

Decentralization

How Bitcoin works?

What is Bitcoin?

Bitcoin = Scroogecoin without Scrooge

How to remove Scrooge?

Two components that we need to achieve decentralization:

- Technical mechanisms (blockchain, proof of work, mining)
- Economic incentives (block reward, transaction fee, community)

They depend on each other in a cyclic manner:

- Technical mechanisms assure security if many people participate in the protocol
- People participate in the protocol if the economic incentives are good

Centralisation vs. Decentralization

A fully decentralized system: Internet

A centralized system: A social network (Facebook, etc.)

Semi - decentralized: email

There is no fully decentralized system

Not even Bitcoin:

- P2P network = completely decentralized
- Mining = almost completely centralized
- Protocol updates = BitcoinCore developers

Decentralization

Some questions

1. **Who stores the blockchain?**
2. **Who decides which transactions are valid?**
3. **Who creates new Bitcoins?**
4. Who decides how to change the protocol rules?
5. How does Bitcoin manage to have value in USD?

Distributed consensus

Distributed consensus is our method to achieve decentralization

Distributed consensus. All the participants in the system need to be in agreement on which transactions are valid. Only in this case a transaction can appear in a block of our blockchain.

Distributed = There is no central entity deciding which transactions are valid. All the participants in the protocol need to be in agreement = **Consensus**.

A common problem in CS

Distributed databases

- E.g. A social network has its data in thousands of servers
- Servers = nodes in a distributed system
- Many data are replicated
- To update a data point, we need to update it on all the servers storing it
- This is achieved via distributed consensus
- A classic application of distributed consensus: a global dictionary

Distributed consensus

Basic protocol of distributed consensus.

We have a network of n nodes. Each node has an input value. Some nodes are malicious. The protocol achieves the following:

1. Upon executing the protocol, all honest nodes are in agreement about the value of the input.
2. The value was generated by a honest node.

Distributed consensus in Bitcoin

What does this mean in Bitcoin?

- Bitcoin is a p2p network of nodes
- Nodes receive transactions
- The nodes need to be in agreement on which transactions are valid
- After executing the protocol, these transactions are added to the global ledger

The nodes want to reach consensus on which transactions are valid at a certain moment in time.

This consensus replaces Scrooge.

Distributed consensus in Bitcoin

Alice wants to pay Bob (two steps):

1. Alice creates the transaction (Inputs, Outputs, signatures)
2. Alice transmits the transaction on the Bitcoin p2p network

Important: It is not necessary that Alice or Bob be nodes on the network.

Alice does not send the transaction to Bob. She only sends it to the network. It does not matter if Bob is not aware of the transaction. If the network accepts it, Bob owns the funds.

Alice pays Bob

Step 1



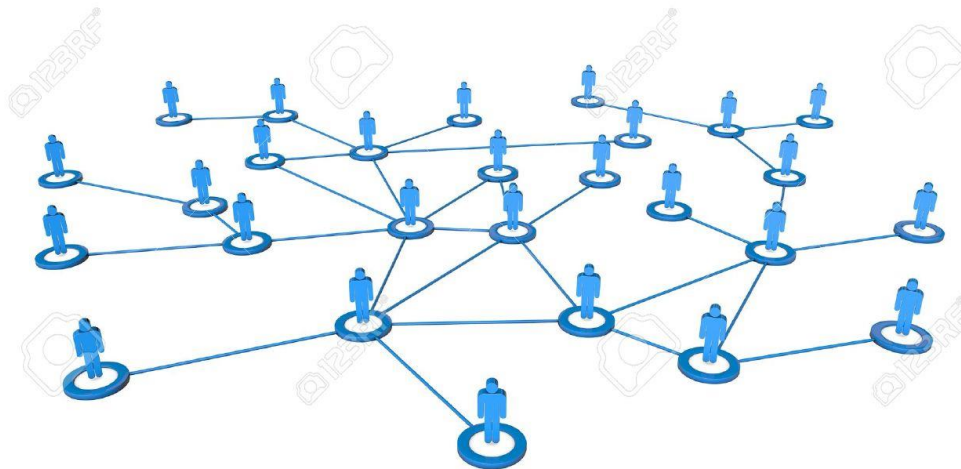
transID: 74		type: PayCoins	
coins consumed			
num	consumed coinID		
0	coinID 73(1)		
coins created			
num	value	recipient	
0	3.2	0xf4...	
Signature by LS _{Alice}			

Alice pays Bob

Step 2



transID:...



What happens in Bitcoin?

How will we store our ledger?

Same as in Scroogecoin. We will use a blockchain.

In each step, the nodes want to reach a consensus on the next block to be added.

Important: All the nodes store the blockchain. The entire blockchain.

There is no central entity storing the blockchain for us!!!

How Bitcoin works

1. At each point the nodes of the Bitcoin network have a blockchain of valid transactions (the ones over which we have a consensus -- almost)
2. The nodes receive new transactions and add them to the pool of transactions that need to be added to the blockchain
3. **Each 10 minutes, every node proposes the next block (based on the transactions in its pool)**
4. **Nodes execute the distributed consensus protocol (input = the block)**
5. If the consensus is reached, we will select a valid block (no double spends,...)

How Bitcoin works

Remarks:

- There are malicious nodes
- If only one node proposes the selected block, this block is still valid
- If a transaction was not included in this block there is no drama (it will be included in one of the subsequent blocks)

Is it easy to achieve this?

Implementation problems:

- The p2p network is not perfect (not all the nodes see all the transactions)
- Nodes join and leave the network
- The nodes do not have a fixed entity (we can not indentify a node)
- There is no global time valid for all the nodes

Global time



Global time



Created: 10:00

transA



Global time



Created: 10:00

transA



Created: 10:05

transB

Global time



Created: 10:00

transA



Created: 10:05

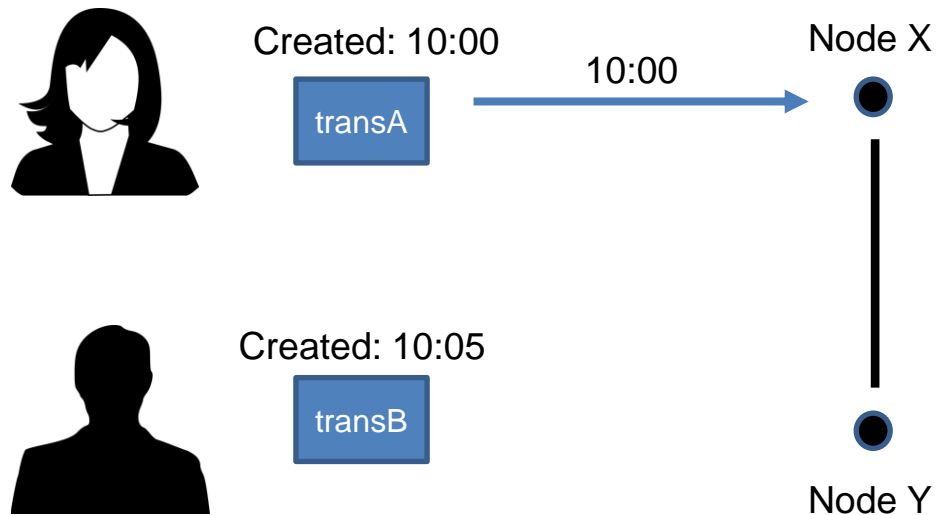
transB

Node X

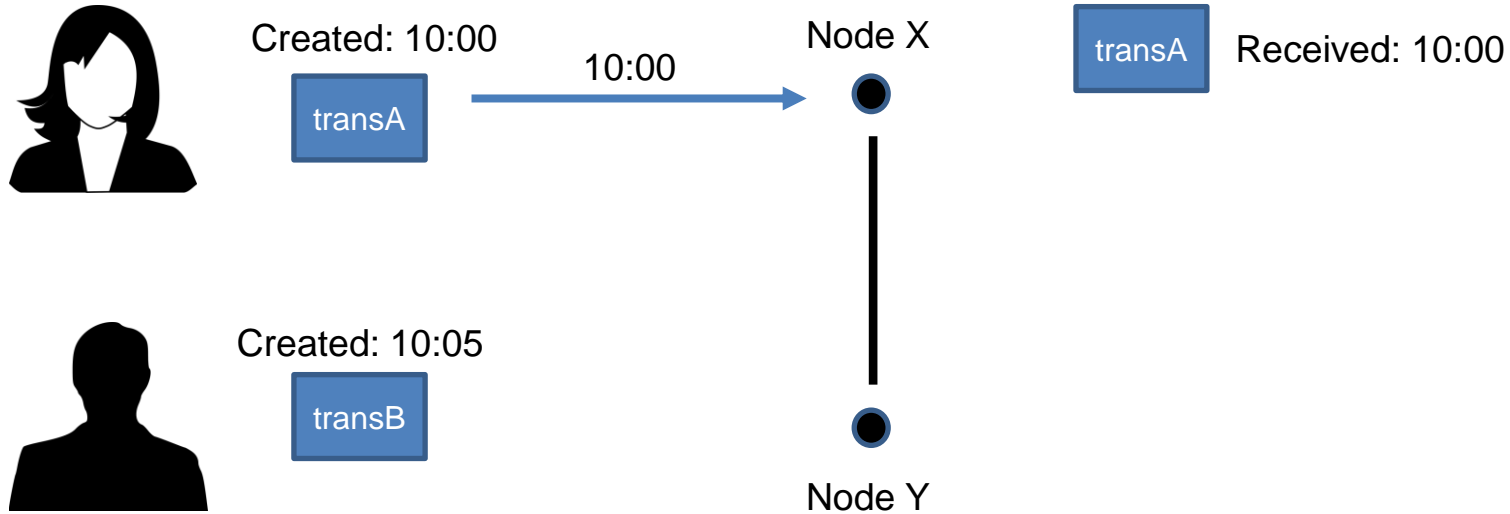


Node Y

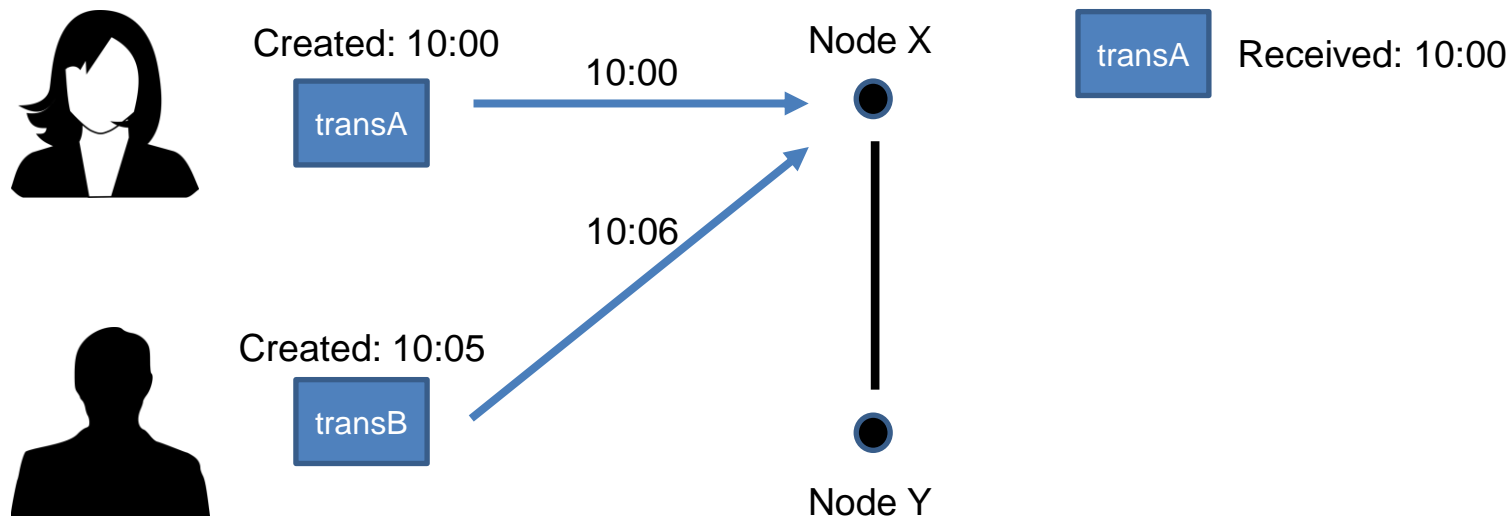
Global time



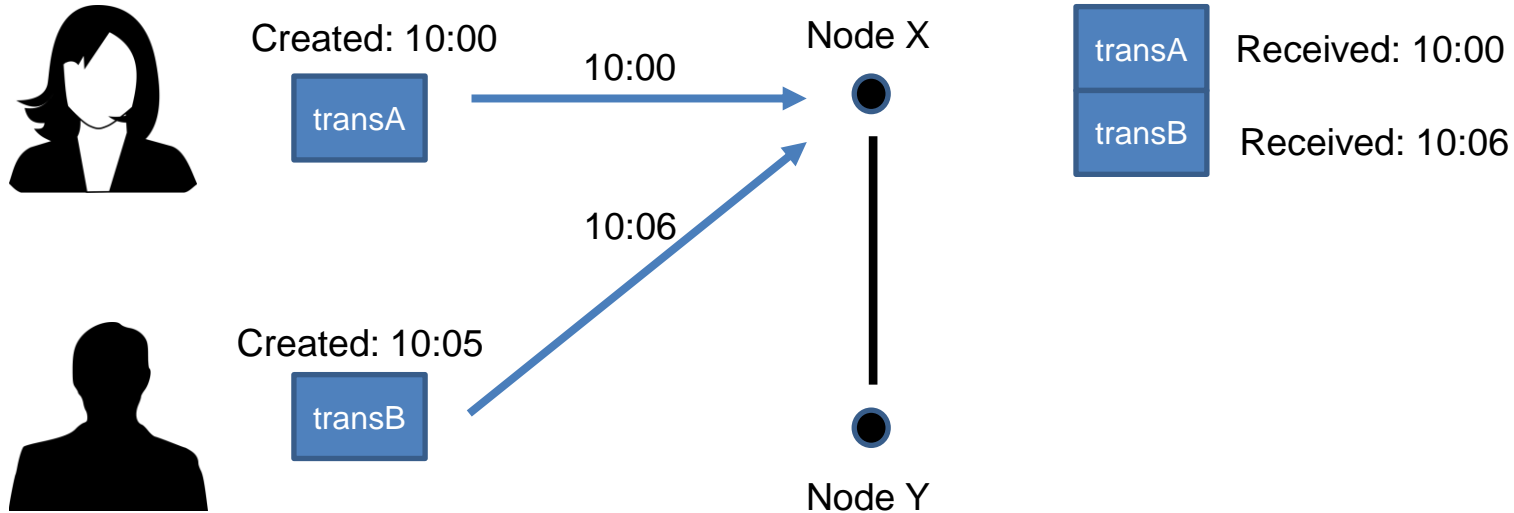
Global time



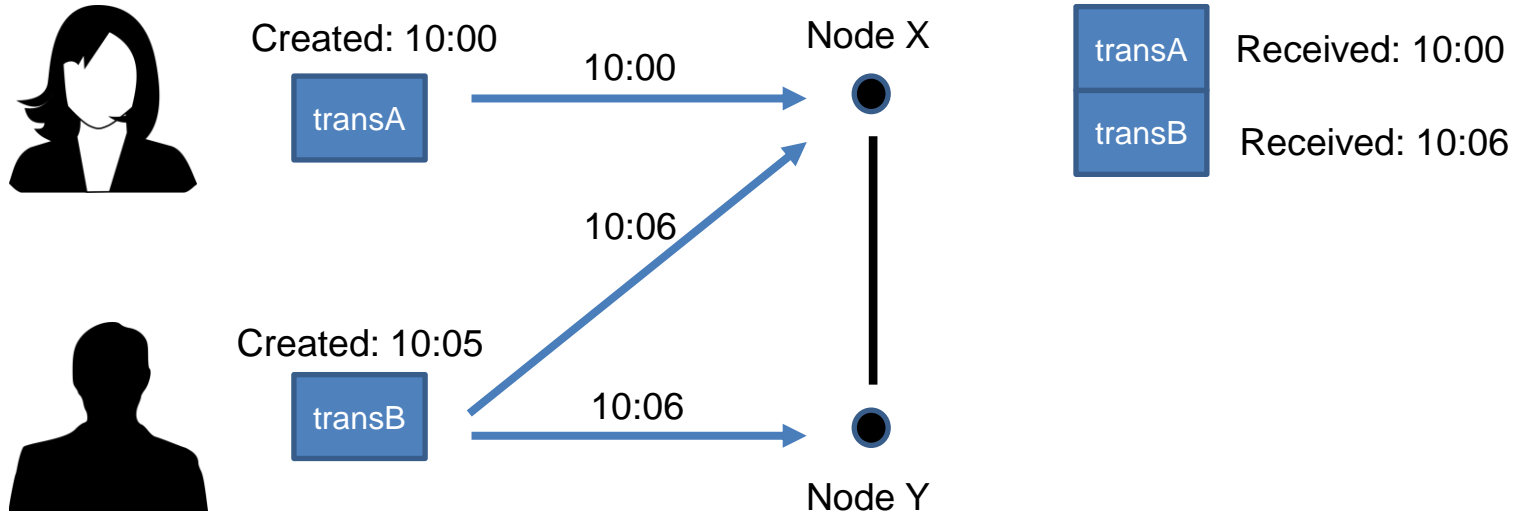
Global time



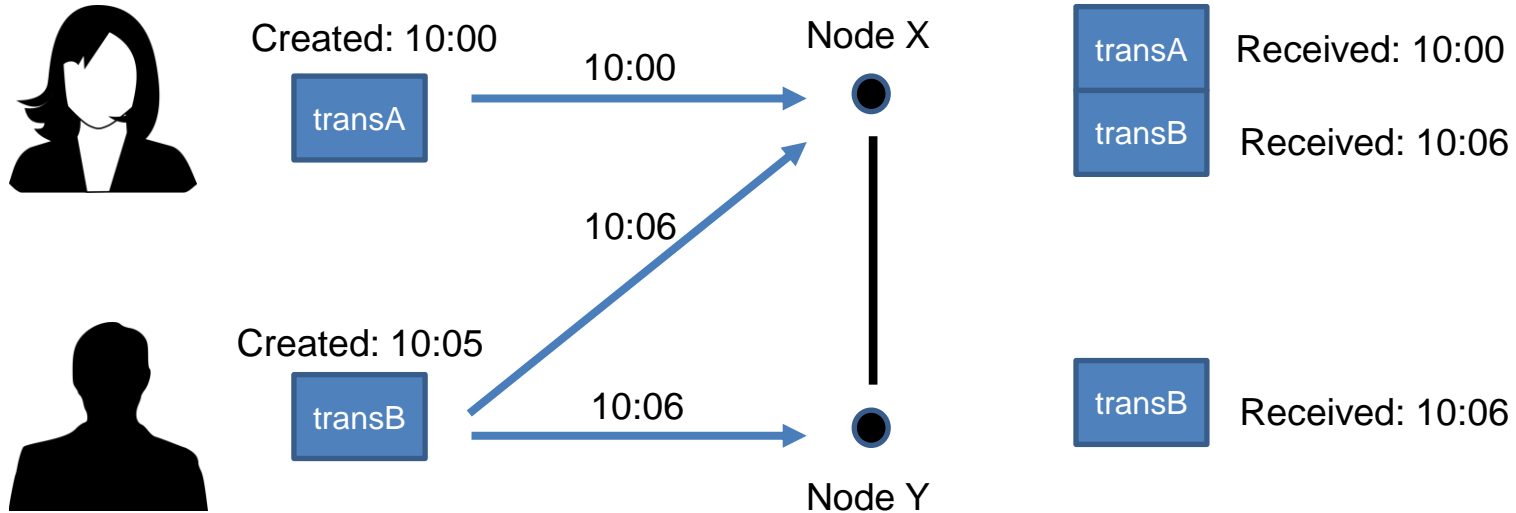
Global time



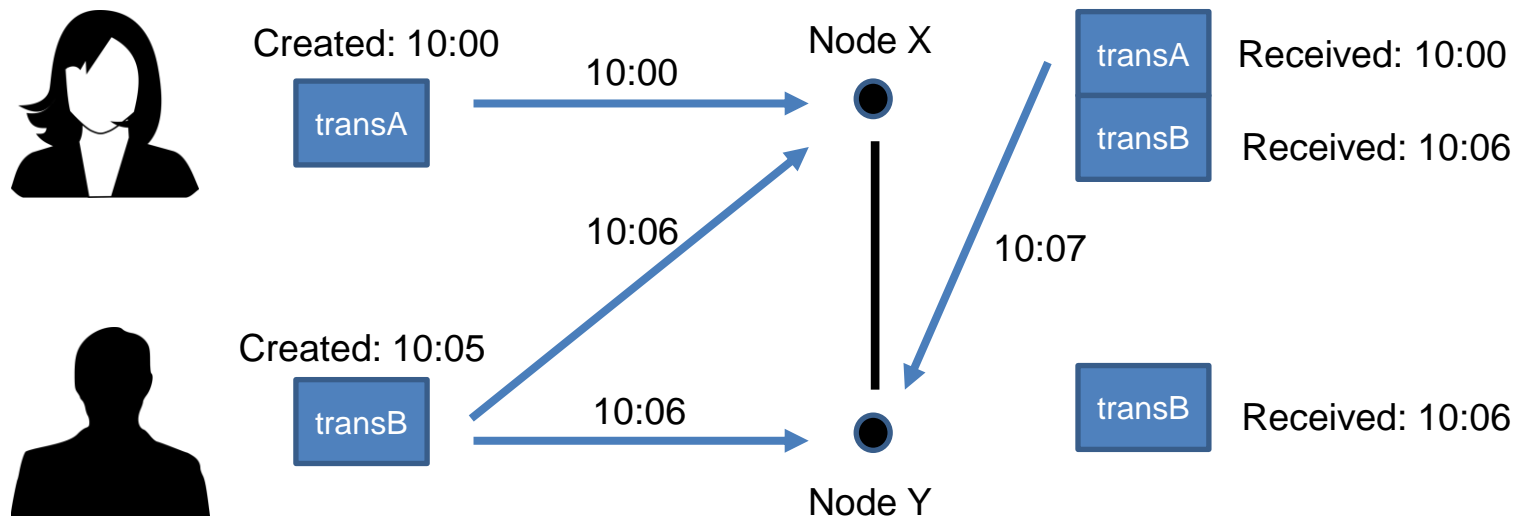
Global time



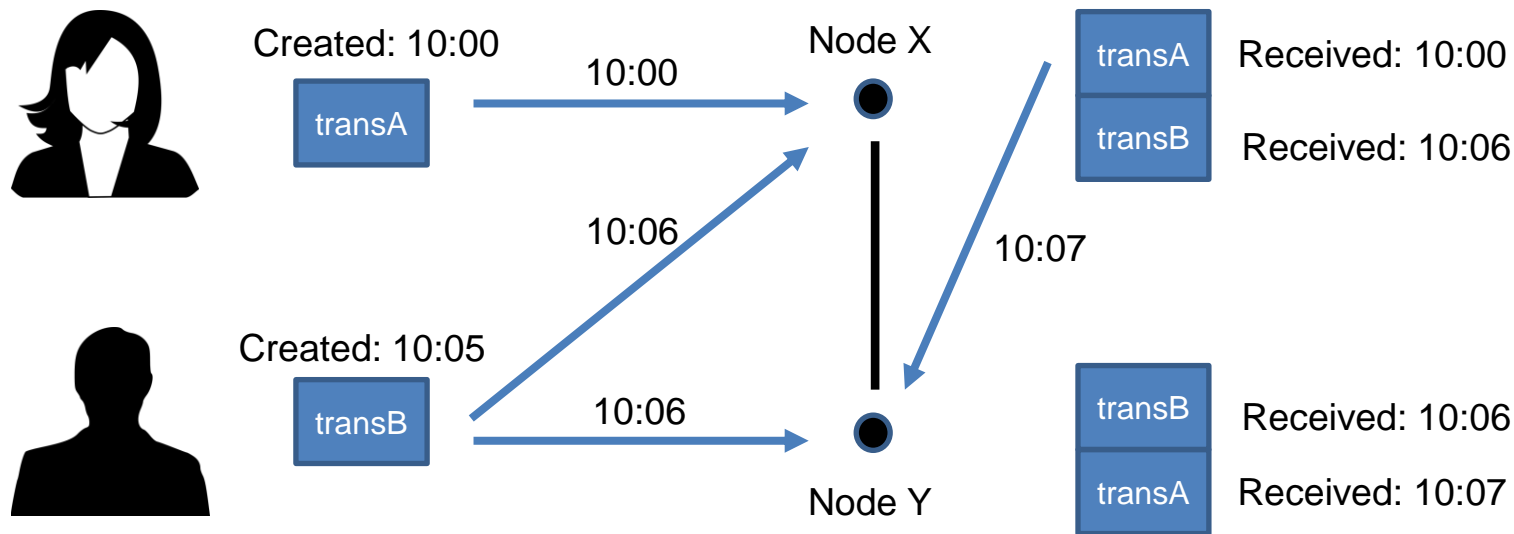
Global time



Global time



Global time



Nodes do not have IDs

By design, nodes in the network do not have IDs

Sybil attack = Create 1000 malicious nodes to attack the network

Since there are no IDs, we can not remove nor punish bad nodes

We can also not estimate how many bad nodes there are

So how do we achieve the consensus?

The consensus does not exist / impossible to reach:

- Byzantine generals (with one third of malicious nodes)
- Fischer-Lynch-Paterson (with a single malicious node)

Paxos:

- Always consistent, but can get into an infinite loop

In practice

In Bitcoin the consensus protocol works perfectly!!!

What is going on here?

The impossibility results are for databases:

- This is a very specific model

How is Bitcoin different?

- Economic incentives (consensus only for cryptocurrencies)
- Randomness (consensus evolves with time)

The theory of Bitcoin consensus is still not completely developed.

Work in progress

Shameless plug

Cryptocurrency Mining Games with Economic Discount and Decreasing Rewards

Marcelo Arenas

PUC Chile & IMFD Chile, Santiago, Chile
marenas@ing.puc.cl

Juan Reutter

PUC Chile & IMFD Chile, Santiago, Chile
jreutter@ing.puc.cl

Etienne Toussaint

University of Edinburgh, Edinburgh, UK
etienne.toussaint@ed.ac.uk

Martín Ugarte

PUC Chile & IMFD Chile, Santiago, Chile
martin@martinugarte.com

Francisco Vial

ProtonMail & IMFD Chile, Santiago, Chile
fvial@pm.me

Domagoj Vrgoč

PUC Chile & IMFD Chile, Santiago, Chile
dvrgoc@ing.puc.cl

STACS 2020

Work in progress

Cryptocurrency Mining Games with and Decreasing Rewards

enas
FD Chile, Santiago, Chile
e.cl
or



fvial@p
Doma
PUC C
dvrgoc@



STACS 2020

Implicit consensus

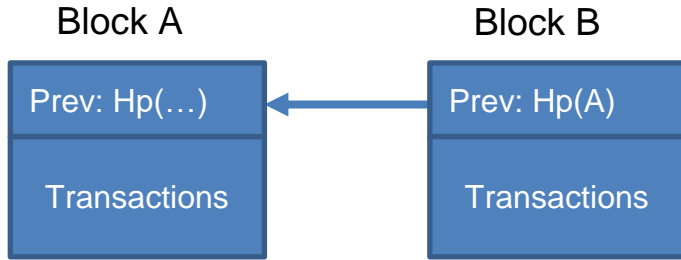
Suppose that we can randomly select a node and avoid the sybil attack at the same time (i.e. We can detect replicated nodes).

The protocol:

1. Nodes listen for/receive new transactions
2. Each node groups transactions into a block
3. Each 10 minutes we **randomly** select a node to propose its block
4. Other nodes will accept the block only if it is valid
5. They express the acceptance by extending their blockchain with this block

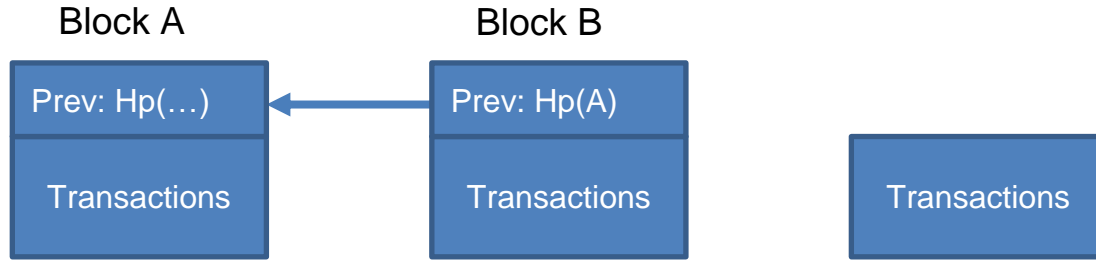
Implicit consensus

Accepting Block X



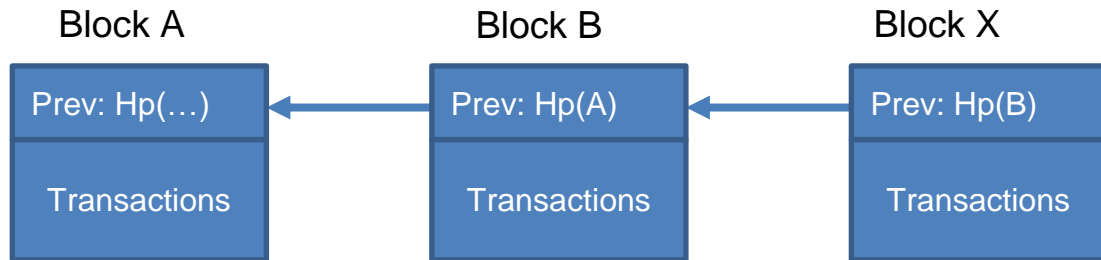
Implicit consensus

Accepting Block X



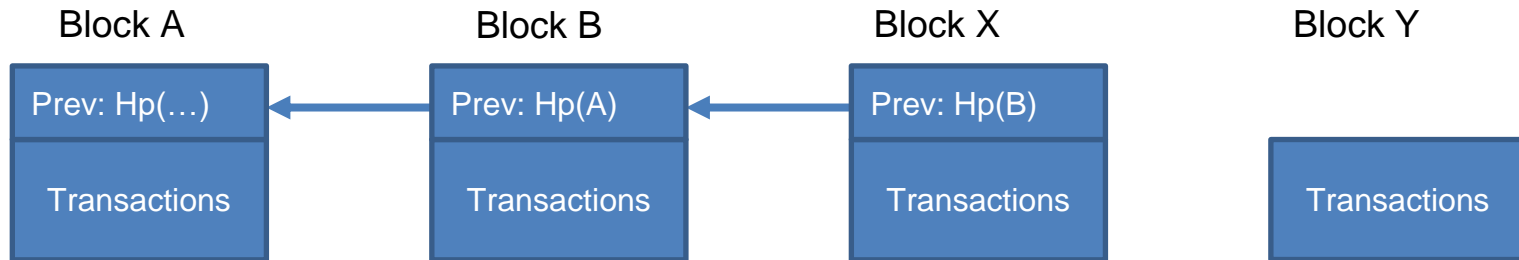
Implicit consensus

Accepting Block X



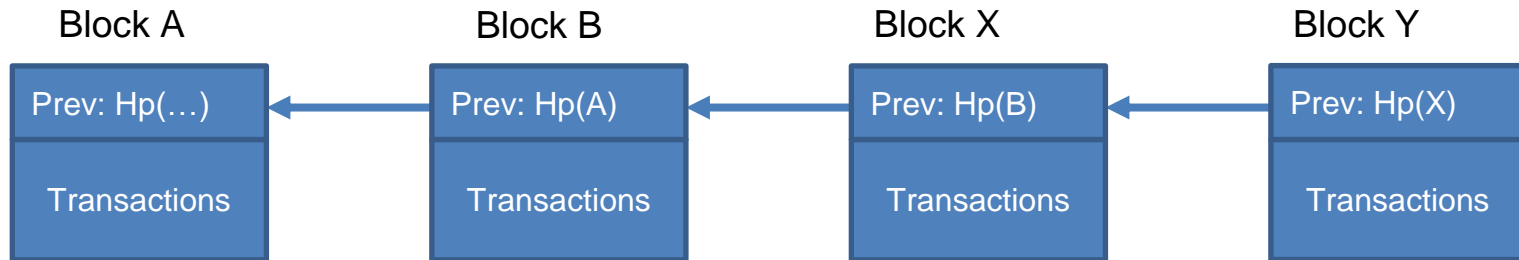
Implicit consensus

Accepting Block X



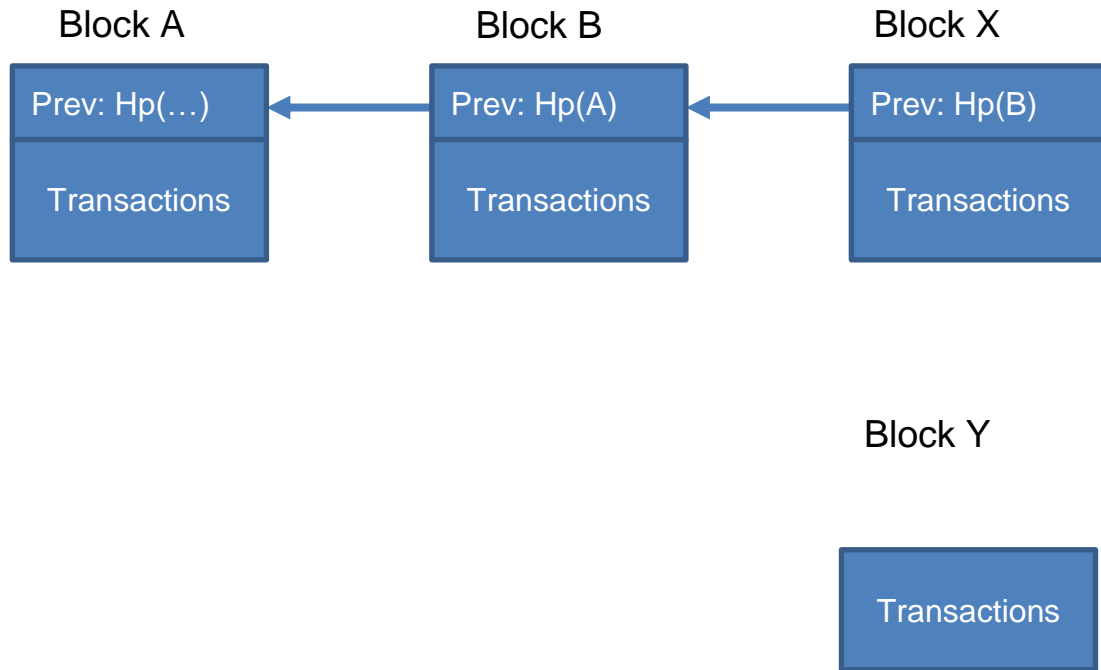
Implicit consensus

Accepting Block X



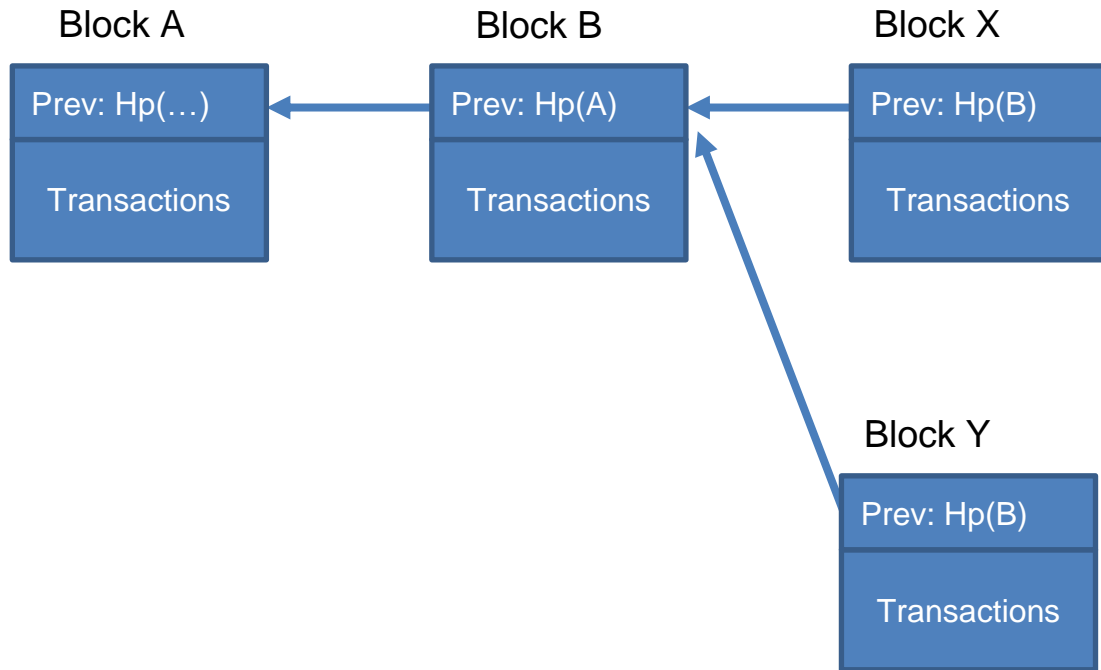
Implicit consensus

Rejecting Block X



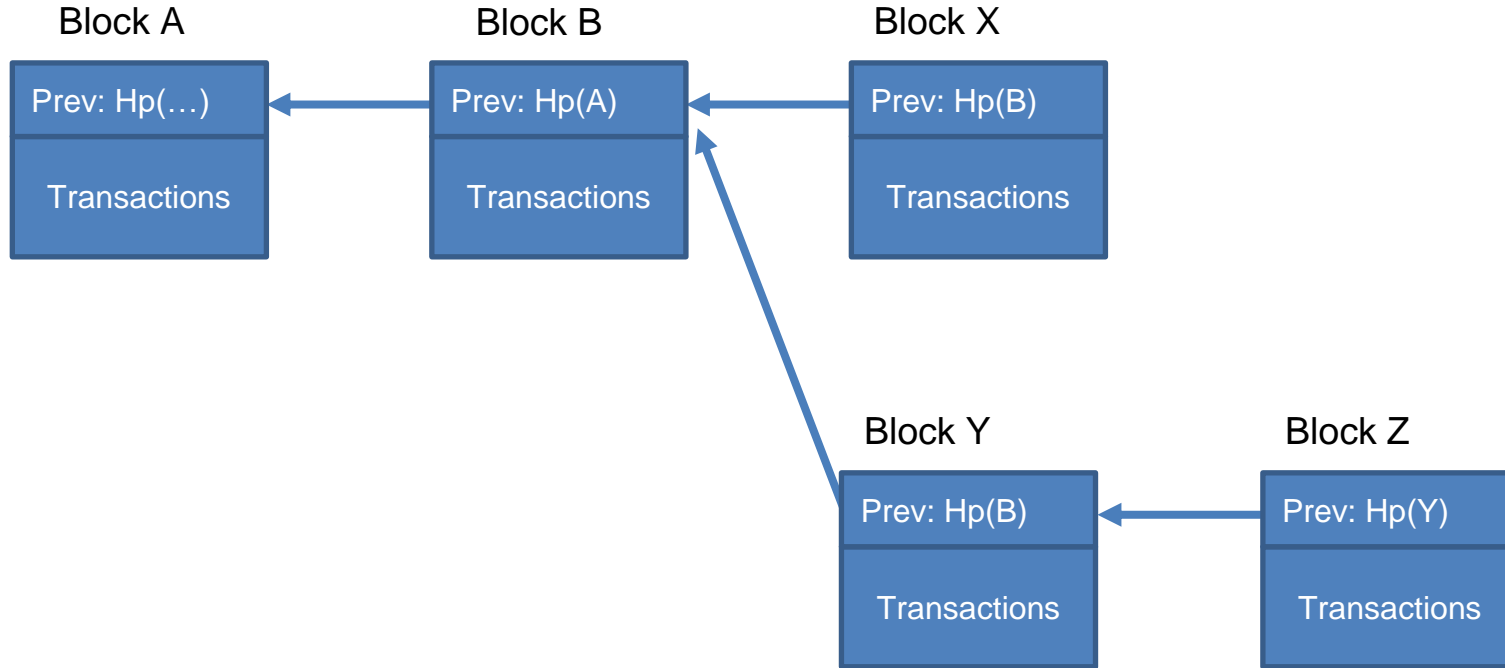
Implicit consensus

Rejecting Block X



Implicit consensus

Rejecting Block X



Implicit consensus

Nodes accept/reject Block X implicitly:

- They do not declare this publicly
- They accept the block by extending their blockchain with this block
- They reject it by not adding it to their blockchain (and extend the blockchain from an earlier block)

Recall how we implemented the blockchain using dictionaries!

Properties of implicit consensus

Can someone steal my Bitcoins in implicit consensus?

Properties of implicit consensus

Can someone steal my Bitcoins in implicit consensus?

- Nah
- If Alice proposes the next block, she should falsify my signature
- Easy to detect (signatures will not validate)
- In conclusion: even if a malicious node proposes the next block, it can not send all the funds to itself (other nodes will reject the block)

Properties of implicit consensus

Can a denial-of-service attack work?

- The selected node can prohibit a specific user to publish their transaction

Properties of implicit consensus

Can a denial-of-service attack work?

- The selected node can prohibit a specific user to publish their transaction
- Only works in the round where the malicious node proposes the block
- If the transaction is valid, a good node will include it in the next block
- There is no attack here (you just have to wait a bit)

Properties of implicit consensus

Can a double spend attack work?

- Bob is an online software vendor
- Bob accepts Bitcoin (once paid, Bob allows downloading the software)
- The question is: what is a Bitcoin payment?

Properties of implicit consensus

How can Alice execute the double-spend attack?

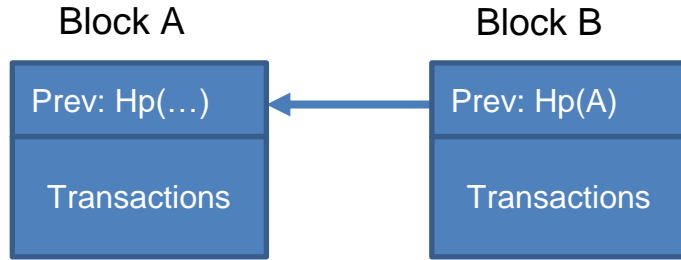
- Alice creates a transaction transferring the funds to Bob
- Alice transmits the transaction to the p2p network
- An honest node includes the transaction in the next block
- Bob allows Alice to download the software

Properties of implicit consensus

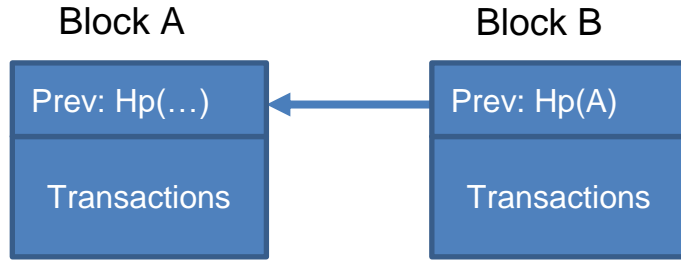
How can Alice execute the double-spend attack?

- Alice creates a transaction transferring the funds to Bob
 - Alice transmits the transaction to the p2p network
 - An honest node includes the transaction in the next block
 - Bob allows Alice to download the software
-
- What is Alice proposes the next block?
 - What can she do?

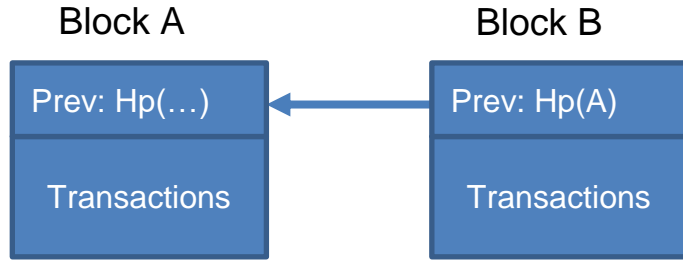
Double spend



Double spend



Double spend



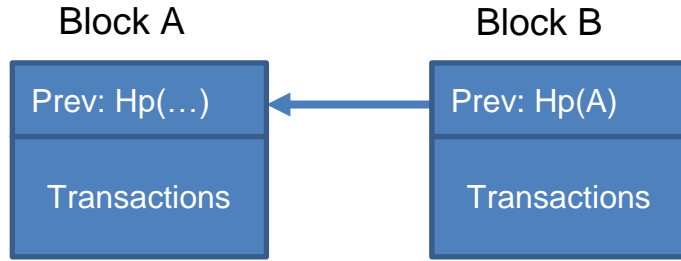
I want to buy
your software



OK, pay me,
and you can
download it



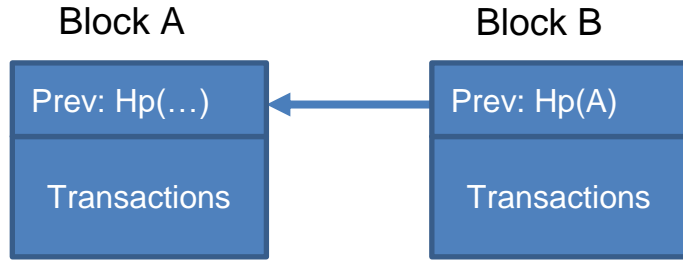
Double spend



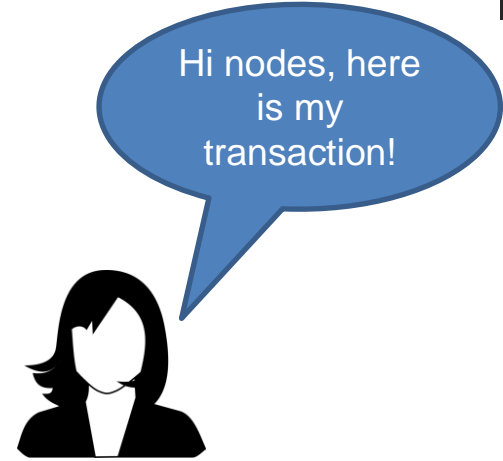
Alice - Bob



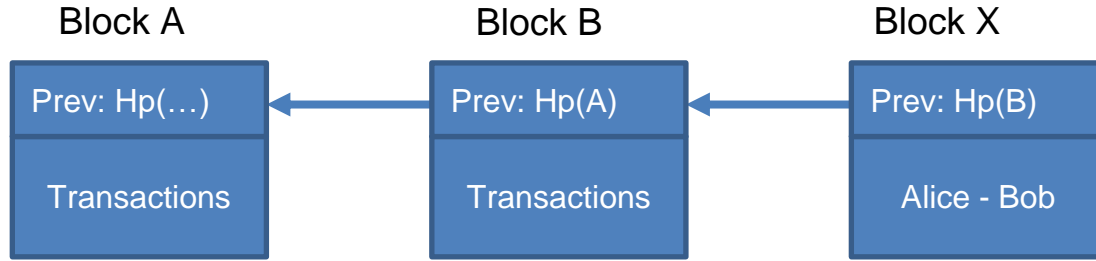
Double spend



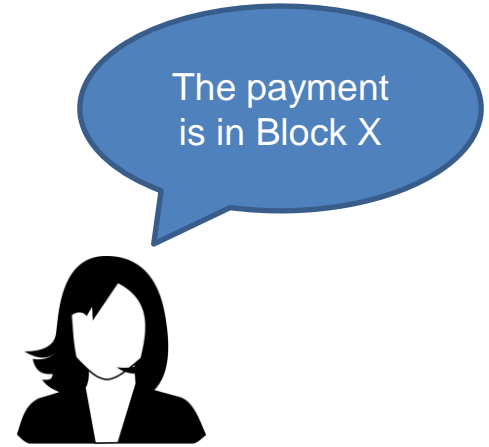
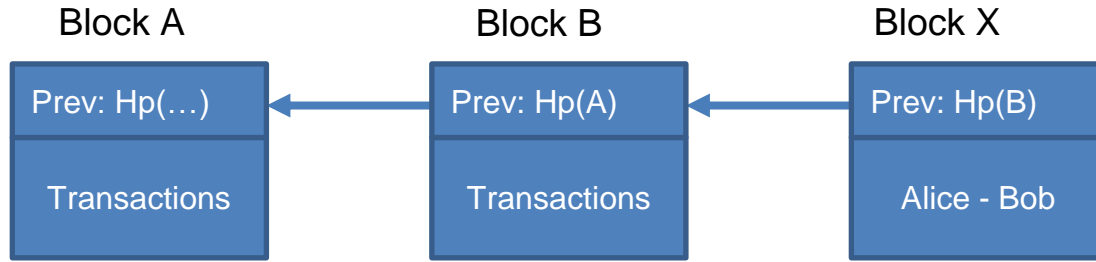
Alice - Bob




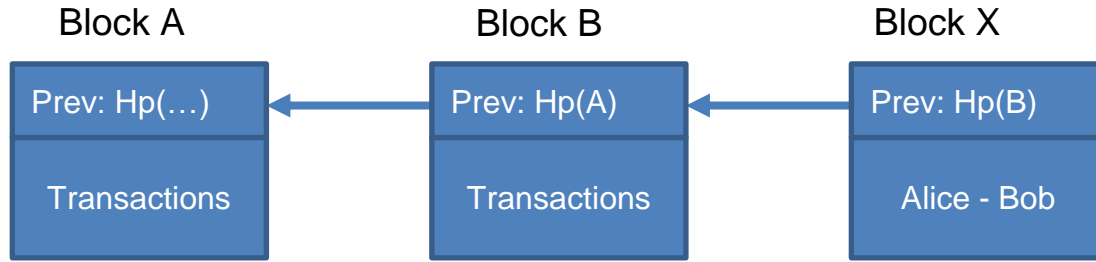
Double spend




Double spend



Double spend

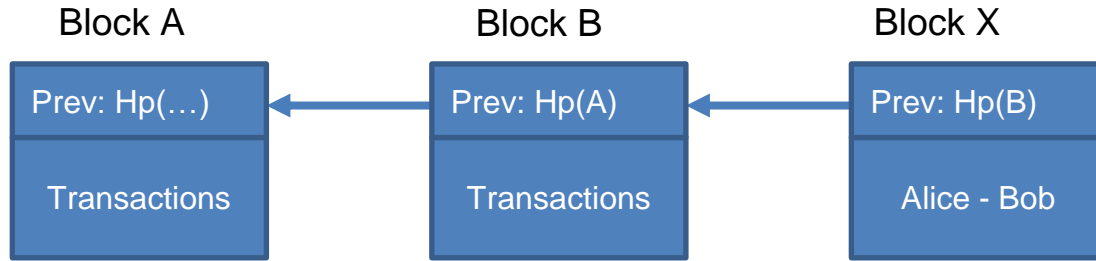


OK, you can
download the
software

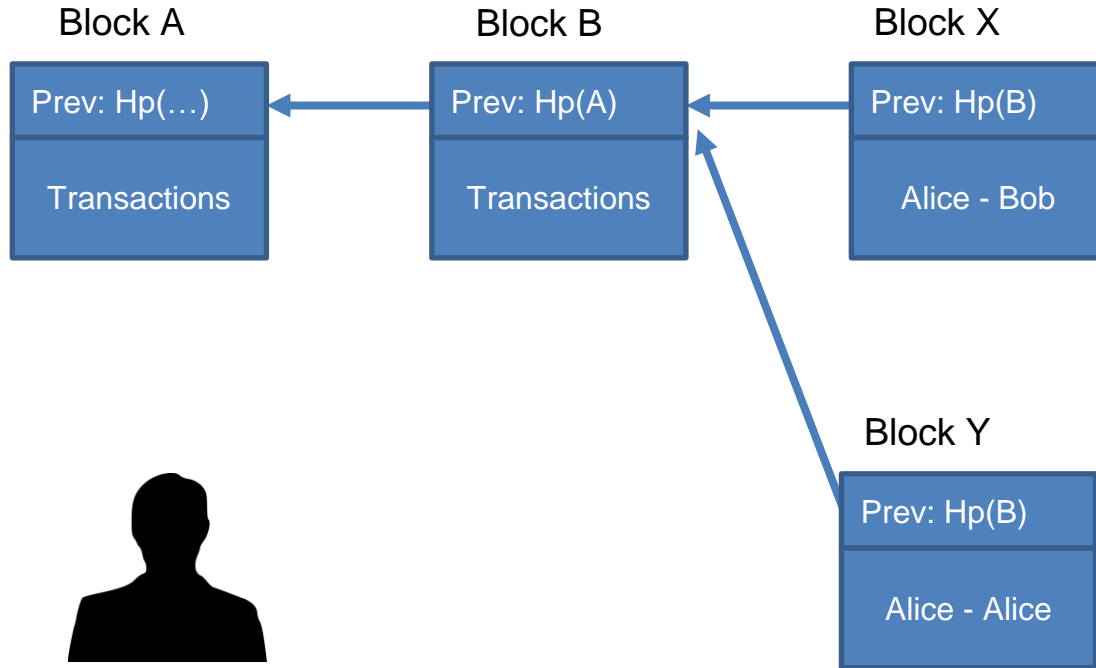


The payment
is in Block X

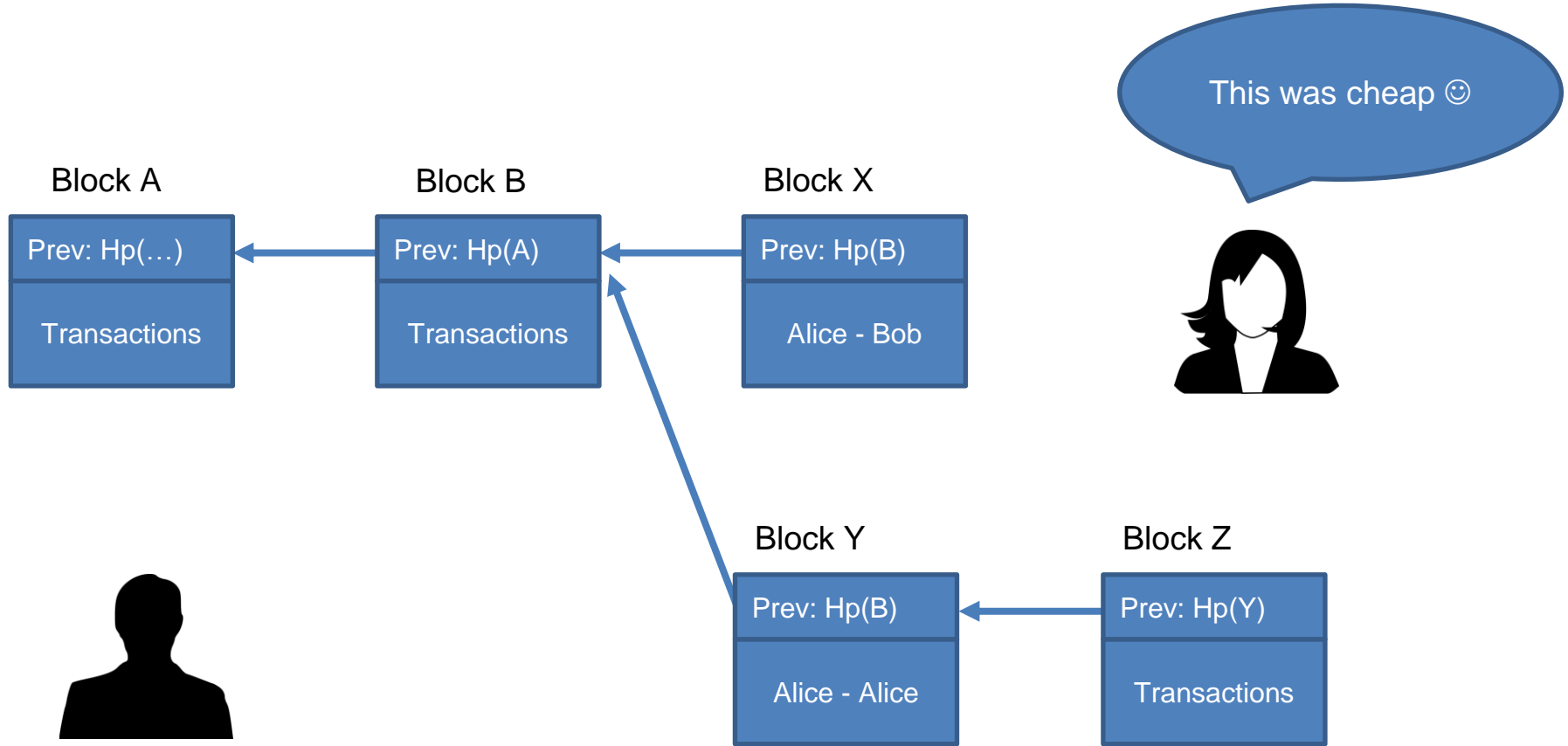
Double spend



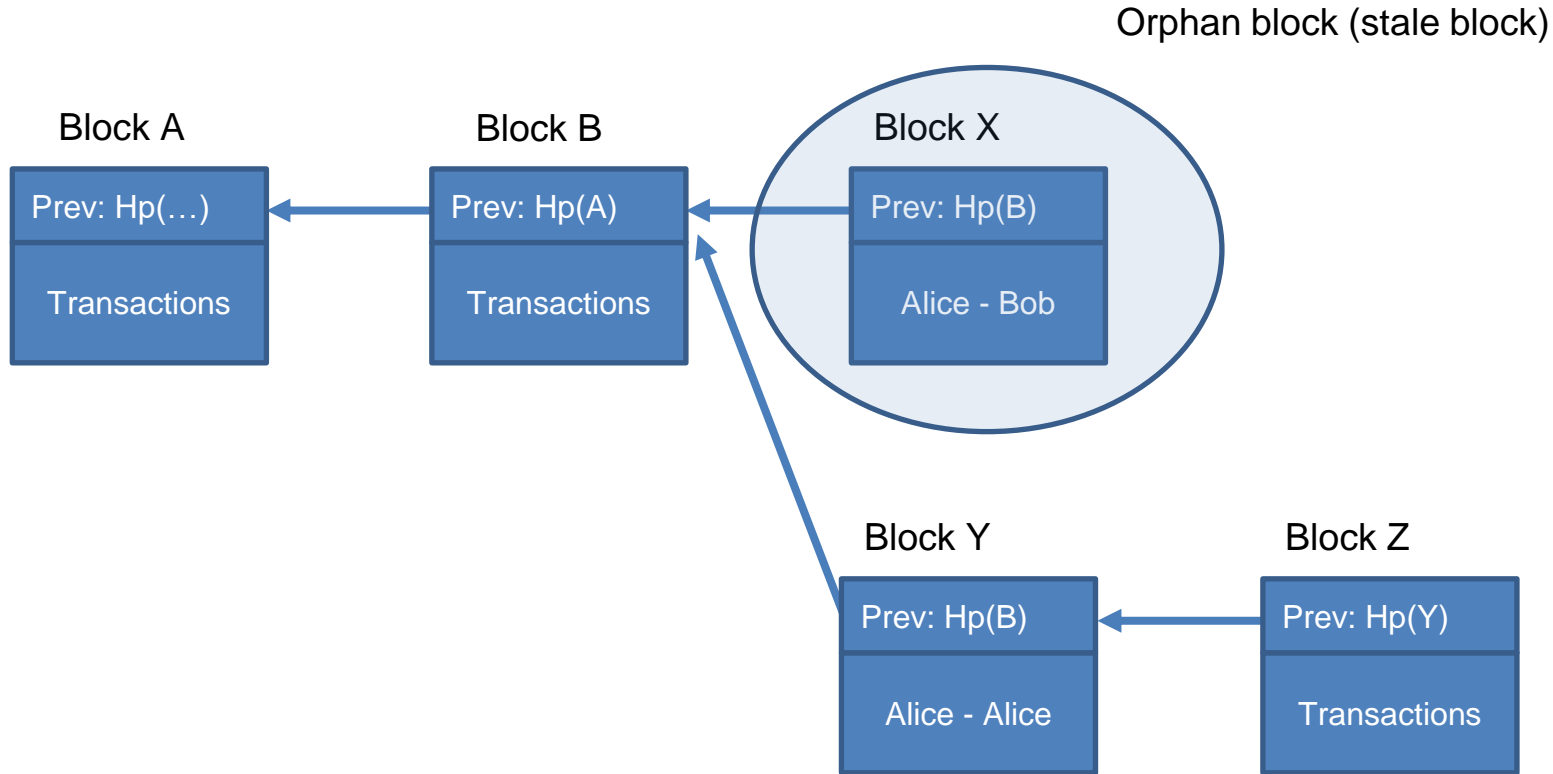
Double spend



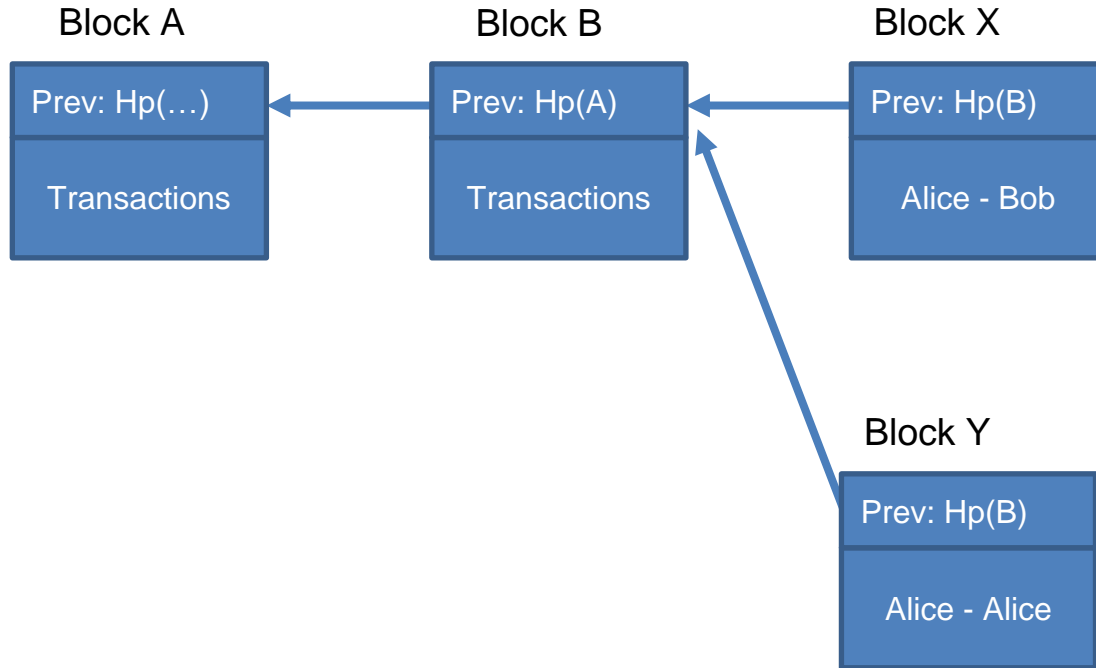
Double spend



Double spend



Double spend



Important:

- Morally X is correct
- Cryptographically it is the same
- Nodes do not know the story
- For them X and Y are both valid
- They can extend any one of them

Properties of implicit consensus

How can Alice execute the double-spend attack?

- Alice creates a transaction transferring the funds to Bob
- Alice transmits the transaction to the p2p network
- An honest node includes the transaction in the next block

Bob has various options:

- Zero confirmations (double-spend very likely)
- One confirmation (double-spend we just explained)
- k confirmations

Multiple confirmations

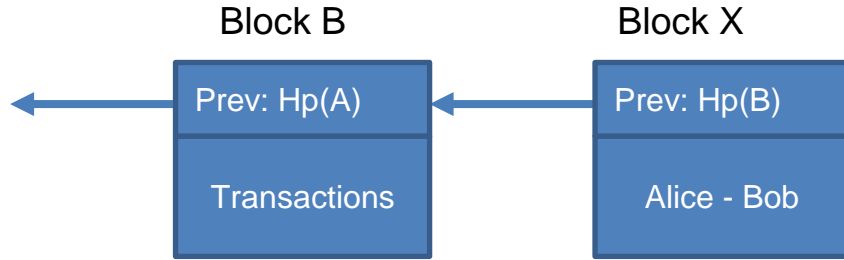
Block B

Prev: Hp(A)

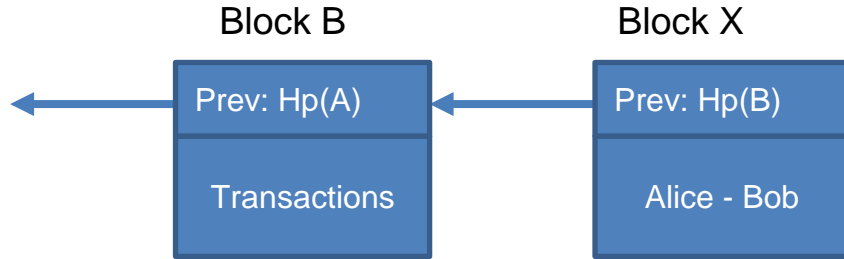
Transactions



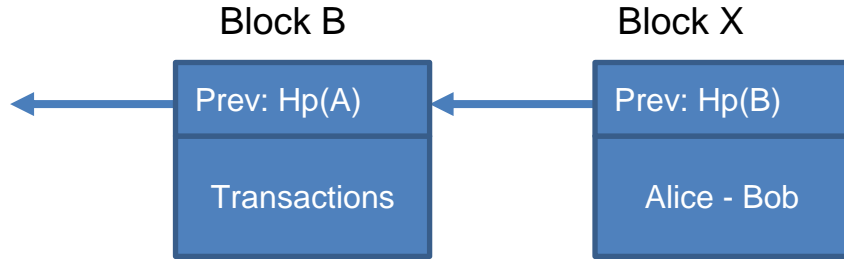
Multiple confirmations



Multiple confirmations



Multiple confirmations



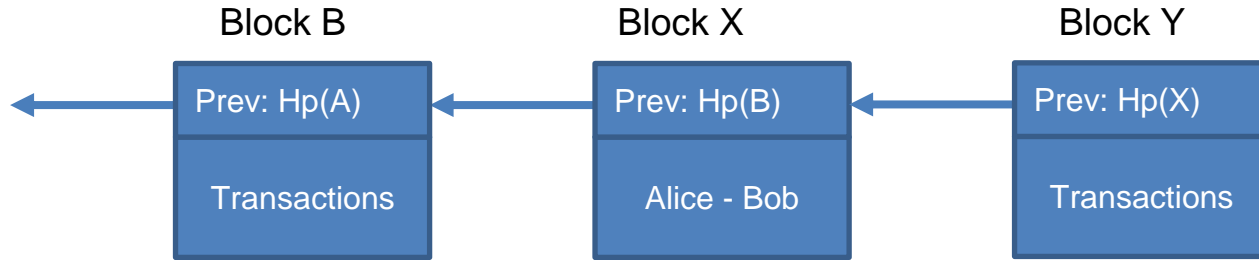
Let's wait for a
bit



The payment
is in Block X



Multiple confirmations



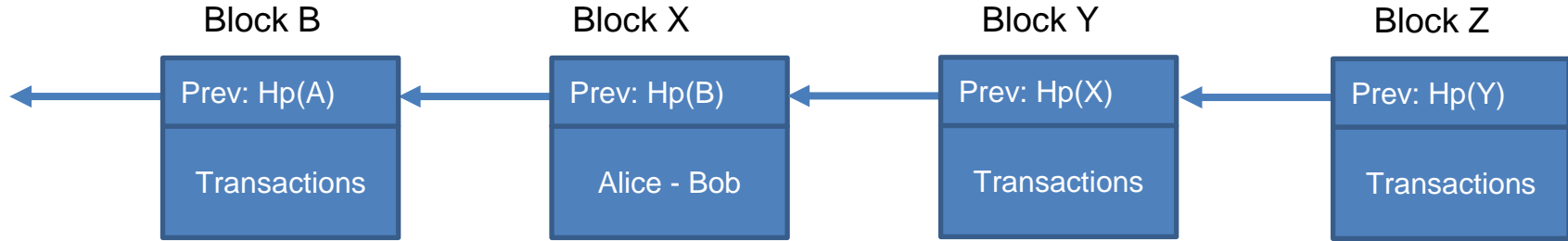
Let's wait for a
bit



The payment
is in Block X



Multiple confirmations



OK, now you
can download!



The payment
is in Block X



Properties of implicit consensus

Bob has various options:

- Zero confirmations (double-spend very likely)
- One confirmation (double-spend we just explained)
- k confirmations (6 for Bitcoin = cca 1h)

Important: 6 confirmations are no guarantee that the transaction will remain in the longest Blockchain. But with each new confirmation, the probability of losing the money drops exponentially. 6 is a good heuristic.

Properties of implicit consensus

Protections against invalid transactions:

- Completely cryptographic (the signature does not validate)
- Enforced by the consensus (good nodes do not include an invalid transaction)

Protection against double-spend:

- By consensus
- The cryptography is good in both branches, and one of them will win out

What is missing?

Implicit consensus assumes:

- That we can select a node randomly
- That at least 50% of the nodes are good

To remove these assumptions Bitcoin uses incentives (\$\$\$)

What is missing?

Implicit consensus assumes:

- That we can select a node randomly
- That at least 50% of the nodes are good

To remove these assumptions Bitcoin uses incentives (\$\$\$)

How to punish malicious nodes?

What is missing?

Implicit consensus assumes:

- That we can select a node randomly
- That at least 50% of the nodes are good

To remove these assumptions Bitcoin uses incentives (\$\$\$)

~~**How to punish malicious nodes?**~~

How to reward good nodes?

Incentives in Bitcoin

How to reward good nodes?

- **With Bitcoins**

Two types of incentives

- 1. Block reward**
- 2. Transaction fees**

Block rewards

Each block in Bitcoin includes a special transaction:

- "Coinbase transaction" == CreateCoins in Scroogecoin
- Creates new Bitcoins and sends them to the block creator
- Reward for adding a new block

Block rewards

Each block in Bitcoin includes a special transaction:

- "Coinbase transaction" == CreateCoins in Scroogecoin
- Creates new Bitcoins and sends them to the block creator
- Reward for adding a new block

Can a node cheat?

Block rewards

Each block in Bitcoin includes a special transaction:

- "Coinbase transaction" == CreateCoins in Scroogecoin
- Creates new Bitcoins and sends them to the block creator
- Reward for adding a new block

Can a node cheat?

- Include a bad block and still receive the reward?

Block rewards

Each block in Bitcoin includes a special transaction:

- "Coinbase transaction" == CreateCoins in Scroogecoin
- Creates new Bitcoins and sends them to the block creator
- Reward for adding a new block

Can a node cheat?

- Include a bad block and still receive the reward?
- No, Coinbase only if it has 100 confirmations
- If the block is bad, the rest of the network will not extend on top of this block
- Incentivizes extending the longest blockchain (and a valid one)

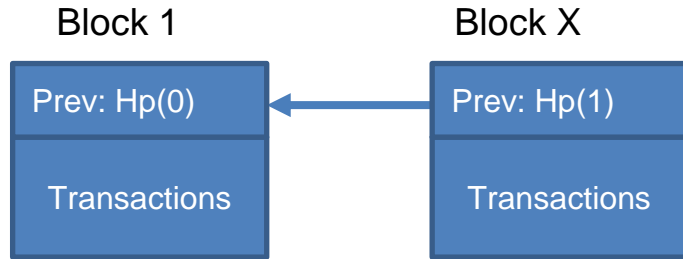
Block reward

Block 1

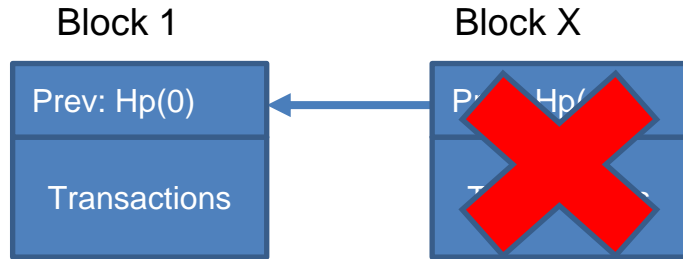
Prev: Hp(0)

Transactions

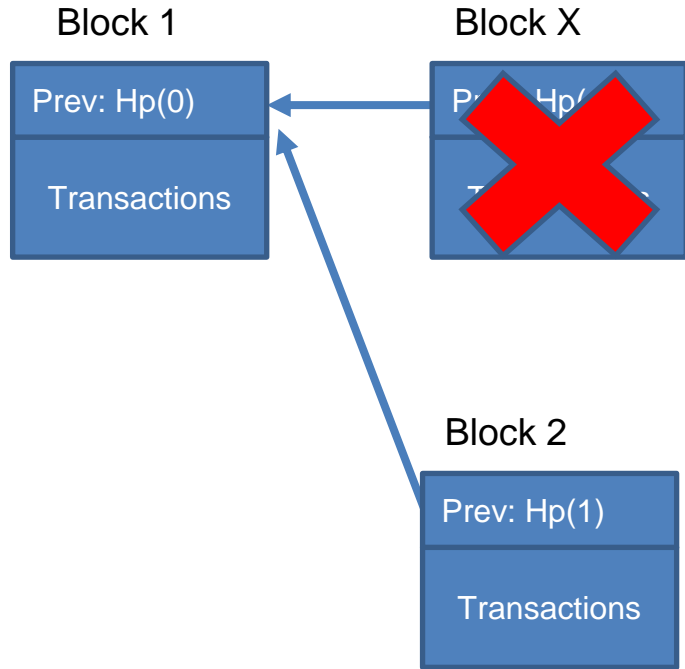
Block reward



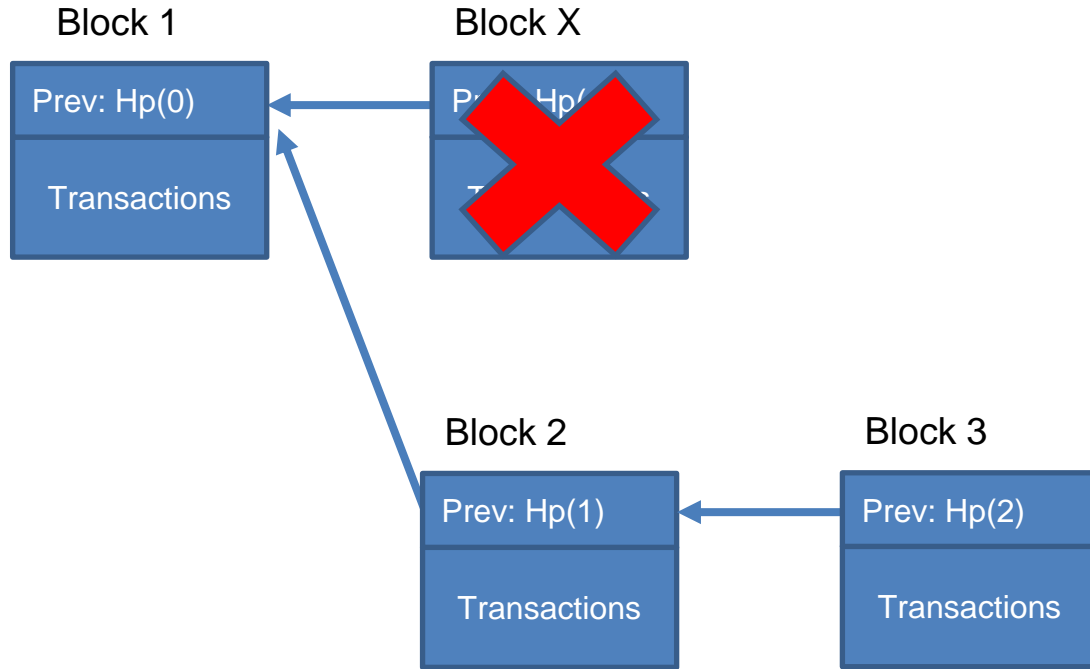
Block reward



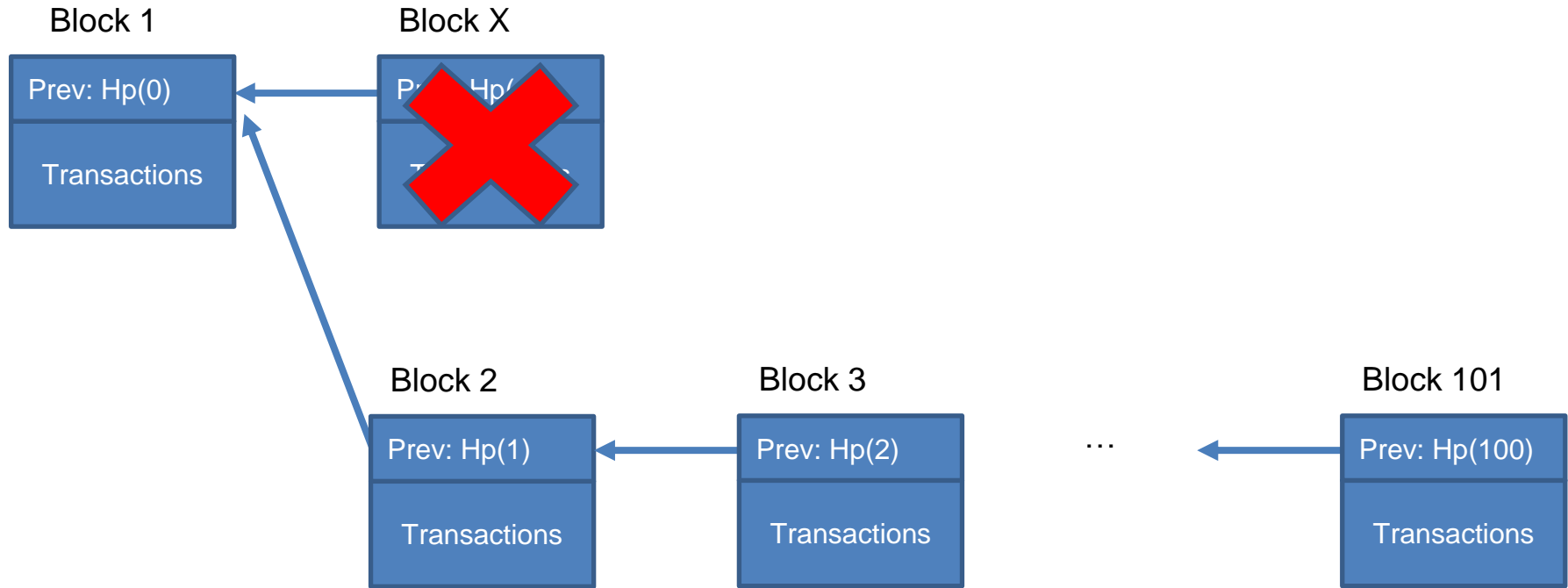
Block reward



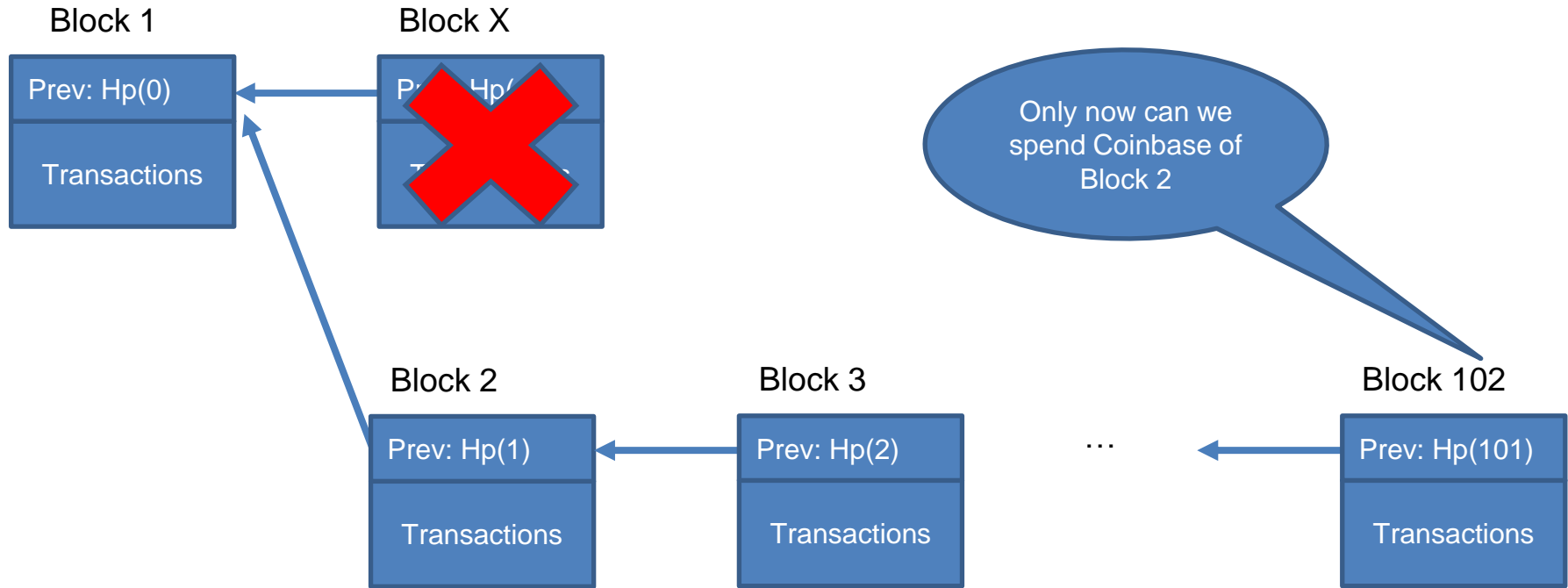
Block reward



Block reward



Block reward



Block rewards

The idea:

- Incentivizes having a single blockchain (no forks)
- The longest blockchain (that is valid) is the king

Block rewards

The block reward is fixed:

- At this point 6.25 Bitcoins
- Divided by 2 each 210.000 blocks (cca. 4 years)
- Started with 50BTC, reduced 3 times: to 25, 12.5, and 6.25 Bitcoins

Coinbase transaction:

- The only way to create new Bitcoins
- This is the emission of Bitcoins (printing money)
- Decentralized process (the network emits new Bitcoins in each block)

There is a finite amount of Bitcoin:

- 1 Bitcoin contains 10^8 Satoshi
- The smallest denomination of BTC = 0.00000001 BTC
- The reward diminishes by 50% every 4 years

- There will be a total of cca. 21 million Bitcoins
- The Block reward will cease to exist in year 2140

Block rewards

Why is it good that there is only a finite amount of Bitcoins?

Block rewards

Why is it good that there is only a finite amount of Bitcoins?

- Controls the inflation
- A scarce resource (if everyone can emit new Bitcoins, they will have no value)

Block reward ends in 2140!!!

Transaction fees

Sum of input value in a transaction can be $<$ sum of output values

- The difference is called "**transaction fee**"
- It is paid to the address specified in the Coinbase transaction
- The idea is that this is a tip for the node proposing the next block
- In order to assure quality of service
- All of the transaction fees in the block go to the miner

Transaction fees

This can cause problems

- The transaction with higher transaction fee is more likely to be included first

Transaction fees

This can cause problems

- The transaction with higher transaction fee is more likely to be included first
- Basic double spend: $A \rightarrow B$ fee: 0.01BTC; $A \rightarrow A$ fee 0.1BTC

In bitcoin each block has a capacity of 1MB:

- Size of the transaction vs. fee
- Number of transactions vs. fee
- A transaction with a gigantic fee
- How to design the next block to maximize the fee?

Still unresolved

1. **How to select the node that proposes the next block?**
2. **How to make sure that all the nodes are doing something useful?**
3. **How to prevent sybil-attack?**

Distributed consensus in Bitcoin

Mining via proof of work

Implicit consensus

Suppose that we can randomly select a node and avoid the sybil attack at the same time (i.e. We can detect replicated nodes).

The protocol:

1. Nodes listen for/receive new transactions
2. Each node groups transactions into a block
3. Each 10 minutes we **randomly** select a node to propose its block
4. Other nodes will accept the block only if it is valid
5. They express the acceptance by extending their blockchain with this block

Proof of work

We will simulate selecting a node randomly by:

- Using a resource that can not be monopolized (easily)
- We will select a node based on its proportion of this resource

The resource used in Bitcoin:

- Computational power (number of hashes per second)
- The nodes will be selected based on the amount of work they do

Proof of work

We will simulate selecting a node randomly by:

- Using a resource that can not be monopolized (easily)
- We will select a node based on its proportion of this resource

The resource used in Bitcoin:

- Computational power (number of hashes per second)
- The nodes will be selected based on the amount of work they do

Resolves the sybil attack

Proof of work

We will use hash puzzles:

- All the nodes share a difficulty called *target*
- The nodes look for a *nonce* s.t.

$$H(\text{nonce} || \text{prev_hash} || \text{tx_1} || \text{tx_2} || \dots || \text{tx_n}) < \text{target}$$

The first node to find the *nonce* published the next block!!!

Proof of work

With hash puzzles

- It is not necessary to randomly select a node
- Randomness is simulated using hash puzzles
- The competition does not depend on a central entity
- No one decides the winner, it is a fair competition (since hash is almost random)

Nodes competing to find the nonce are called **miners**

The process of solving the hash puzzle = **Bitcoin mining**

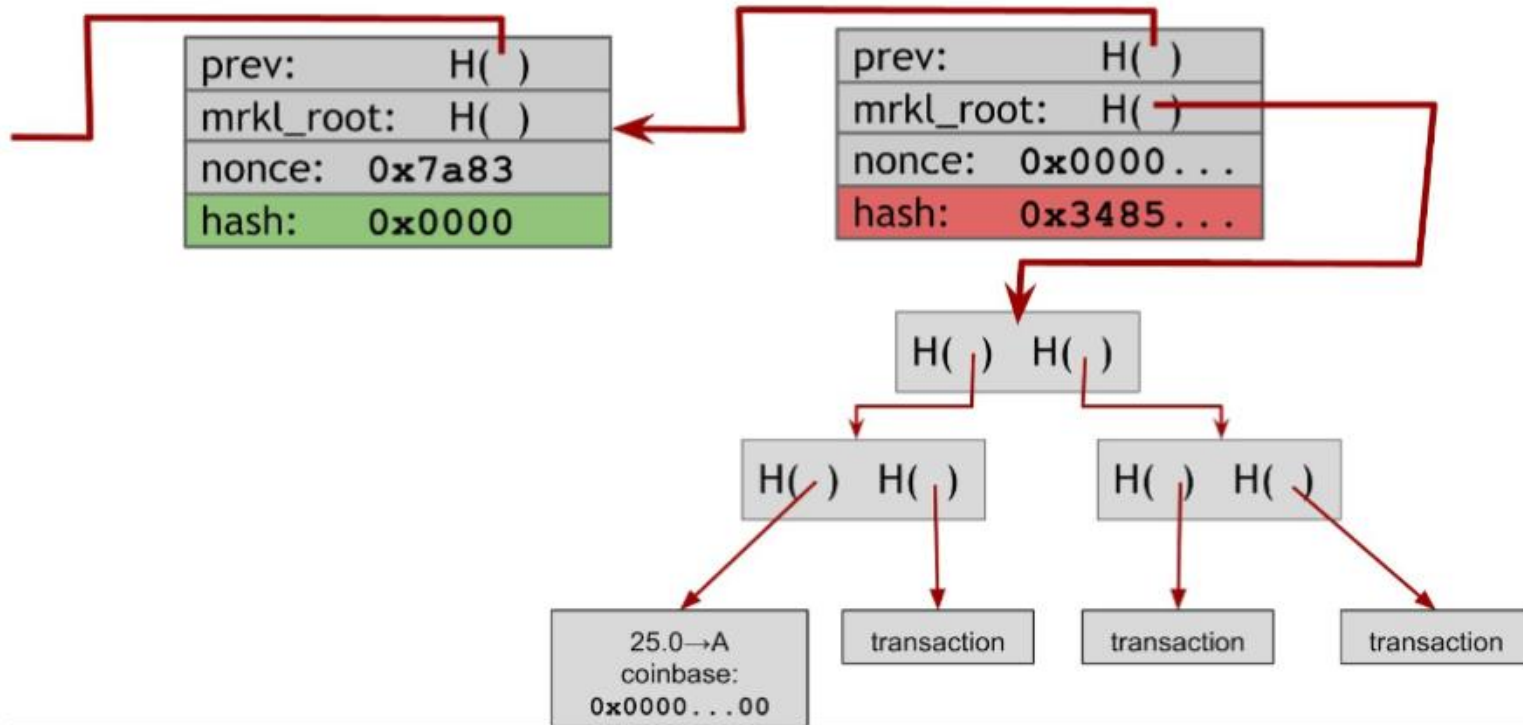
Proof of work

What is really hashed in Bitcoin?

- **Block header**

Field	Purpose	Updated when...	Size (Bytes)
Version	Block version number	You upgrade the software and it specifies a new version	4
hashPrevBlock	256-bit hash of the previous block header	A new block comes in	32
hashMerkleRoot	256-bit hash based on all of the transactions in the block	A transaction is accepted	32
Time	Current timestamp as seconds since 1970-01-01T00:00 UTC	Every few seconds	4
Bits	Current target in compact format	The difficulty is adjusted	4
Nonce	32-bit number (starts at 0)	A hash is tried (increments)	4

How does this work?



Proof of work

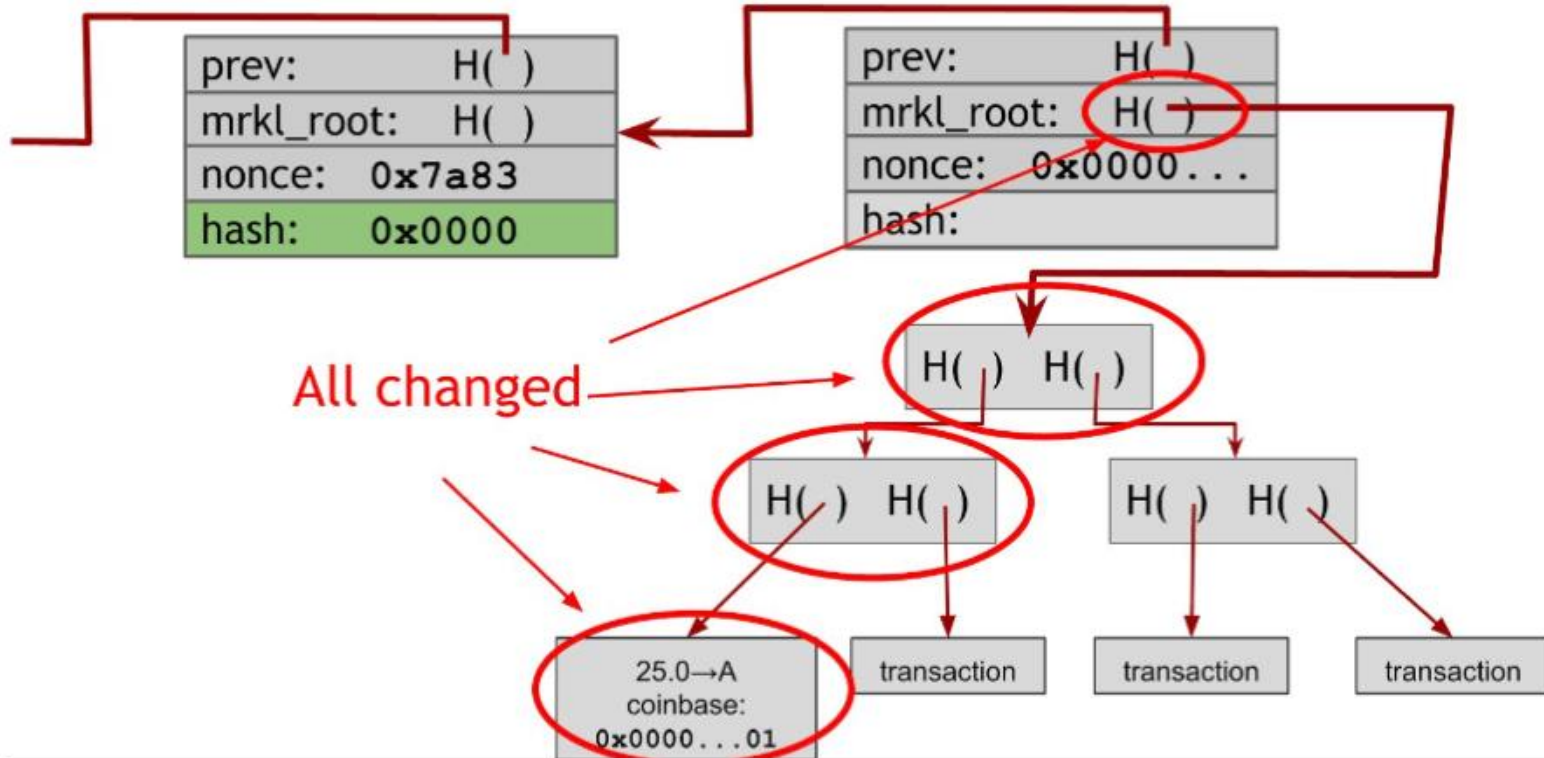
What is really hashed in Bitcoin?

- **Nonce has 32 bits!!!**
- **Target (currently) requires on average 2^{75} bits!!!**
- **The field extra nonce in Coinbase!!!**

Difficulty in block header is a bit different!!!

How does this work?

Updating the extra nonce



The difficulty is adjusted automatically:

- Recall: new block each 10 minutes
- If there are more miners, the blocks will be found quicker
- There is a difficulty readjustment every 2016 blocks

$$nextTarget = \frac{currentTarget \cdot 2016 \cdot 10 \text{ min}}{time \text{ to mine last 2016 blocks}}$$

Proof of work

Why 10 minutes?

- No particular reason
- Everyone is in agreement it should be a fixed amount of time
- That can not be lowered arbitrarily

Proof of work

Easy to verify:

- Finding the *nonce* is difficult
- Verifying that a *nonce* is valid takes computing one hash (easy)
- Once published, the validity of a block is easy to verify

Consensus in Bitcoin

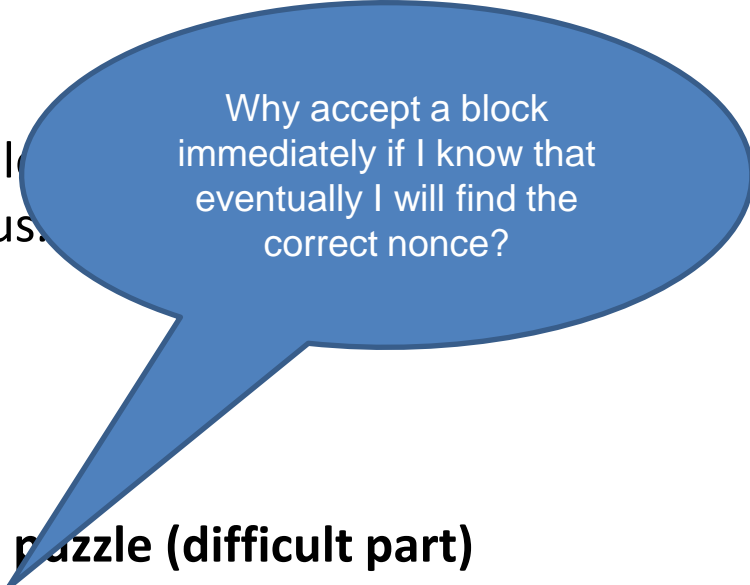
P2P network. Imperfect network, nodes join and leave. Messages do not propagate instantly (delay). Nodes are anonymous.

The protocol:

1. Nodes listen for/receive new transactions
2. Each node groups transactions into a block
3. **Nodes look for a nonce that solves the hash puzzle (difficult part)**
4. **Node that finds the nonce first publishes the next block**
5. Other nodes will accept the block only if it is valid (**easy to check**)
6. Accepting a block is expressed by extending the blockchain with this block

Consensus in Bitcoin

P2P network. Imperfect network, nodes join and leave, propagate instantly (delay). Nodes are anonymous.



Why accept a block immediately if I know that eventually I will find the correct nonce?

The protocol:

1. Nodes listen for/receive new transactions
2. Each node groups transactions into a block
3. **Nodes look for a nonce that solves the hash puzzle (difficult part)**
4. **Node that finds the nonce first publishes the next block**
5. Other nodes will accept the block only if it is valid (**easy to check**)
6. Accepting a block is expressed by extending the blockchain with this block

Consensus in Bitcoin

Security:

- More than 50% of the nodes are honest

(The value was generated by an honest node)

Percentage of hash power = percentage of mined blocks

- If Alice has 0.1% of hash power, she will mine 1 block in each 1000 new blocks
- Long term

- If Alice has 1000 more hash power than Bob
- Alice = 99.9%, Bob = 0.01%
- Bob will discover one new block each 10000, Alice the rest
- It is not true that Alice will always win!!!!

Miners:

- Win money in terms of Bitcoin
- But they have hardware costs
- And the electricity costs

One expects an equilibrium:

- If miners do not win money, they will leave (the remaining ones will win more)
- If it is profitable to mine Bitcoin, more people will do it (making it more difficult/secure)

Consensus is the king

Alice has 10 BTC:

- Signifies that the network is in agreement that the addresses owned by Alice have a total of 10 unspent BTC
- The consensus can change in the future, but this is very unlikely if the transactions have already been confirmed many times

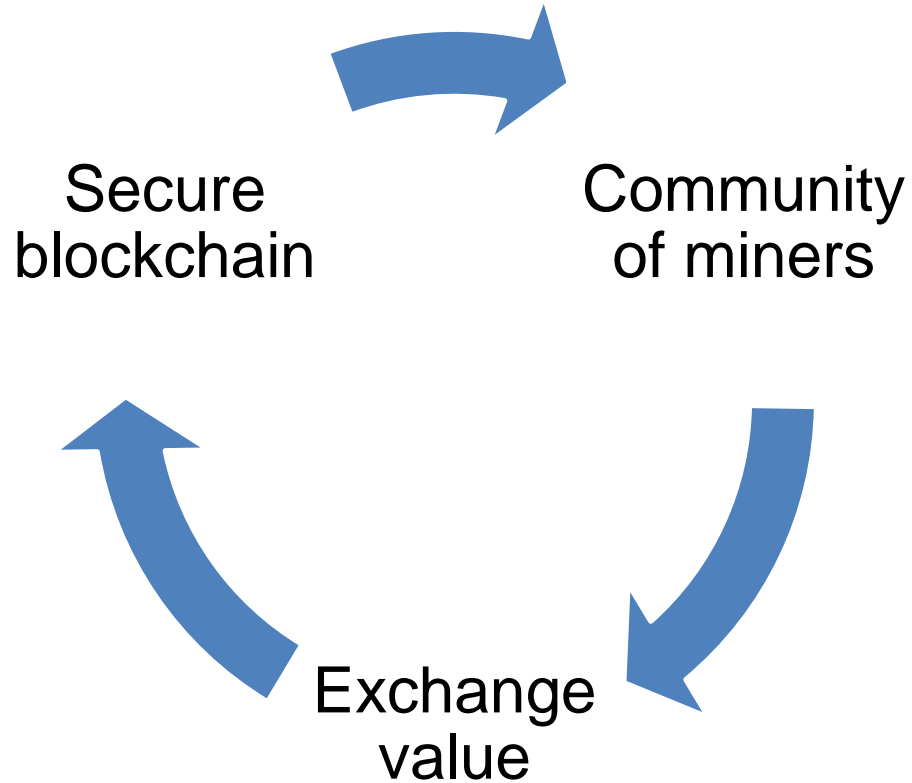
Bootstrapping

What does a cryptocurrency need to be successful

- A secure blockchain
- A healthy community of miners
- Value in USD

When do these things exist?

Bootstrapping



Bootstrapping

When Bitcoin started:

- At the beginning there was only Satoshi
- Blockchain had only one person mining
- Any person with fast computers could have attacked the network
- It had no value in USD

Why did Bitcoin succeed?

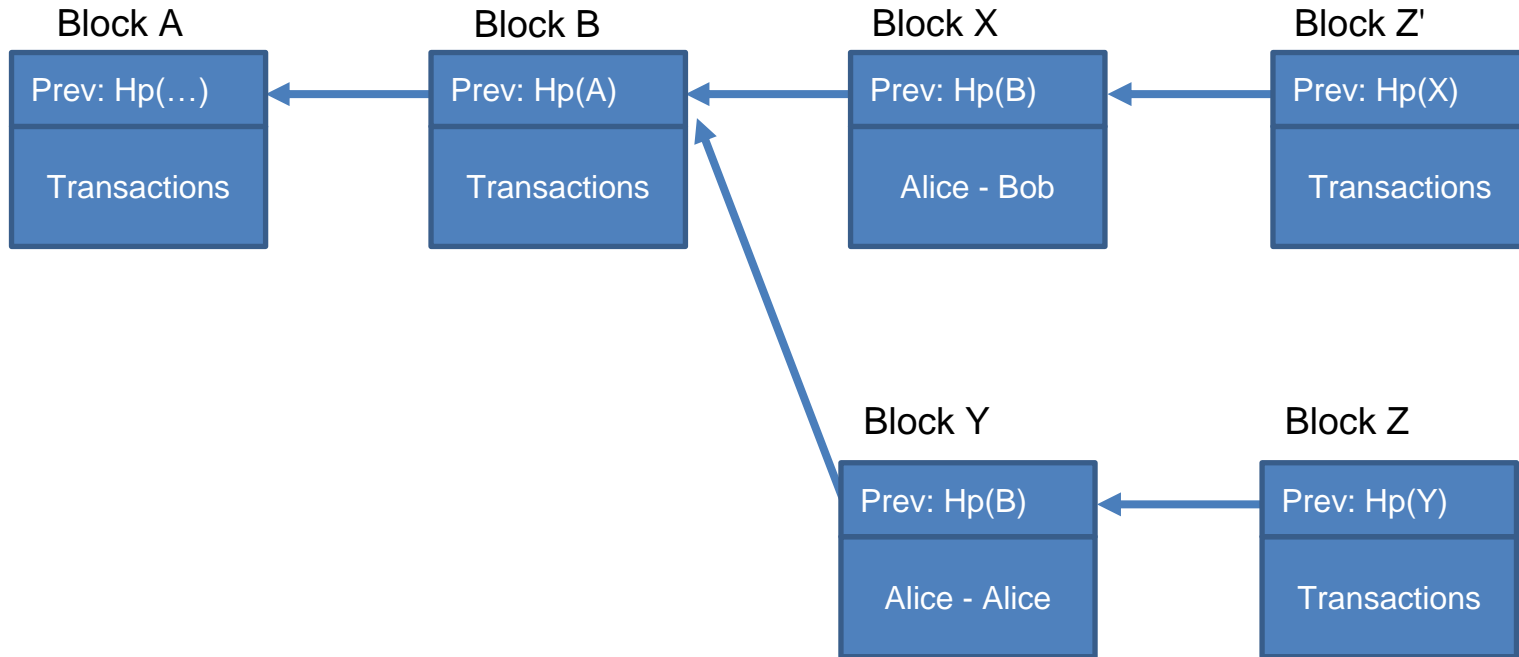
- Media attention
- More people interested in mining
- Safer blockchain
- More value
- ...
- More people interested
- Safer
- More value
- ...

What can a 51% achieve?

1. Steal my Bitcoins? (no, other nodes will not accept the block with invalid signatures)
2. Blacklisting? (yes, but it is easy to detect this)
3. Change the block reward? (no, other nodes will not accept blocks with an invalid award amount)
4. Destroy the confidence in the system? (yes)

Forks

Two (or more) branches in the blockchain



A rule change:

- All the nodes should upgrade their software
- Does not occur naturally
- **Hard forks vs. Soft forks**

Hard fork:

- A change that is not forward compatible
- Introduces rules that were previously not valid
- Example: now you need to check double hash $H(H(tx))$, and not only $H(tx)$
- Nodes that upgraded are fine, but the ones with the old version are not
- The blockchain divides
- Historic example: Bitcoin Cash

Soft fork:

- A change that is forward compatible
- Makes the rules more strict
- Nodes with old software will accept all the new blocks
- Nodes with the new software will reject some blocks that were previously OK
- Miners with old version of the software will realize they need to upgrade when the network starts rejecting their mined blocks
- Example: (almost) any BIP; e.g. P2SH

References

Narayanan et al., Chapter 2