CMPSC 390
Decentralization

Janyl Jumadinova

Credit: Authors of "Bitcoin and Cryptocurrency Technologies"

January 25, 2021

Decentralization in Bitcoin

1. **Who maintains the ledger?**
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?

Decentralization in Bitcoin

Peer-to-peer:

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.

Distributed Consensus

The protocol terminates and all correct nodes decide on the same value.

This value must have been proposed by some correct node.

How consensus *could* work in Bitcoin

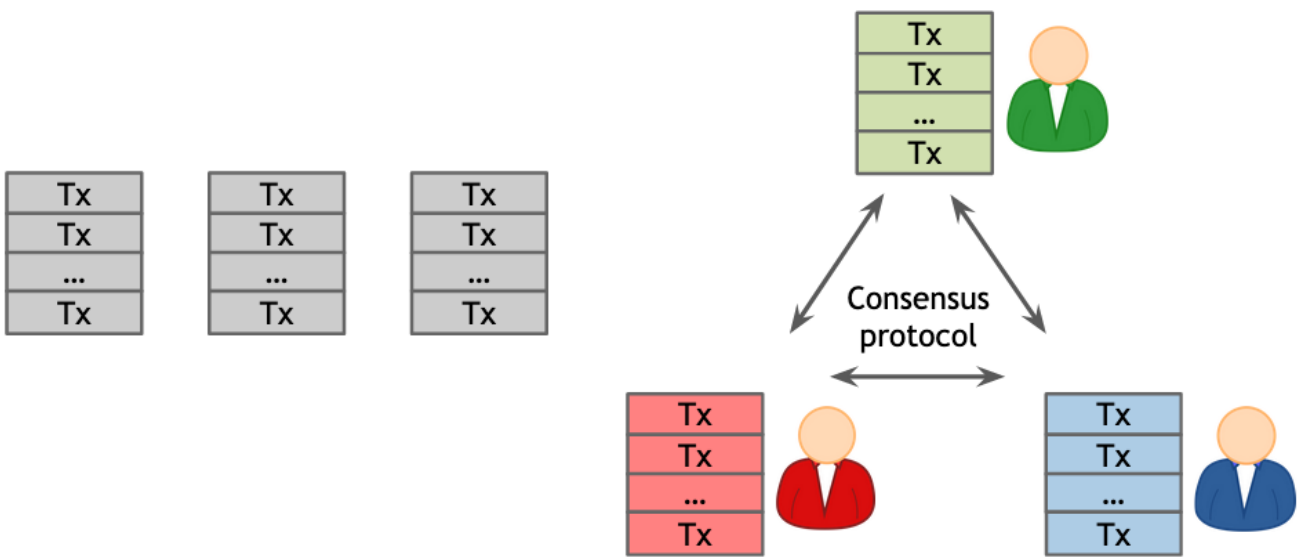
At any given time:

- 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.

How consensus *could* work in Bitcoin

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.



Consensus is hard

- Nodes may crash.
- Nodes may be malicious.

Consensus is hard

- Nodes may crash.
- Nodes may be malicious.
- Network is imperfect:
  - Not all pairs of nodes connected.
  - Faults in network.
  - Latency.

Bitcoin Consensus

- Bitcoin consensus works better in practice than in theory.
- Theory is still catching up.

Bitcoin Consensus

- Bitcoin consensus works better in practice than in theory.
- Theory is still catching up.
- But theory is important, can help predict unforeseen attacks.

Bitcoin Consensus

Introduces incentives

- Possible because it is a currency!

Bitcoin Consensus

Introduces incentives

- Possible because it is a currency!

Embraces randomness:

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

Implicit Consensus

- In each round, random node is picked.

Implicit Consensus

- In each round, random node is picked.
- This node proposes the next block in the chain.

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.

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.

Consensus algorithm (simplified)

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.

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.

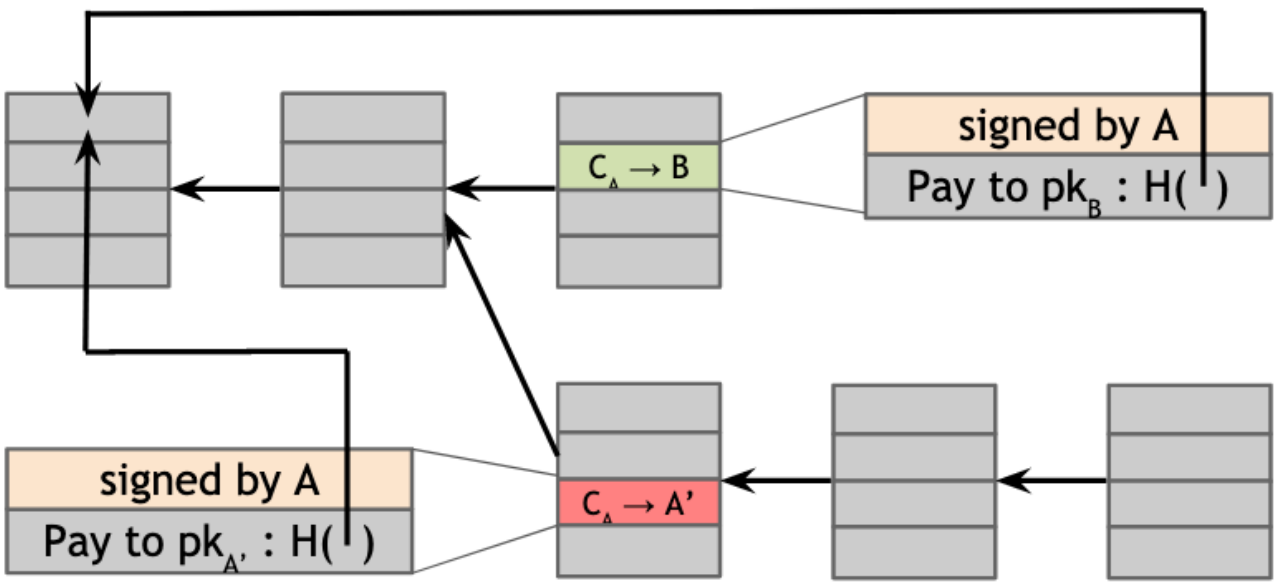
Consensus algorithm (simplified)

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).

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.

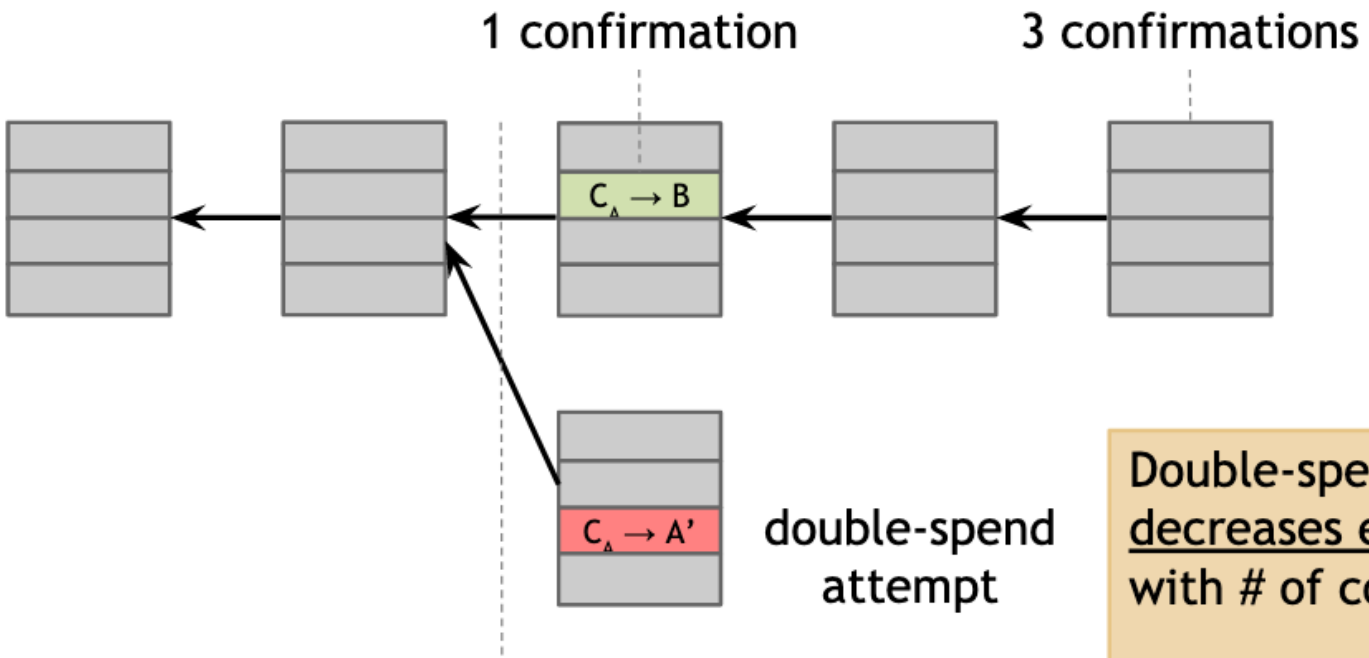
Malicious node?



Double-spending attack

Honest nodes will extend the longest valid branch

From merchant's point of view



Double-spend probability decreases exponentially with # of confirmations

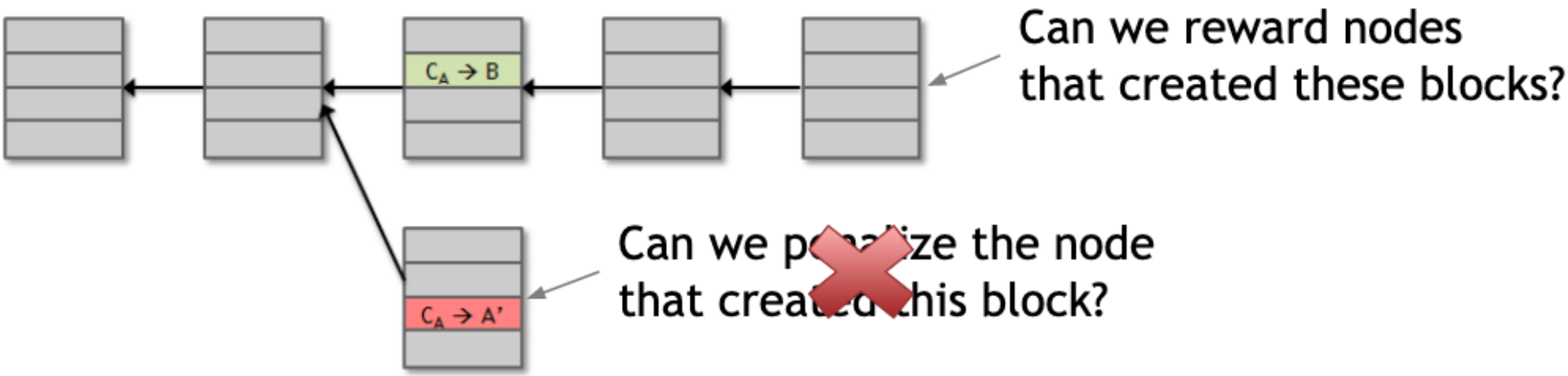
Most common heuristic: 6 confirmations

Summary

- Protection against invalid transactions is cryptographic, but enforced by consensus.
- Protection against double-spending is purely by consensus.
- You are never 100% certain a transaction is in consensus branch. Guarantee is probabilistic.

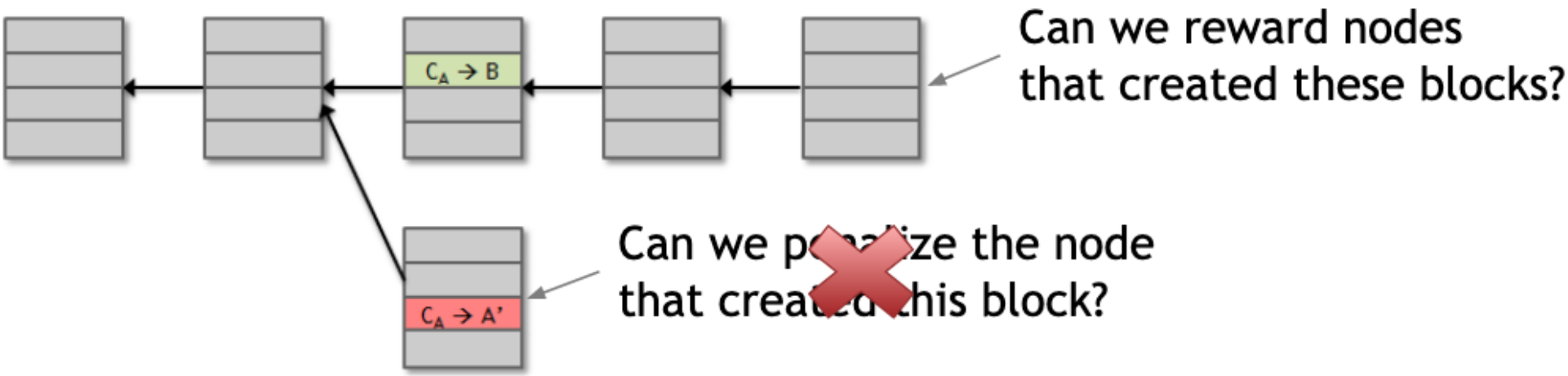
Assumption of honesty is problematic

Can we give nodes incentives for behaving honestly?



Assumption of honesty is problematic

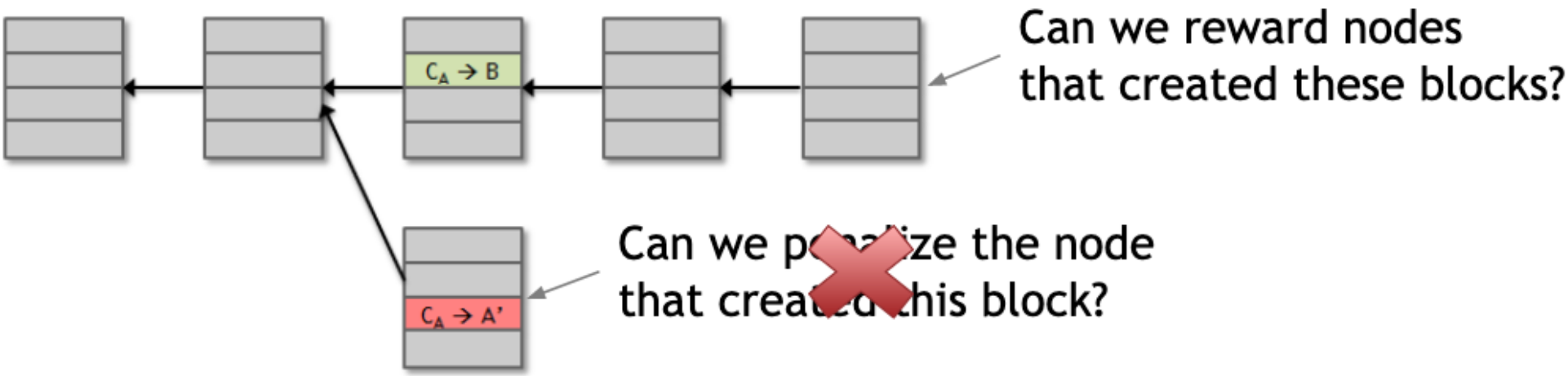
Can we give nodes incentives for behaving honestly?



Everything so far is just a distributed consensus protocol ...

Assumption of honesty is problematic

Can we give nodes incentives for behaving honestly?



Everything so far is just a distributed consensus protocol ... but now we utilize the fact that the currency has value.

Incentive 1: block reward

Creator of block gets to:

- include *special coin-creation transaction* in the block,
- choose recipient address of this transaction.

Incentive 1: block reward

Creator of block gets to:

- include *special coin-creation transaction* in the block,
- choose recipient address of this transaction.

Value is fixed, halves every 4 years.

Incentive 1: block reward

Creator of block gets to:

- include *special coin-creation transaction* in the block,
- choose recipient address of this transaction.

Value is fixed, halves every 4 years.

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

Incentive 2: transaction fees

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

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.

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.

Remaining Problems

We still need to answer:

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

Proof of Work

To approximate selecting a random node:

- select nodes in proportion to a resource,
- that no one can monopolize (we hope).

Proof of Work

To approximate selecting a random node:

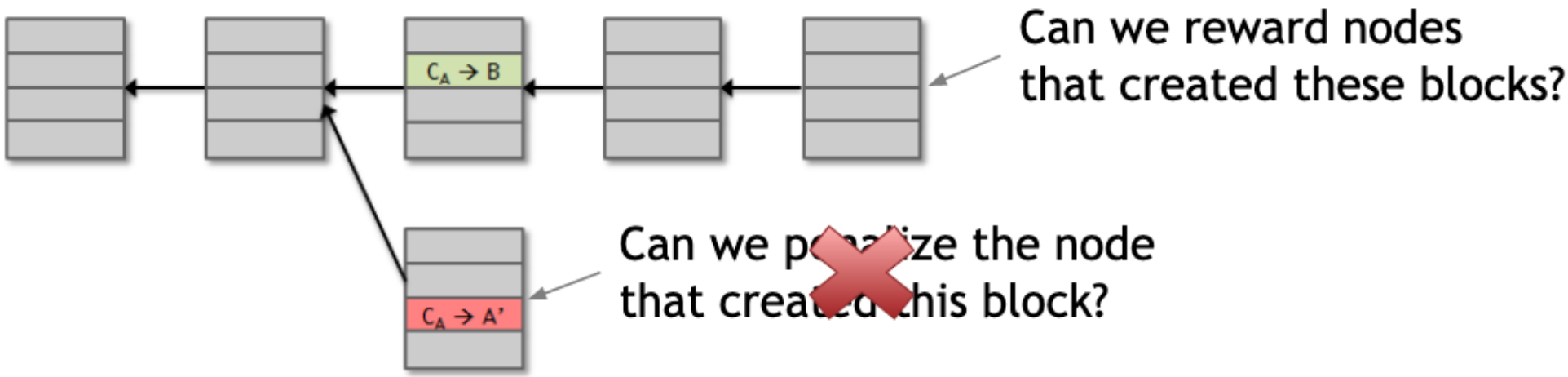
- select nodes in proportion to a resource,
- that no one can monopolize (we hope).

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

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

Hash Puzzles

To create a block, find nonce such that:
$H(nonce||prev\_hash||tx||...||tx)$ is very small or
$H(nonce||prev\_hash||merkleRoot < target$.



Proof of Work Properties

Property 1: Difficult to compute.

Only some nodes bother to compete - miners.

Proof of Work Properties

Property 1: Difficult to compute.

Only some nodes bother to compete - miners.

Property 2: Parameterizable cost.

Nodes automatically re-calculate the target every two weeks.

Proof of Work Properties

Property 1: Difficult to compute.

Only some nodes bother to compete - miners.

Property 2: Parameterizable cost.

Nodes automatically re-calculate the target every two weeks.
Goal: *average* time between blocks = 10 minutes

Proof of Work Properties

Property 1: Difficult to compute.

Only some nodes bother to compete - miners.

Property 2: Parameterizable cost.

Nodes automatically re-calculate the target every two weeks.
Goal: *average* time between blocks = 10 minutes

Property 3: Trivial to verify.

Nonce must be published as part of block.
Other miners simply verify that $H(nonce prev_hash tx \ldots tx) < target$

Summary

Bitcoin has three types of consensus:

1. Value
2. State
3. Rules

Summary

Bitcoin has three types of consensus:

1. Value
2. State
3. Rules

Next time:

- How do we get from consensus to currency?
- What else can we do with consensus?