserved)

- Arithmetic
- If/then
- Logic/data handling
- Crypto!

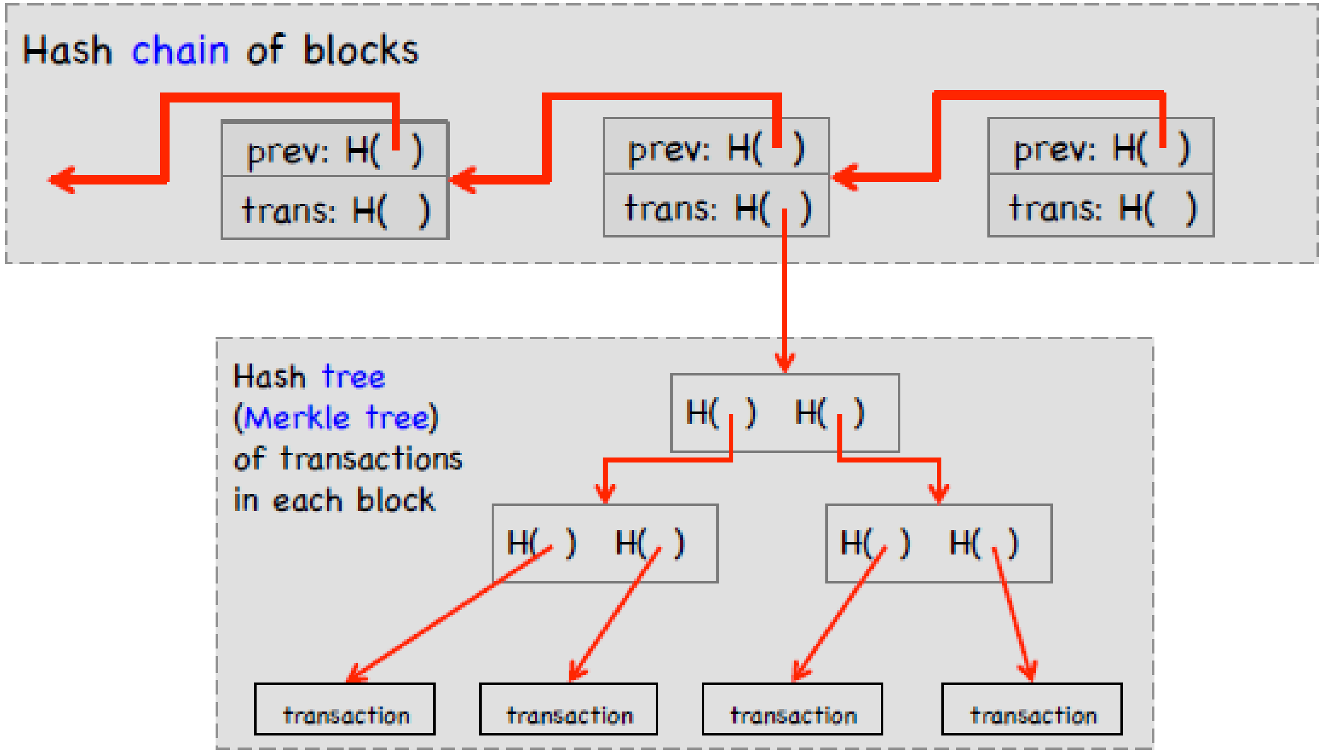


OP_DUP	Duplicates the top item on the stack
OP_HASH160	Hashes twice: first using SHA-256 and then RIPEMD-160
OP_EQUALVERIFY	Returns true if the inputs are equal. Returns false and marks the transaction as invalid if they are unequal
OP_CHECKSIG	Checks that the input signature is a valid signature using the input public key for the hash of the current transaction
OP_CHECKMULTISIG	Checks that the *k* signatures on the transaction are valid signatures from

Q: Why bundle transactions together?

1. Requiring consensus for each transaction separately would reduce transaction acceptance rate.
2. Hash-chain of blocks is much shorter.
3. Faster to verify history.

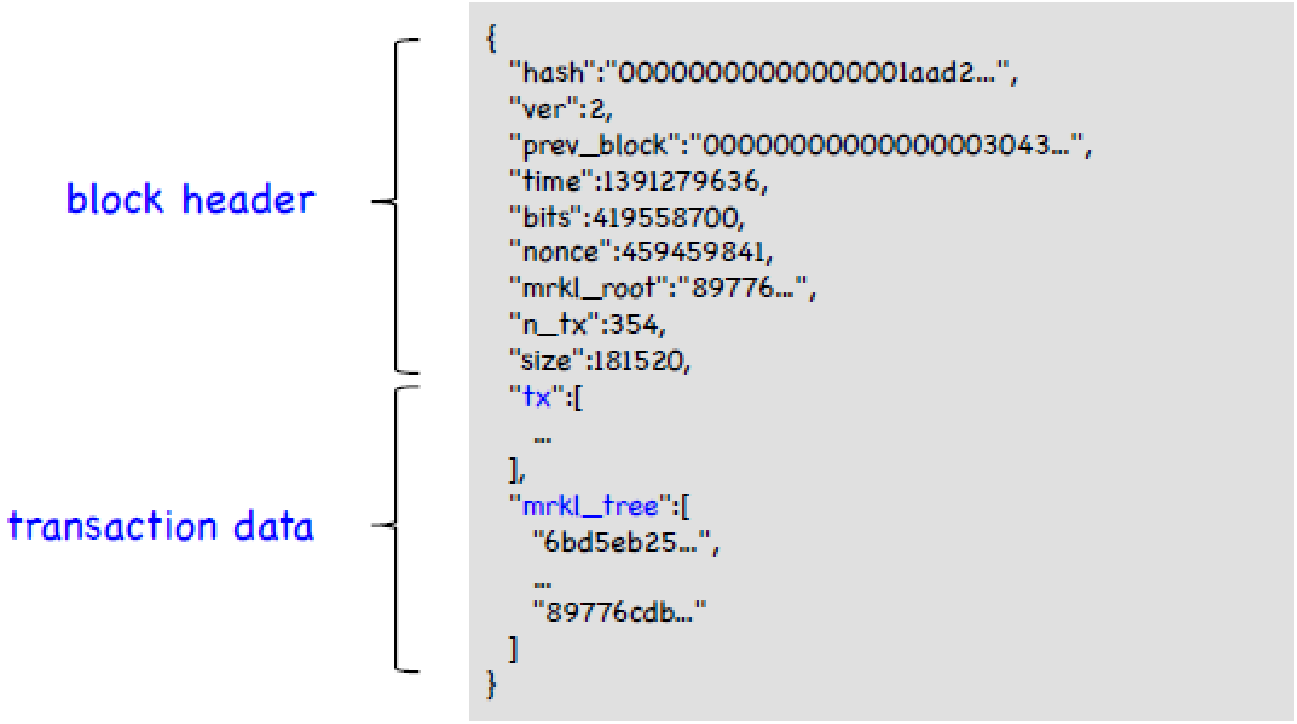
Bitcoin Block Structure



Merkle benefits

- **Merkle Trees** have four sizable **benefits**
- They provide a way to prove both the integrity and validity of data.
- They significantly reduce the amount of memory needed to do the above.
- The required proof and management only needs small amounts of information to be transmitted across networks

The Real Deal: a Bitcoin Block

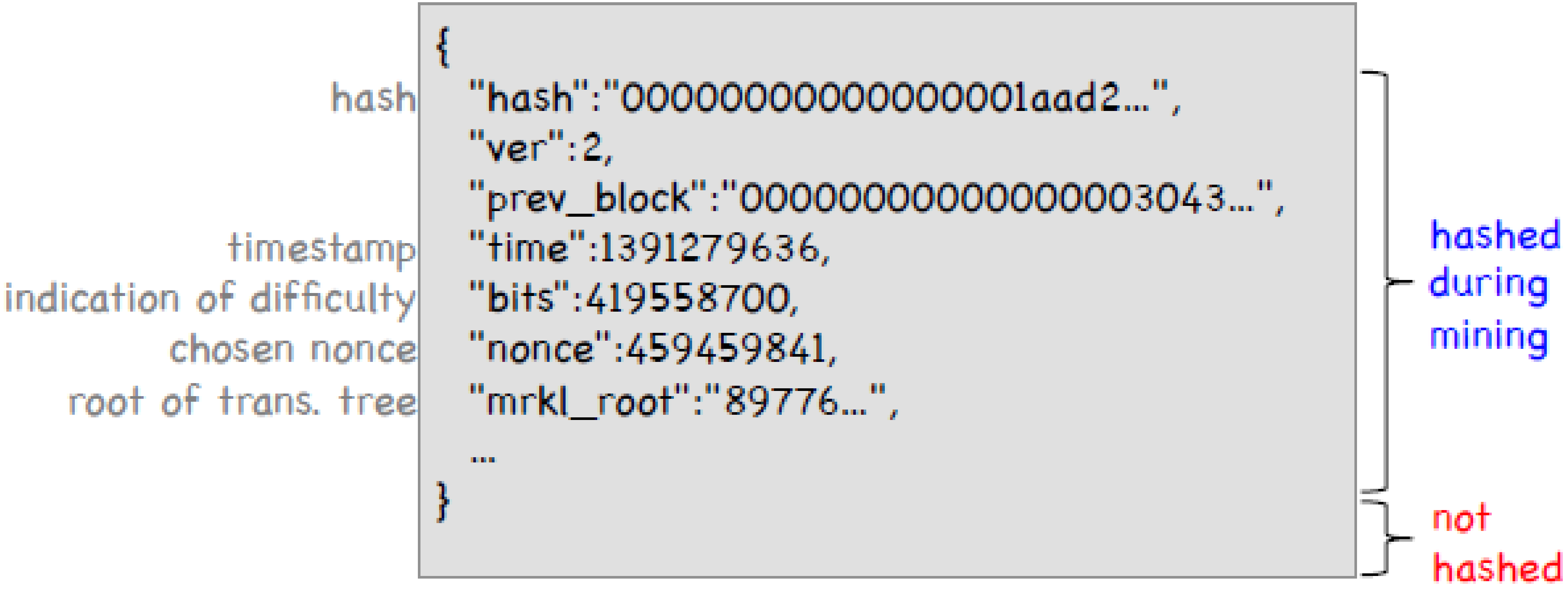


block header

transaction data

```
{
  "hash":"0000000000000001aad2...",
  "ver":2,
  "prev_block":"00000000000000003043...",
  "time":1391279636,
  "bits":419558700,
  "nonce":459459841,
  "mrkl_root":"89776...",
  "n_tx":354,
  "size":181520,
  "tx":[
    ...
  ],
  "mrkl_tree":[
    "6bd5eb25...",
    ...
    "89776cdb..."
  ]
}
```

The Real Deal: a Bitcoin Block Header



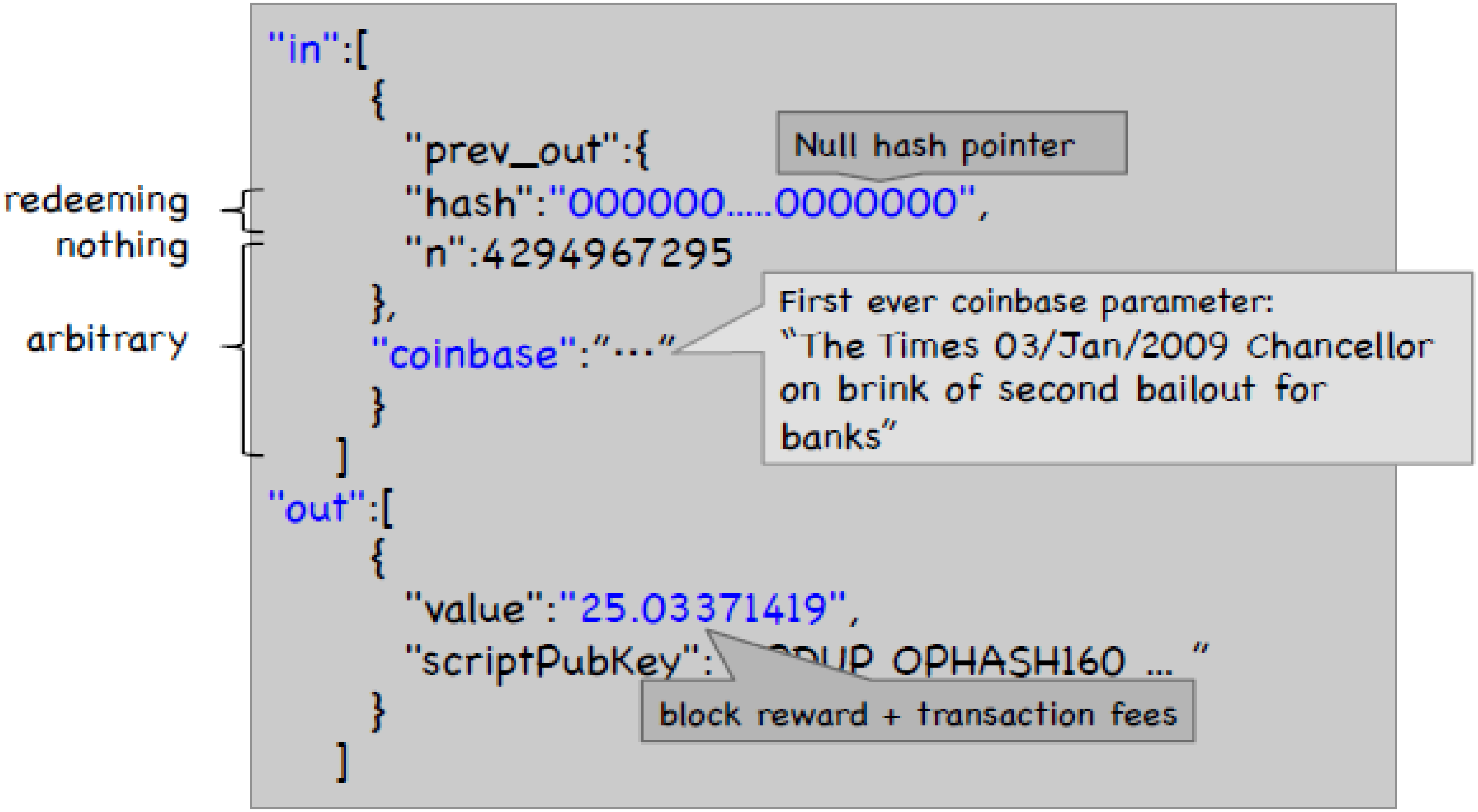
```
{
  "hash":"0000000000000001aad2...",
  "ver":2,
  "prev_block":"0000000000000003043...",
  "time":1391279636,
  "bits":419558700,
  "nonce":459459841,
  "mrkl_root":"89776...",
  ...
}
```

coinbase Transaction

New coins are created with coinbase transaction:

- Single input and single output
- Does not redeem previous output
  - Hash pointer is null
- Output value is miner's revenue from block:
  - *output value = mining reward + transaction fees*
  - transaction fees come from all transactions in block
- Special coinbase parameter
  - contains arbitrary value

The Real Deal: coinbase Transaction



```
"in":[
      {
        "prev_out":{
        "hash":"000000.....0000000",
        "n":4294967295
      },
      "coinbase":"..."
      }
    ]
"out":[
      {
        "value":"25.03371419",
        "scriptPubKey":"OPDUP OPHASH160 ... "
      }
    ]
```

redeeming nothing

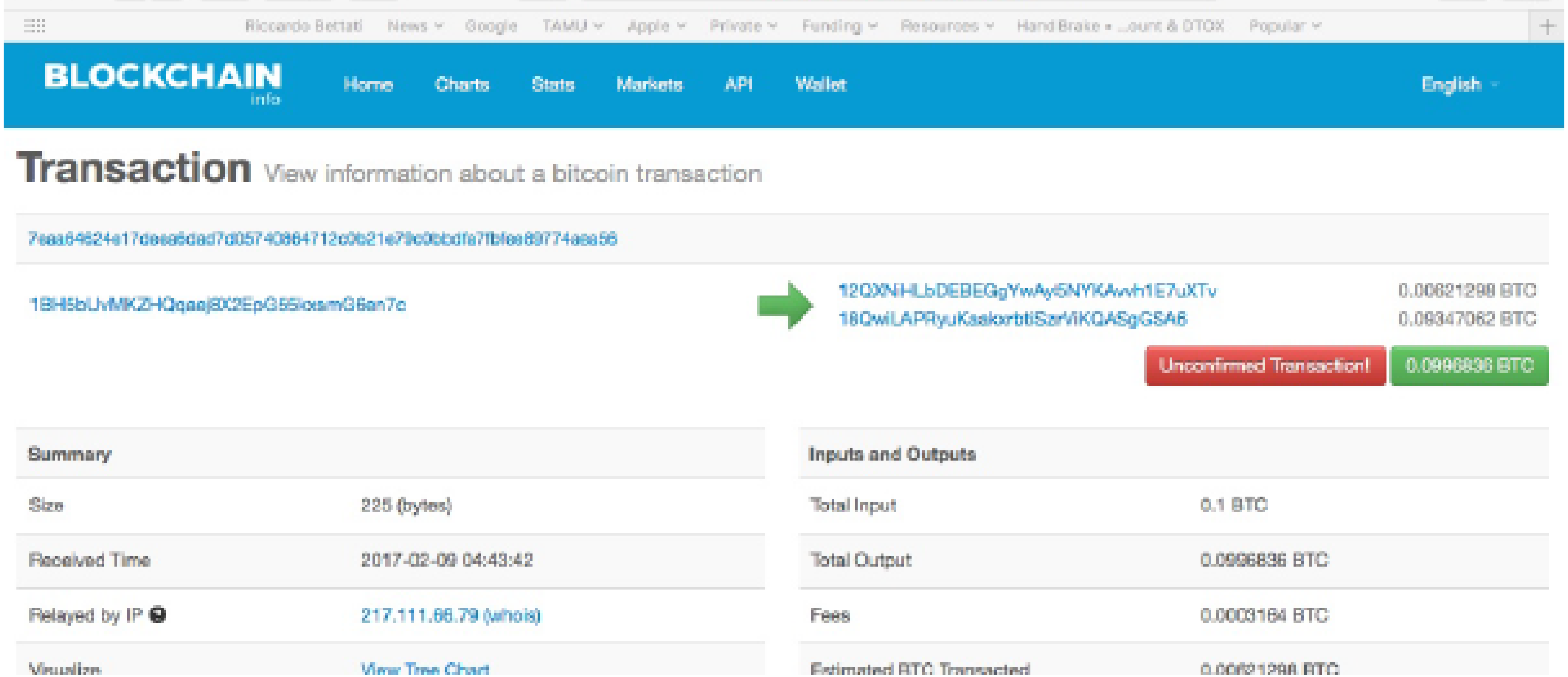
arbitrary

Null hash pointer

First ever coinbase parameter: “The Times 03/Jan/2009 Chancellor on brink of second bailout for banks”

block reward + transaction fees

See for yourself!



Transaction View information about a bitcoin transaction

7eaa64624e17deea6dad7d0574086471 2c0b21e79c0bbdfa7fbfee89774aea56

1BH5bUvMKZHQqaej8X2EpG55ioxsmG6an7c → 12QXNHLbDEBEGgYwAy5NYKAwh1E7uXTv 0.00621298 BTC

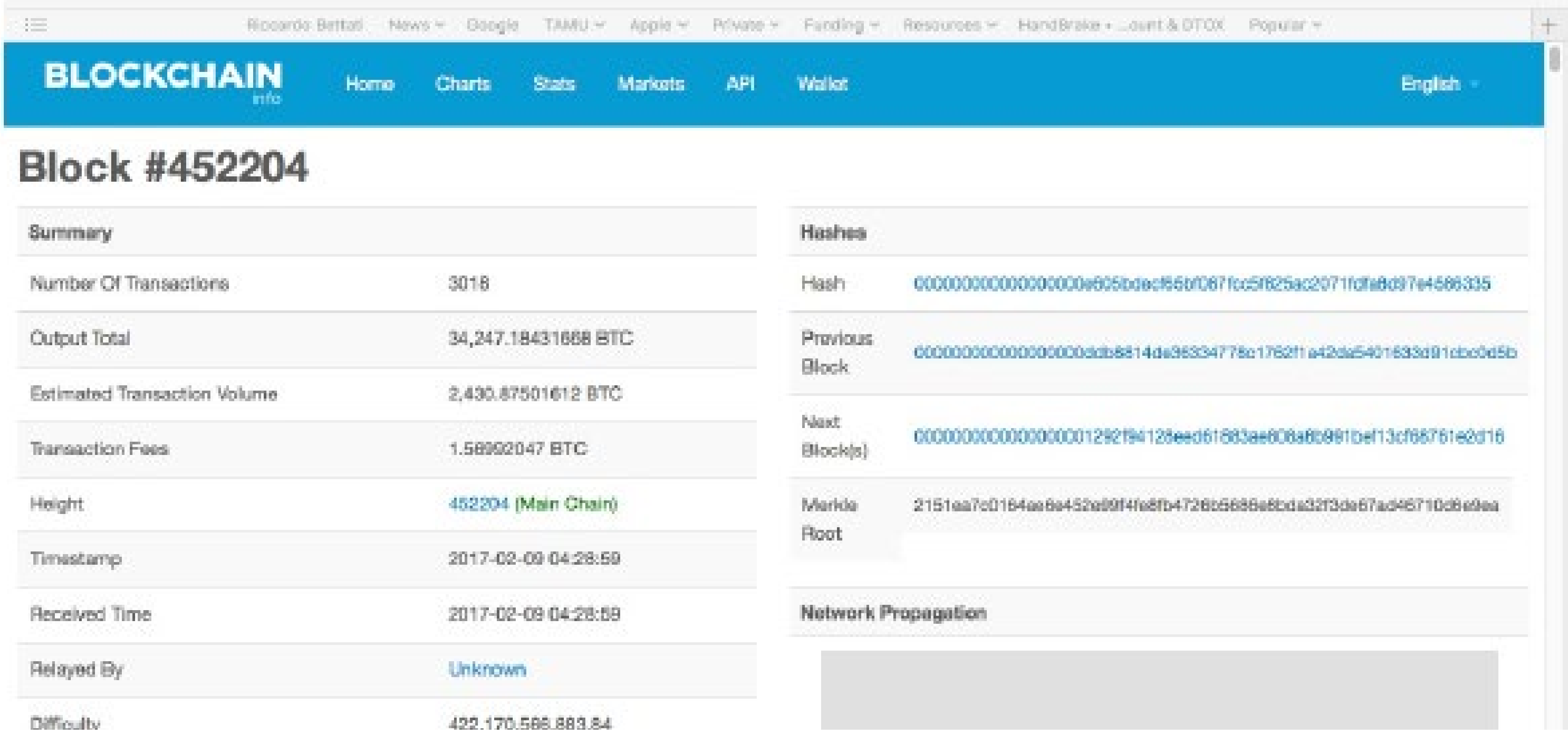
18QwiLAPRyuKaakxrbtiSzrViKQASgGSA6 0.09347062 BTC

Unconfirmed Transaction! 0.0996836 BTC

Summary	
Size	225 (bytes)
Received Time	2017-02-09 04:43:42
Relayed by IP	217.111.66.79 (whois)
Visualize	View Tree Chart

Inputs and Outputs	
Total Input	0.1 BTC
Total Output	0.0996836 BTC
Fees	0.0003164 BTC
Estimated BTC Transacted	0.00621298 BTC

See for yourself!



Block #452204

Summary	
Number Of Transactions	3018
Output Total	34,247.18431668 BTC
Estimated Transaction Volume	2,430.87501612 BTC
Transaction Fees	1.58992047 BTC
Height	452204 (Main Chain)
Timestamp	2017-02-09 04:28:59
Received Time	2017-02-09 04:28:59
Relayed By	Unknown
Difficulty	422,170,586,883.84

Hashes	
Hash	0000000000000000000e605bdecf55bf067fcc5f825ac2071fdfa8d97e4586335
Previous Block	0000000000000000000ddb8814da36334778c1762f1a42da5401633d91cbc0d5b
Next Block(s)	000000000000000000001292f94128eed61583ae808a80991bef13cf68761e2d16
Merkle Root	2151aa7c0164ae6a452e89f4fe8fb4726b5686e8bda32f3de67ad46710d8e9ea

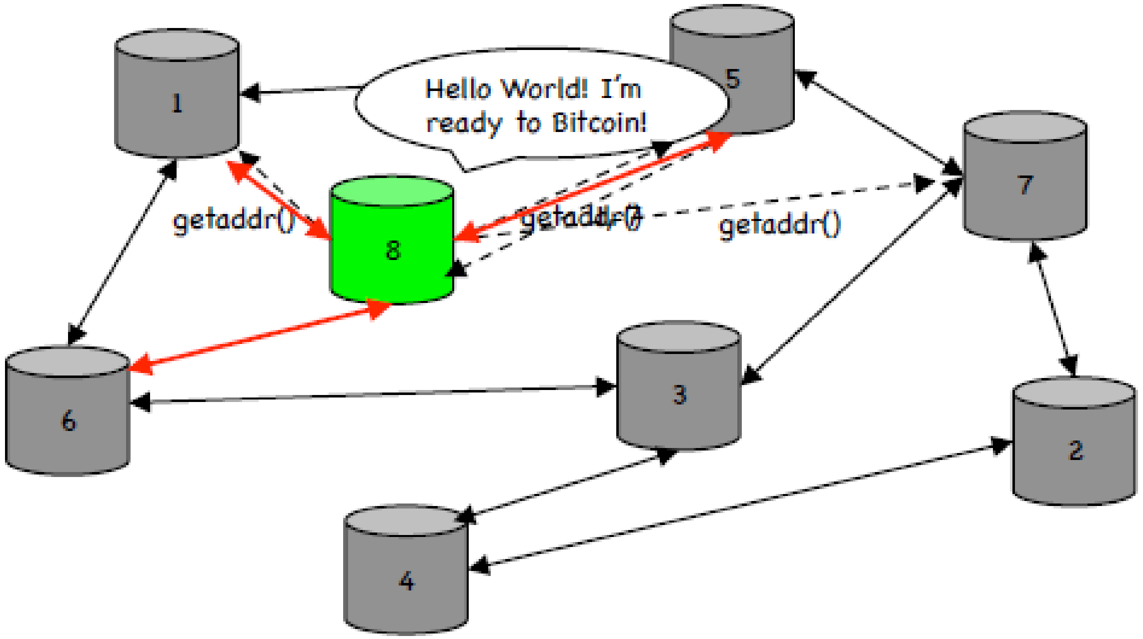
Network Propagation

Bitcoin P2P Network

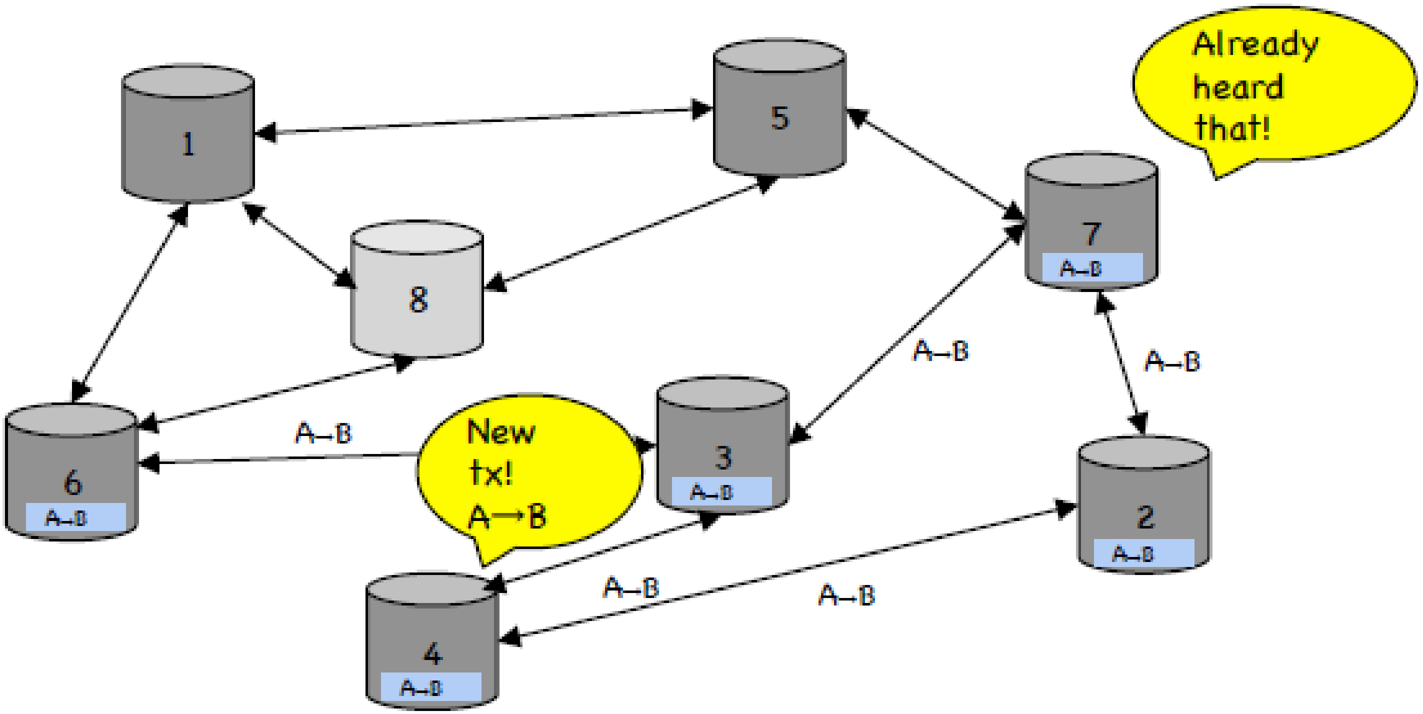
Participants can
- publish transactions
- insert transactions into block chain

The network:
- Ad-hoc protocol (runs on TCP port 8333)
- Ad-hoc network with random topology
- All nodes are equal
- New nodes can join at any time
- Forget non-responding nodes after 3 hr

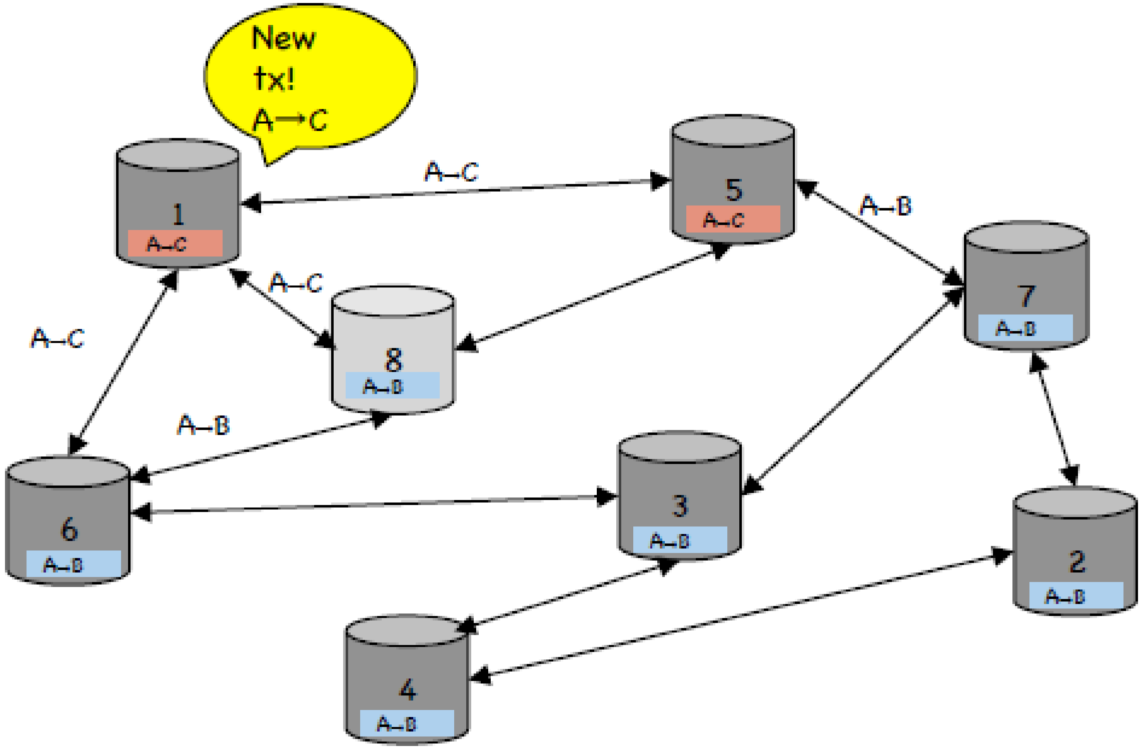
Joining the Bitcoin P2P Network



Transaction Propagation (Flooding)



Nodes may differ on Transaction Pool



Race Conditions

Transactions or blocks may conflict

- This is called **"race condition"**
- Default behavior: accept what you hear first
- Tie broken by whoever mines next block
  - picks only one transaction/block
- Network position matters
- Miners may implement other logic!

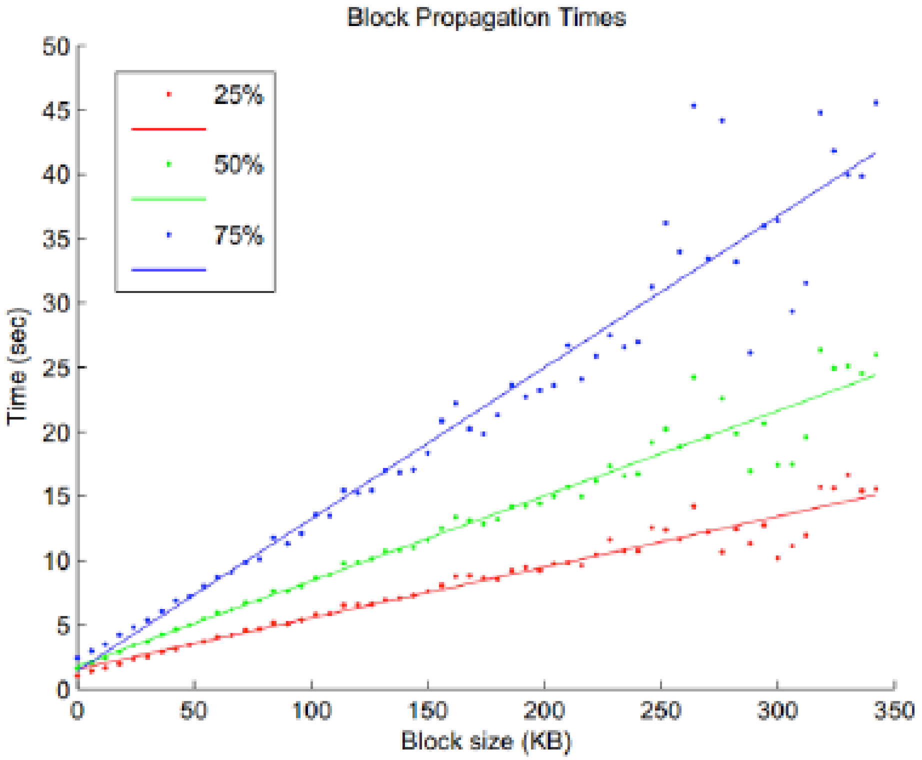
Block Propagation

Propagation of blocks is nearly identical:

Relay a new block when you hear it if:

1. Block meets the hash target
2. Block has all valid transactions
   - Run *all* scripts, even if you wouldn't relay
3. Block builds on current longest chain
   - Avoid forks

Latency of Flooding Algorithm



Source: Yonatan Sompolinsky and Aviv Zohar: "Accelerating Bitcoin's Transaction Processing" 2014

Size of the Network

Q: How big is the Network?

Impossible to measure exactly

- Estimates-up to 1M IP addresses/month
- Only about 5-10k "full nodes"
  - Permanently connected
  - Fully-validating
- This number may be dropping!

Fully-validating Nodes:

- Permanently connected
- Store entire block chain
- Hear and forward every node/ transaction

Hard-coded Limits in Bitcoin

- 10 min. average creation time per block
- 1 M bytes in a block
- 20,000 signature operations per block
- 100 M *satoshis* per bitcoin
- 23M total bitcoins maximum
- 50,25,12.5... bitcoin mining reward

These affect economic balance of power too much to change now