Blockchains & Cryptocurrencies

Bitcoin

Instructor: Matthew Green
Johns Hopkins University - Fall 2020

Many slides based on NBFMG

Housekeeping

- Readings: due today Lamport, P2P blog
- A1 is still due 9/24 11:59pm Baltimore time
- **Project groups and proposal is due 10/8 end of day**

News?

News?

India Said to Be Preparing to Ban Cryptocurrency Trading: Bloomberg

Sep 15, 2020 at 13:03 UTC ▪ Updated Sep 15, 2020 at 13:30 UTC

News?


Story from Policy & regulation →
Iran May Fund Car Imports With Cryptocurrency Minin
Sep 14, 2020 at 09:31 UTC ▪ Updated Sep 14, 2020 at 13:46 UTC

News?

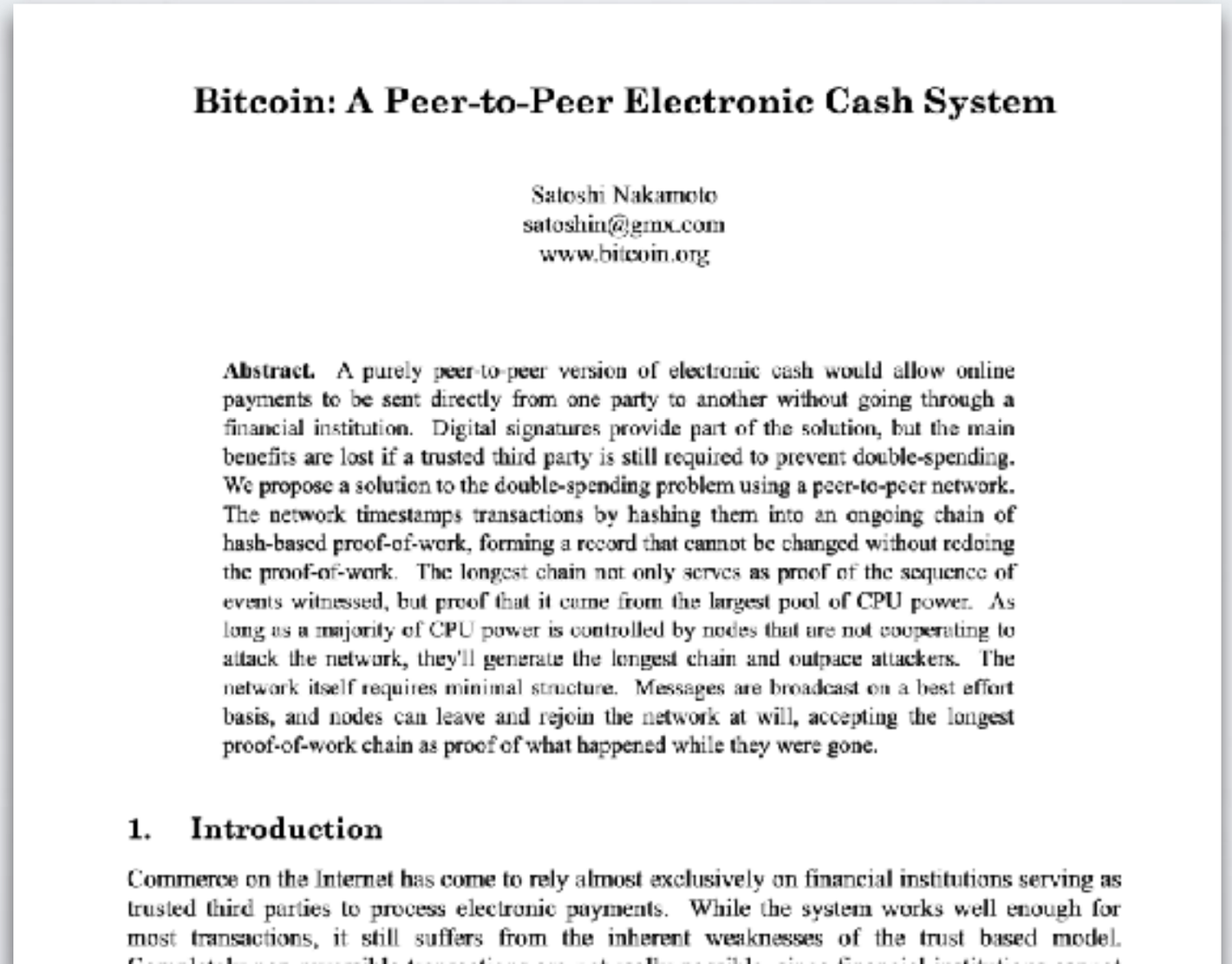
IRS offers grants for software to trace privacy-focused cryptocurrency trades

Grants of up to $625,000 will be issued in exchange for cryptocurrency tracking technologies.

MORE FROM CHARLIE OSBORNE

Today

- We're going to finish talking about consensus and finally work our way on to Bitcoin



Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

1. Introduction

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model.

How do we solve double spending?

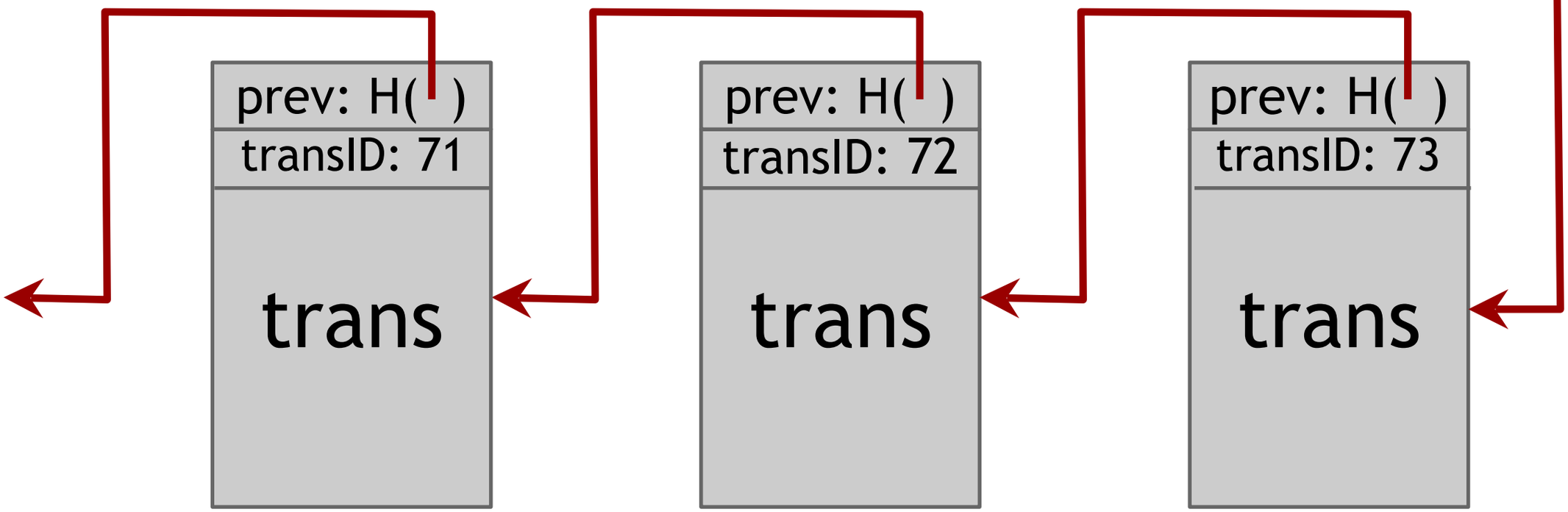
- Simplest answer: send all transactions to an atomic, append-only centralized ledger
- Have the ledger provide a definite ordering for transactions
  - If two transactions conflict, simply disallow the later one
- No TX is valid unless the ledger has "approved" and ordered it

Scrooge publishes a history of all transactions in an “append-only” ledger

H( )

Sig

Implement the ledger using a block chain, signed by Scrooge



optimization: put multiple transactions in the same block

CreateCoins transaction creates new coins

Valid, because I said so.

transID: 73 type:CreateCoins

coins created

num	*value*	*recipient*
0	3.2	0x...
1	1.4	0x...
2	7.1	0x...

coinID 73(0)

coinID 73(1)

coinID 73(2)

signature

CreateCoins transaction creates new coins

Valid, because I said so.

transID: 73	type:CreateCoins	
coins created		
num	*value*	*recipient*
0	3.2	0x...
1	1.4	0x...
2	7.1	0x...

These are public keys!

coinID 73(0)

coinID 73(1)

coinID 73(2)

signature

PayCoins transaction consumes (and destroys) some coins, and creates new coins of the same total value

transID: 73 type:PayCoins

consumed coinIDs:
68(1), 42(0), 72(3)

coins created

num	*value*	*recipient*
0	3.2	0x...
1	1.4	0x...
2	7.1	0x...

signatures

Valid if:
-- consumed coins valid,
-- not already consumed,
-- total value out = total value in, and
-- signed by owners of all consumed coins

One signature for each consumed coin

Distributed consensus

Bitcoin is a peer-to-peer system

When Alice wants to pay Bob:
she broadcasts the transaction to all Bitcoin nodes

signed by Alice
Pay to pk_{Bob} : H( )

Note: Bob's computer is not in the picture

Bitcoin is a peer-to-peer system

This network is a fill/flood style P2P network: all nodes perform basic validation, then relay to their peers

This introduces bootstrapping, spam and DoS problems, which are dealt with through “seeders” and “reputation” scores

signed by Alice

Pay to pk_{Bob} : H( )

Why aren’t we done here?

Why can’t we just trust this system to eliminate invalid blocks, and give everyone a robust view of the Tx history?

signed by Alice

Pay to pk_{Bob} : H( )

Bitcoin’s key challenge

Key technical challenge of decentralized e-cash: distributed consensus

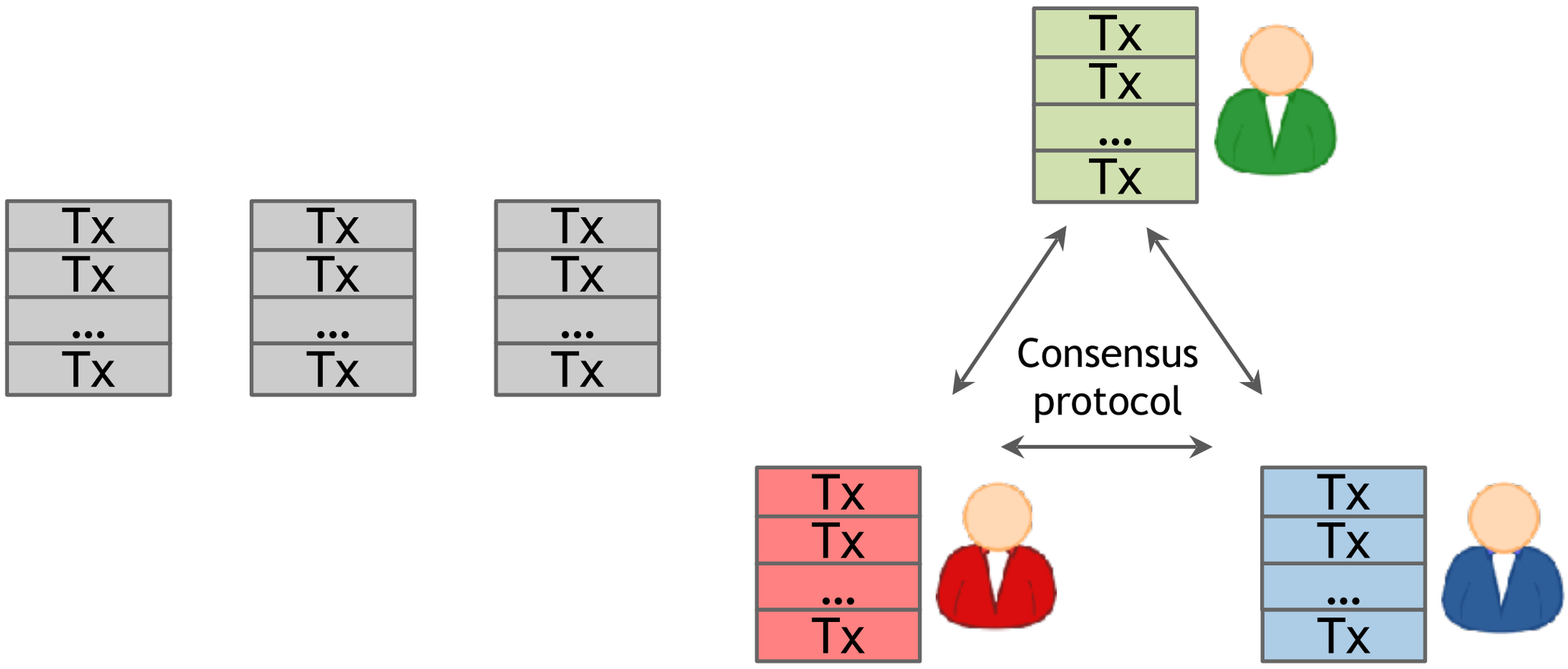
or: how do all of these nodes agree on an ordered history of transactions?

Defining distributed consensus

The protocol terminates and all honest nodes decide on the same **value**

This value must have been proposed by some honest node

How consensus <u>could</u> work in Bitcoin



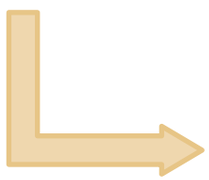
OK to select any valid block, even if proposed by only one node

Why consensus is hard

Nodes may crash
Nodes may be malicious

Network is imperfect

- Not all pairs of nodes connected
- Faults in network (“partitioning”)
- Latency



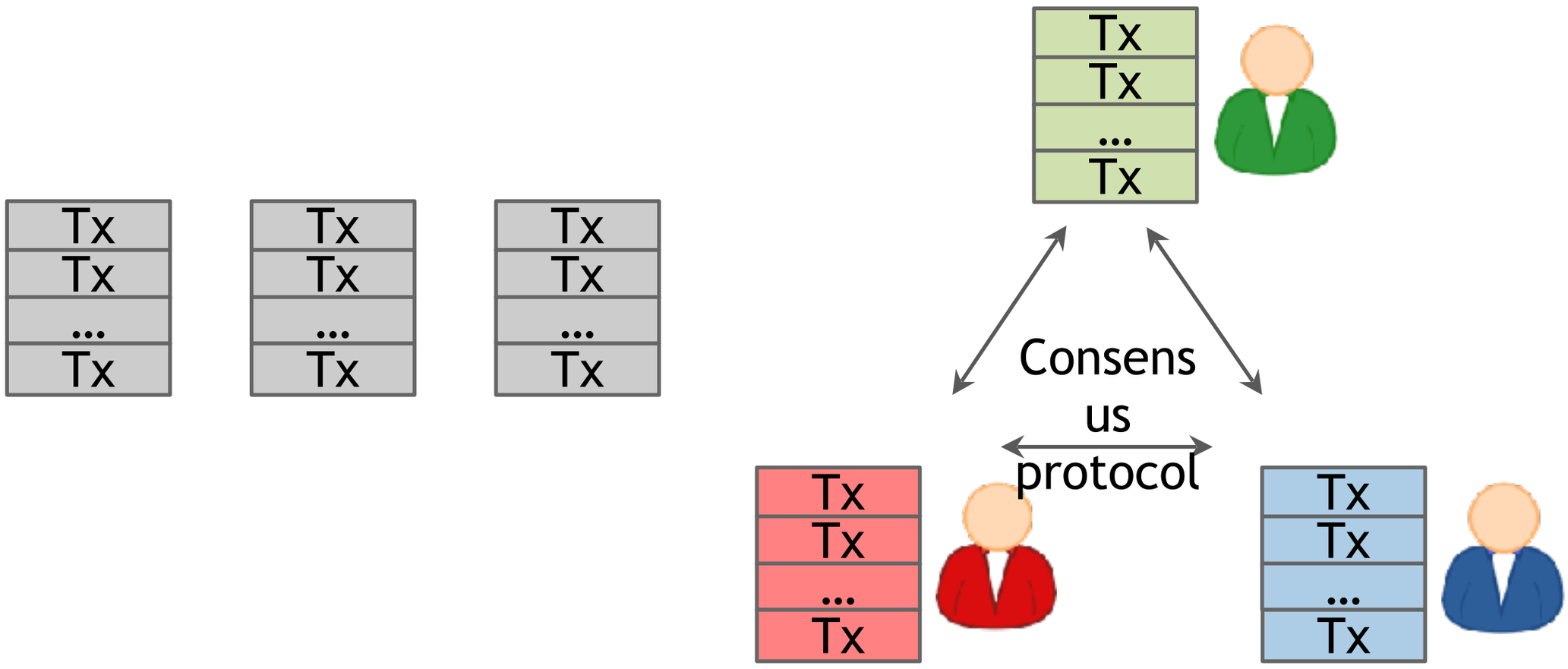
No notion of global time

Defining distributed consensus

The protocol terminates and all honest nodes decide on the same value (history)

This value must have been proposed by some honest node

How consensus <u>could</u> work in Bitcoin



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

Many impossibility results

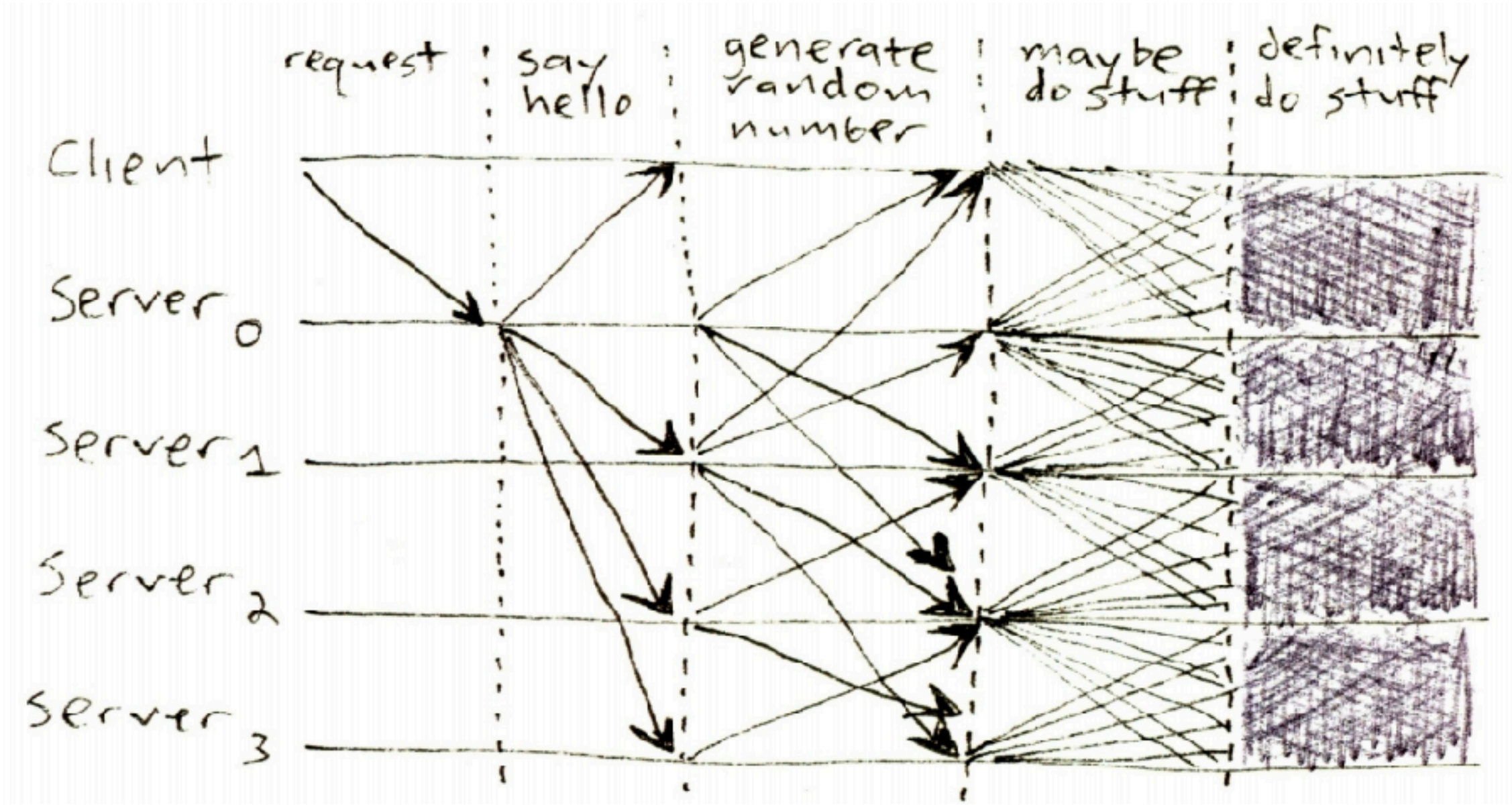
- Impossible without 2/3 honest majority [Pease, Shostak, Lamport'80]

- Impossible with a single faulty node, in the fully asynchronous setting, with deterministic nodes [Fischer-Lynch-Paterson'85]

Why do these results matter?

- Because without node identities, an attacker could easily crash these networks by impersonating many nodes (“Sybil attack”)

- Because synchronicity is hard


request
say hello
generate random number
maybe do stuff
definitely do stuff
Client
Server 0
Server 1
Server 2
Server 3

Some positive results

Example: Paxos [Lamport]

Never produces inconsistent result, but can (rarely) get stuck

Understanding impossibility results

These results say more about the model than about the problem

The models were developed to study systems like distributed databases

Bitcoin consensus: theory & practice

- Bitcoin consensus: initially, seemed to work better in practice than in theory

- Theory has been steadily catching up to explain why Bitcoin consensus works [e.g., Garay-Kiayias-Leonardos'15,Pass-Shelat-Shi'17,Garay-Kiayias-Leonardos'17,…]

- Theory is important, can help predict unforeseen attacks

Some 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 identity: the blockchain

Why identity?

Pragmatic: some protocols need node IDs

Security: assume less than 50% malicious

Why don’t Bitcoin nodes have identities?

Identity is hard in a P2P system — Sybil attack

Pseudonymity is a goal of Bitcoin

Weaker assumption: select random node

Analogy: lottery or raffle

When tracking & verifying identities is hard, we give people tokens, tickets, etc.

Now we can pick a random ID & select that node

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) and it builds on a chain they already accept
5. Nodes express their acceptance of the block by including its hash in the next block they create

So how do we pick a random node?

Resources & Consensus

- One computer can easily pretend to be many "nodes", so simple random selection != good
- But an observation: resources (e.g., hardware, storage, CPU, GPU, etc.) are much harder to fake
- Idea: make your probability of winning the vote proportional to your overall resources

Key idea: implicit consensus

In each round, all nodes compete to solve a puzzle

The winner proposes the next block in the chain
and sends out their solution along with it

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

H( )

prev: H( )

nonce

List of transactions

prev: H( )

nonce

List of transactions

prev: H( )

nonce

List of transactions

What’s the puzzle?

The puzzle is simply the hash of the new block, which must be chained off the previous block

I.e., find a **nonce** such that H(block | nonce) < T for some T

The winner proposes the next block in the chain
and sends out their nonce

Other nodes implicitly accept/reject this block
- by either extending it
- or ignoring it and extending chain from earlier block

<u>What happens if nodes ignore the block?</u>

“Mining”

The process of finding new blocks that solve the puzzle, broadcasting, and appending them to the chain is called “mining”

This is expensive in terms of computation and electricity

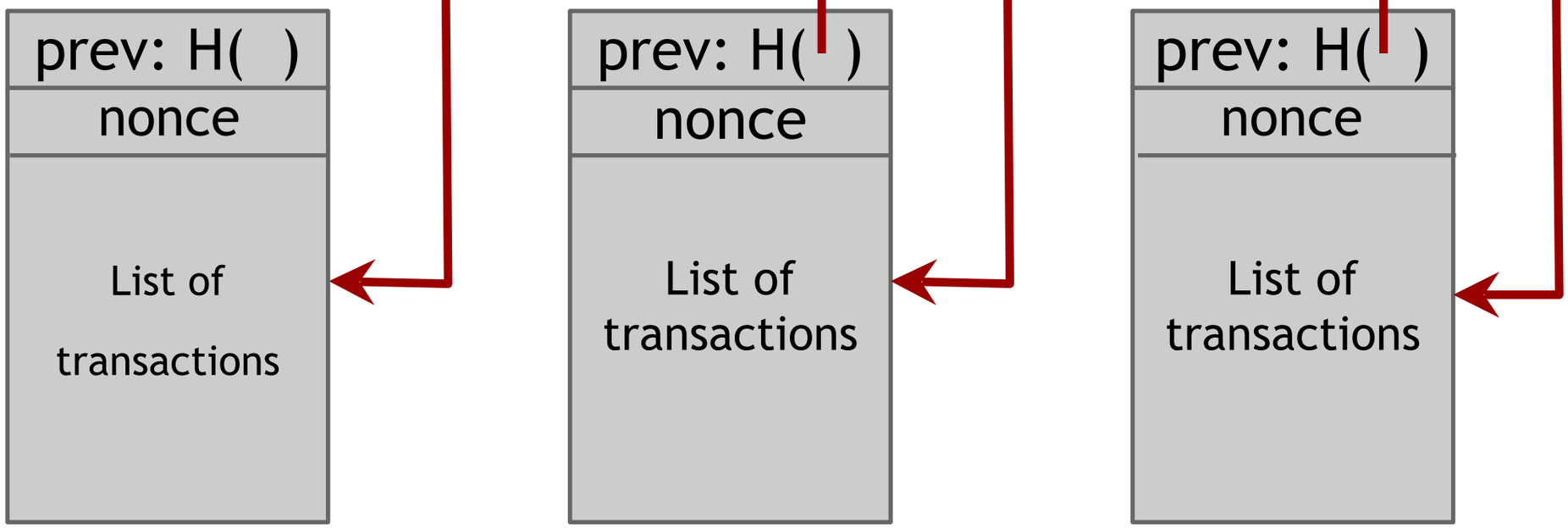
There are incentives for nodes to do it

Genesis (Honest)

Evil

Honest

H( )



Where do we get *T*?

The value T is called the “block difficulty target”.
It’s adjustable.

Ideas:

- Choose T once at the start, keep fixed
- Change T from time to time

What are the impacts of these choices?

Selecting T (Bitcoin)

Bitcoin's difficulty changes every 2016 blocks

Goals:

- Bitcoin block time should <u>average</u> 10 minutes
- Everyone in the network agrees on *T*
- *Hence must be a function of chain data*

This brings back a notion of time

Selecting T (Bitcoin)

Every block contains a (packed) encoding of T (target) which corresponds to ***"difficulty" (d)***

d = ((2^{16} - 1) * 2^{8*26}) / T <- relationship between d, T

Goals:

- Bitcoin block time should <u>average</u> 10 minutes
- 2016 blocks * 10 minutes == 2 weeks
- Everyone in the network agrees on *T*
- *Hence must be a function of chain data*

<u>This brings back a notion of time</u>

Bitcoin blocks include timestamps

Every block contains a timestamp that alleges when it was found

Remember that nodes can be adversarial! So some timestamps may be lies! This requires heuristics:

- Each timestamp must be no sooner than the median of the past 11 blocks
- Honest nodes must reject timestamps that are > 2hrs in the future

This brings back a notion of time

What if there’s a collision?

Sometimes two separate nodes find a valid solution simultaneously

This can result in network partition

What if there's a collision?

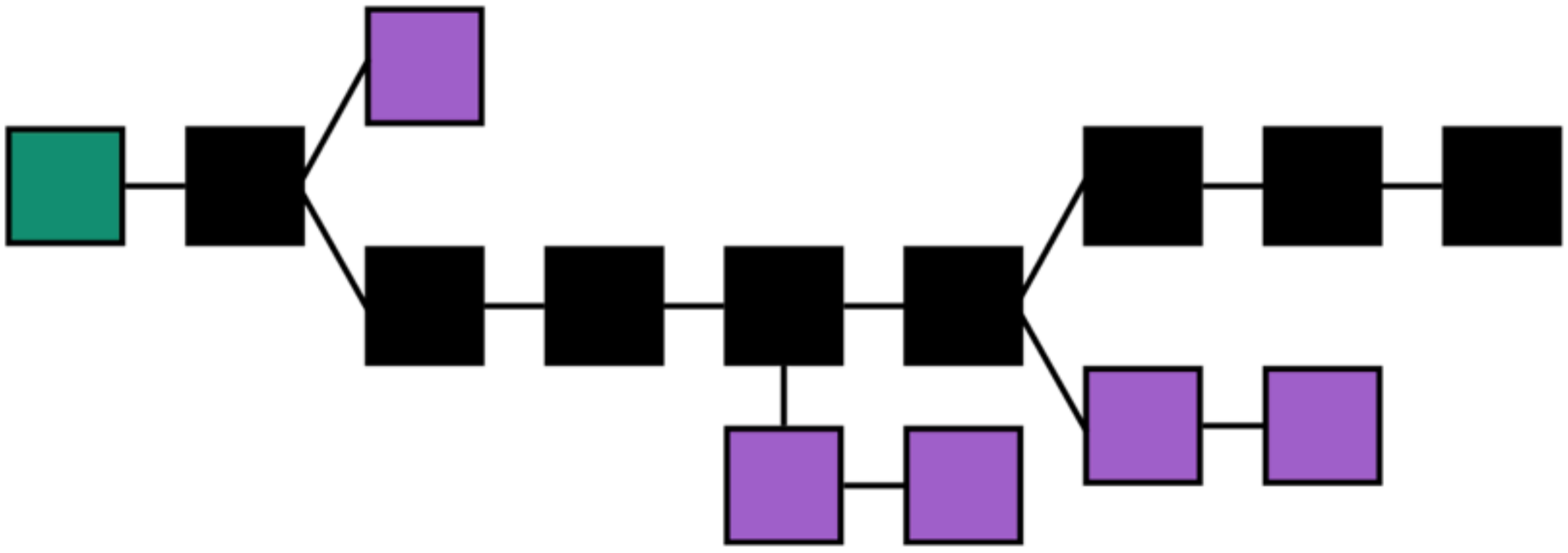


Image CC-BY-3 Theymos taken from the Bitcoin wiki

“Longest chain rule”

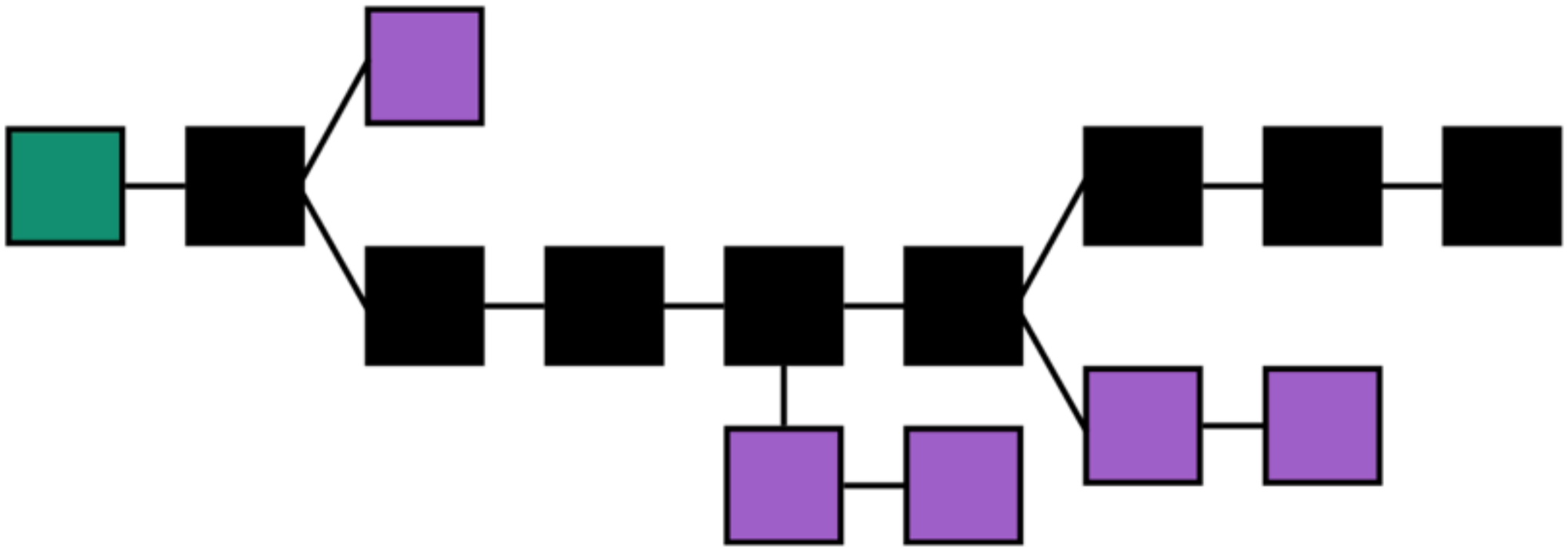


Image CC-BY-3 Theymos taken from the Bitcoin wiki

“Longest chain rule”

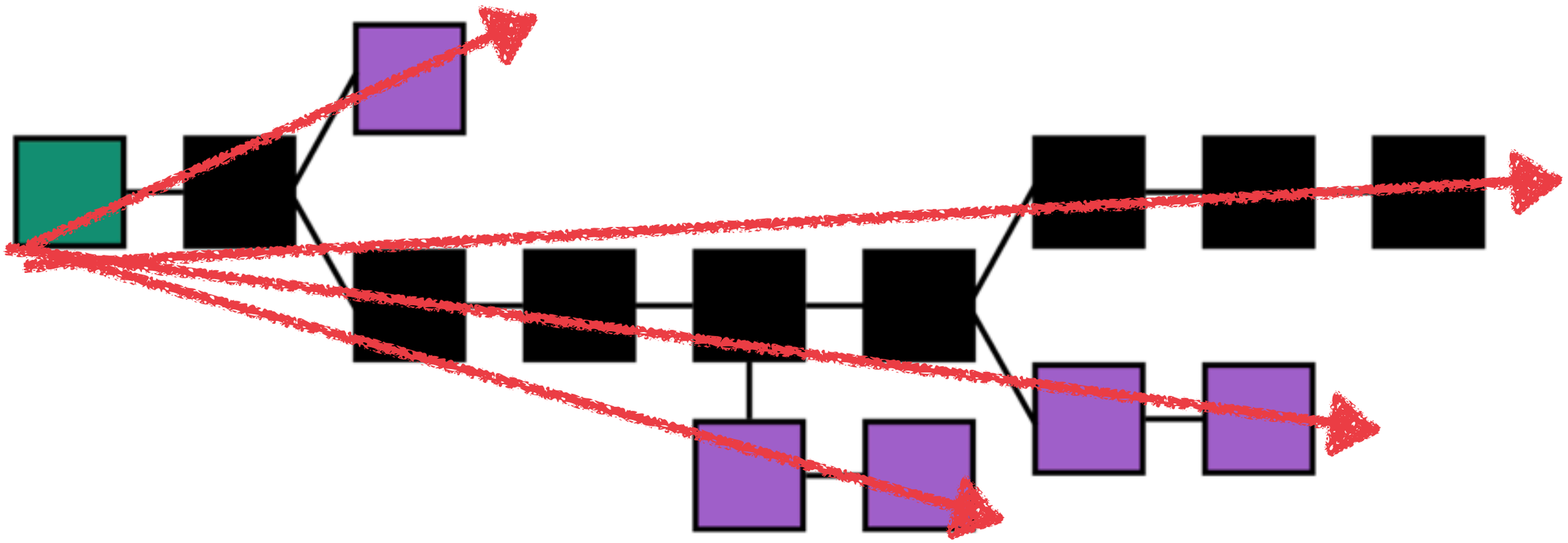


Image CC-BY-3 Theymos taken from the Bitcoin wiki

This is good and bad

Good: if we experience a “chain fork” and the network is connected (I.e., not totally partitioned), then eventually we will learn about both forks

Good: if the “hash power” behind the two chains is unequal, we will probably end up with one chain getting longer

Even if the hash power is equal, the inherent randomness of the puzzle (PoW) will likely cause an advantage

As one chain grows longer, other nodes will adopt it, and start adding to it

What’s the bad?

What’s the bad?

When a chain becomes longer than the “current chain” a node thinks is the longest chain, they must abandon that older chain

Finality

“Finality is the assurance or guarantee that cryptocurrency transactions cannot be altered, reversed, or canceled after they are completed.”

https://academy.binance.com/glossary/finality

Finality

“Finality is the assurance or guarantee that cryptocurrency transactions cannot be altered, reversed, or canceled after they are completed.”

Bitcoin’s finality is probabilistic (and computational)

Reorganizations become less probable (and more expensive) over time, but they never become impossible*

How many blocks can the adversary make?

Consider an adversary that controls a r-fraction of the hash power

How many blocks in expectation can they build in a t-block window?

What is the probability that they dominate that t-block window?

How many blocks can the adversary make?

Consider an adversary that controls a r-fraction of the hash power

How many blocks in expectation can they build in a t-block window?

What is the probability that they dominate that t-block window?

We will see some attacks that leverage these simplified calculations later….

How do we incentivize mining?

How do we incentivize mining?

Two answers to this question in Bitcoin:

1. “Transaction fees” Each transaction has a “tip” that can be collected by the node who mines it into a block (incentivizes inclusion of transactions)

2. “Block reward” 50/25/12.5/… BTC made from scratch (in a special Coinbase transaction) and given to the miner