Blockchains & Cryptocurrencies

Consensus & Bitcoin (II)

Instructor: Matthew Green & Abhishek Jain
Johns Hopkins University - Spring 2023

Many slides based on NBFMG

Housekeeping

- New readings (you will care about this on the midterm!)
- **Project news:**
  - Please start finding team members (**3-4**), no individual projects
  - If you have trouble finding a team member, post on Piazza and/or talk to the TAs: we will help you
  - We will provide a list of project topics by tonight (Piazza)
  - **1 page writeup due February 15**

Housekeeping (II)

- Some logistics updates:
  - We will be having a "late" midterm (mid-April, after break)
  - **There will be no final exam in this class**
  - All waitlisted students can enroll

Errata

- 1 bitcoin == 100 million satoshi (10^8)

News?

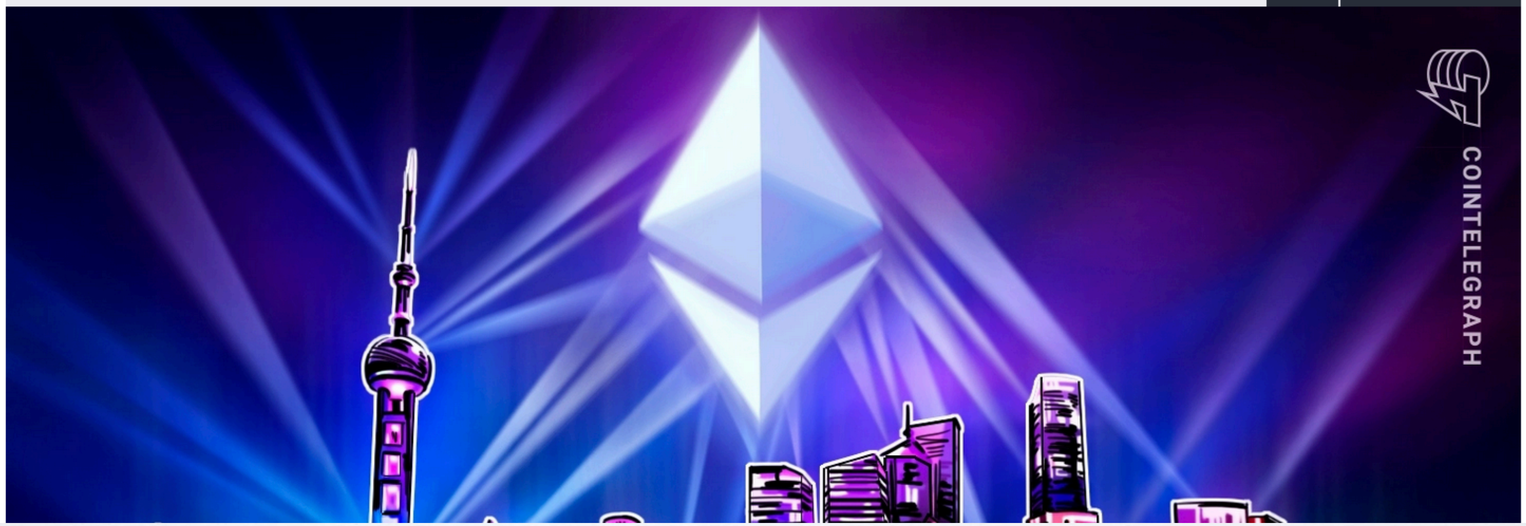
News?

KONSTANTIN BOYKO-ROMANOVSKY 15 HOURS AGO

Ethereum's Shanghai fork is coming — but it doesn't mean investors should dump ETH

Ethereum’s Shanghai upgrade is tentatively set for March, which means approximately 14% of ETH supply will unlock in the year ahead.

4335 Total views **26** Total shares Listen to article 7:08



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

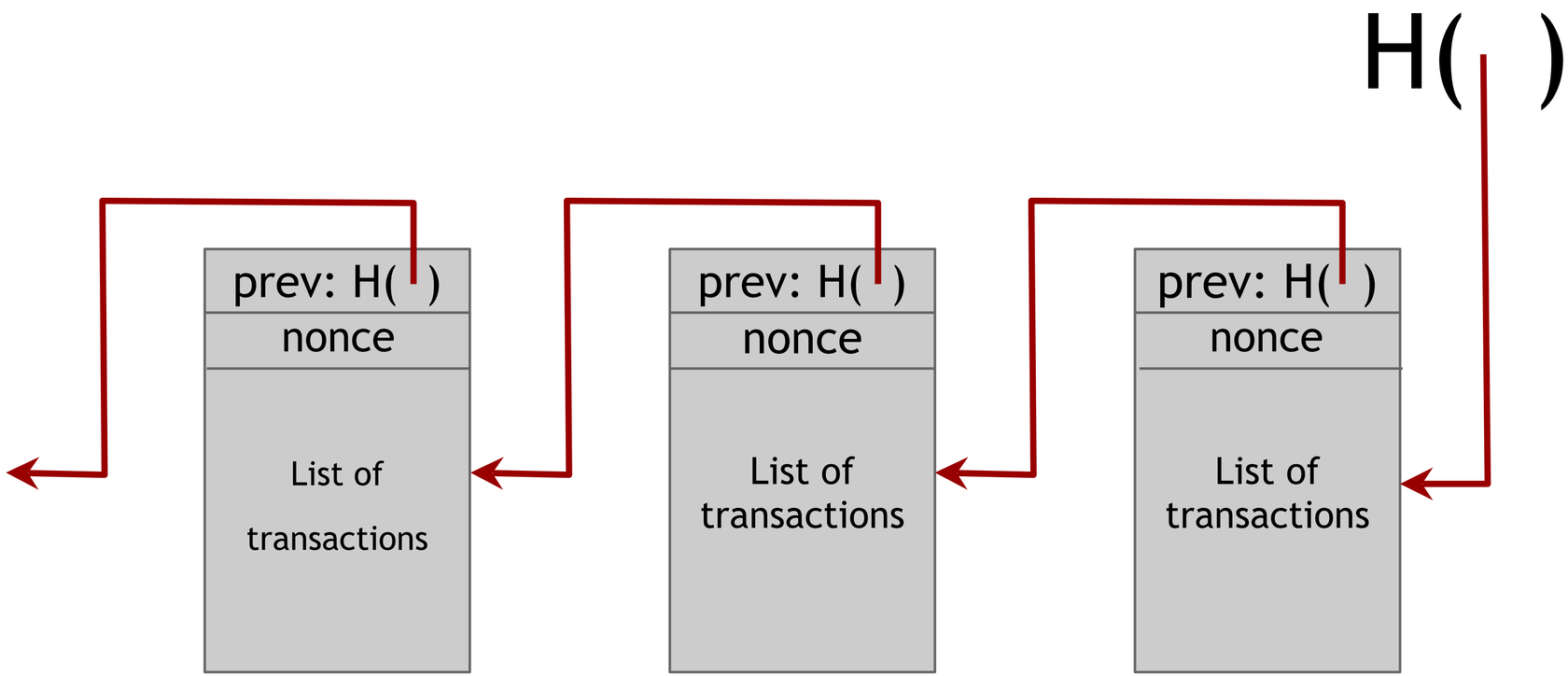
Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required 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

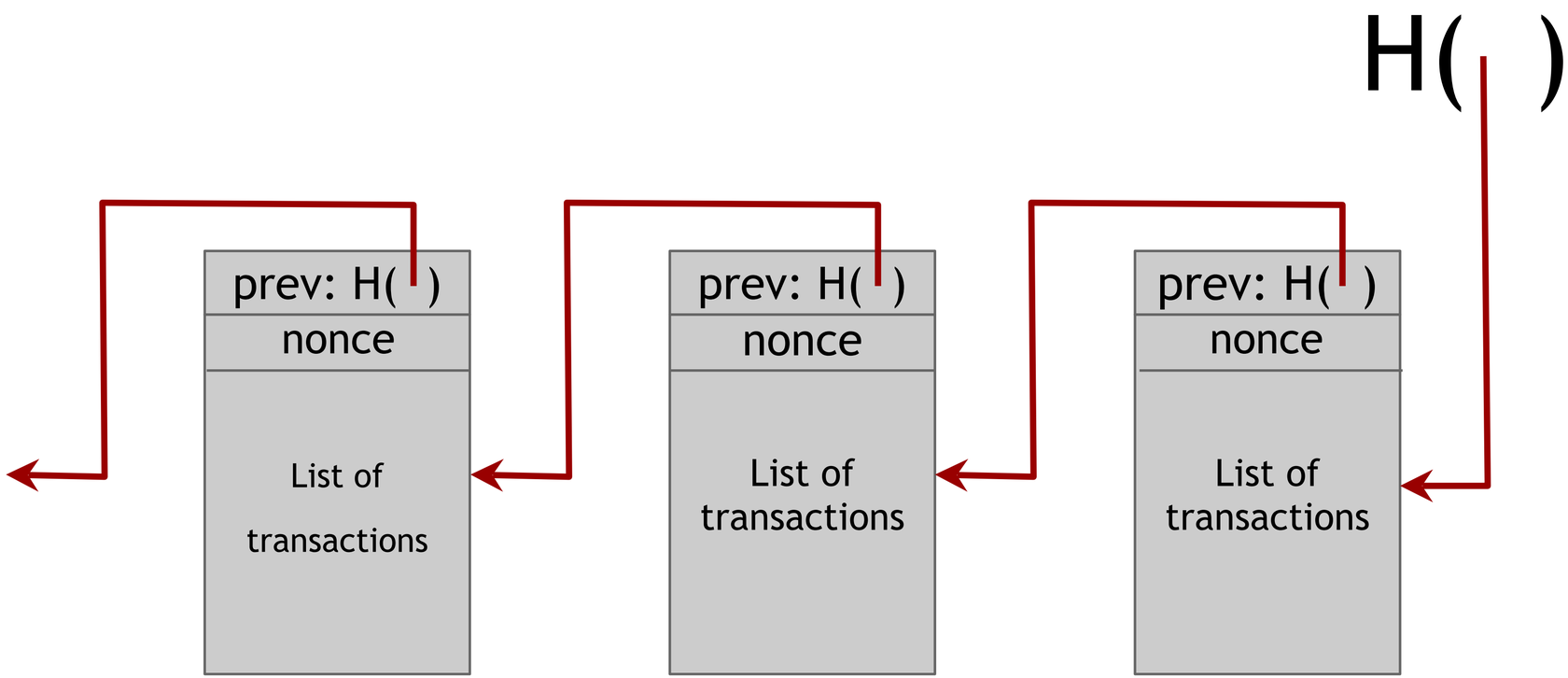
Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model.

A few ingredients from last time

Blockchain (linked list)



Blockchain (linked list)



Nice features?

Transactions

“Coin creation”

1. Consumes no previous TXO
2. Outputs at least one new TXO (value + public key)

(Who gets to make this? No idea!)

“Payment”

1. Consumes one or more previous TXOs
2. Outputs at least one new TXO (value + public key)
3. Includes one signature per consumed TXO
4. Anyone can make a payment!

Payment transaction

transID: 73 type:PayCoins
consumed coinIDs: 68(1), 42(0), 72(3)
transaction outputs

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

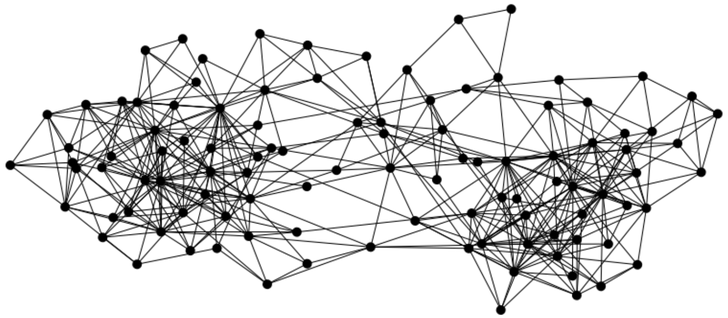
Peer-to-peer “gossip” network

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( )

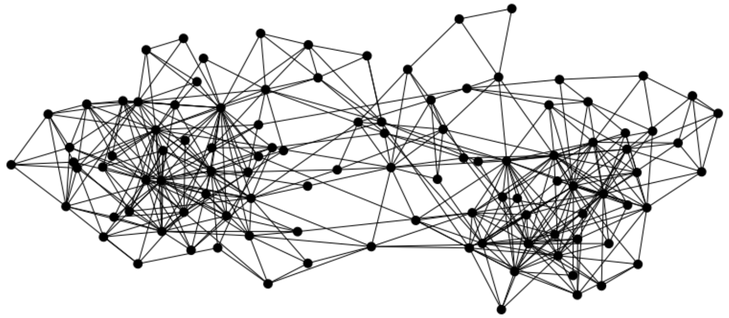


Bitcoin is a peer-to-peer system

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’s key challenge

Key technical challenge of decentralized e-cash: distributed consensus

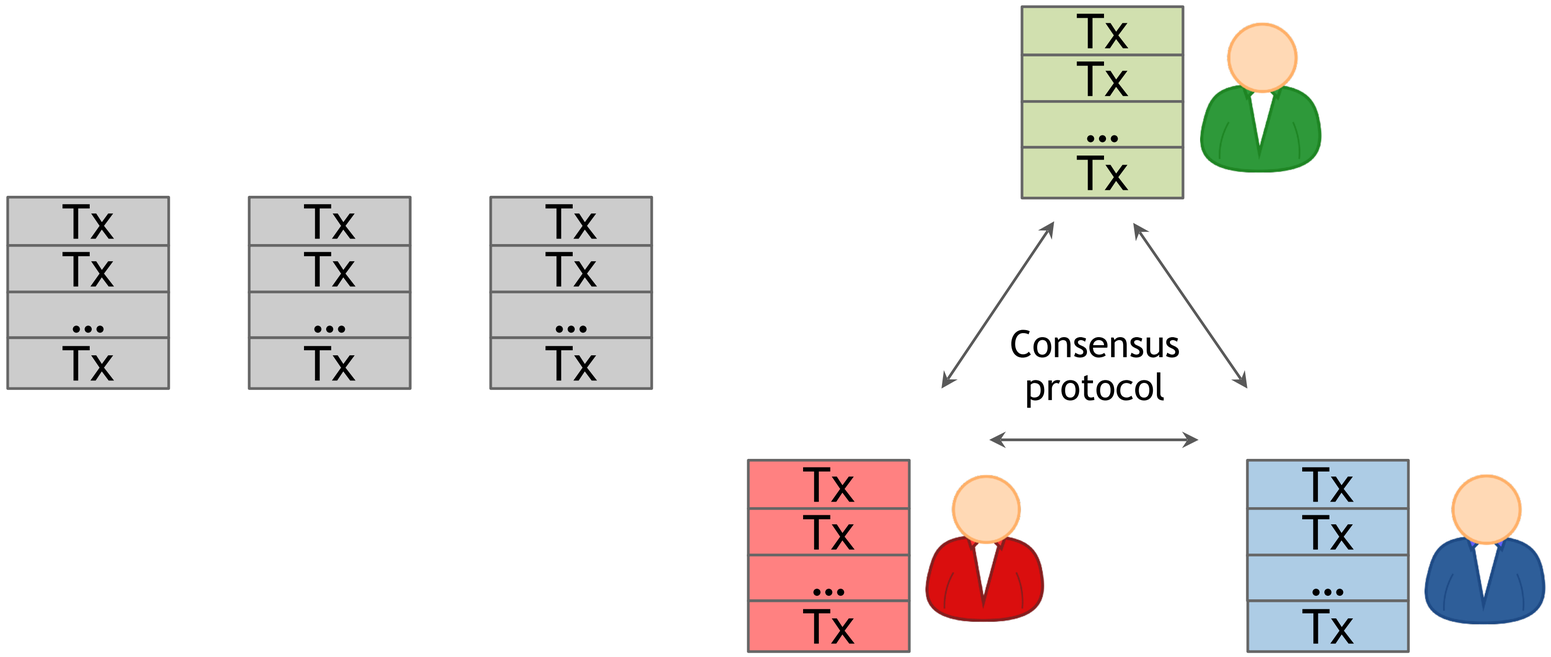
or: how do all of these nodes agree on an ordered history of transactions?

How consensus could work in Bitcoin

At any given time:

- All nodes have a sequence of blocks of transactions they've reached consensus on
  (AKA a "current blockchain")
- (Blocks are also distributed via p2p network)
- Each node has a set of outstanding transactions it's heard about

How consensus could 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
Nodes may be “sybils”, operated by one user

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

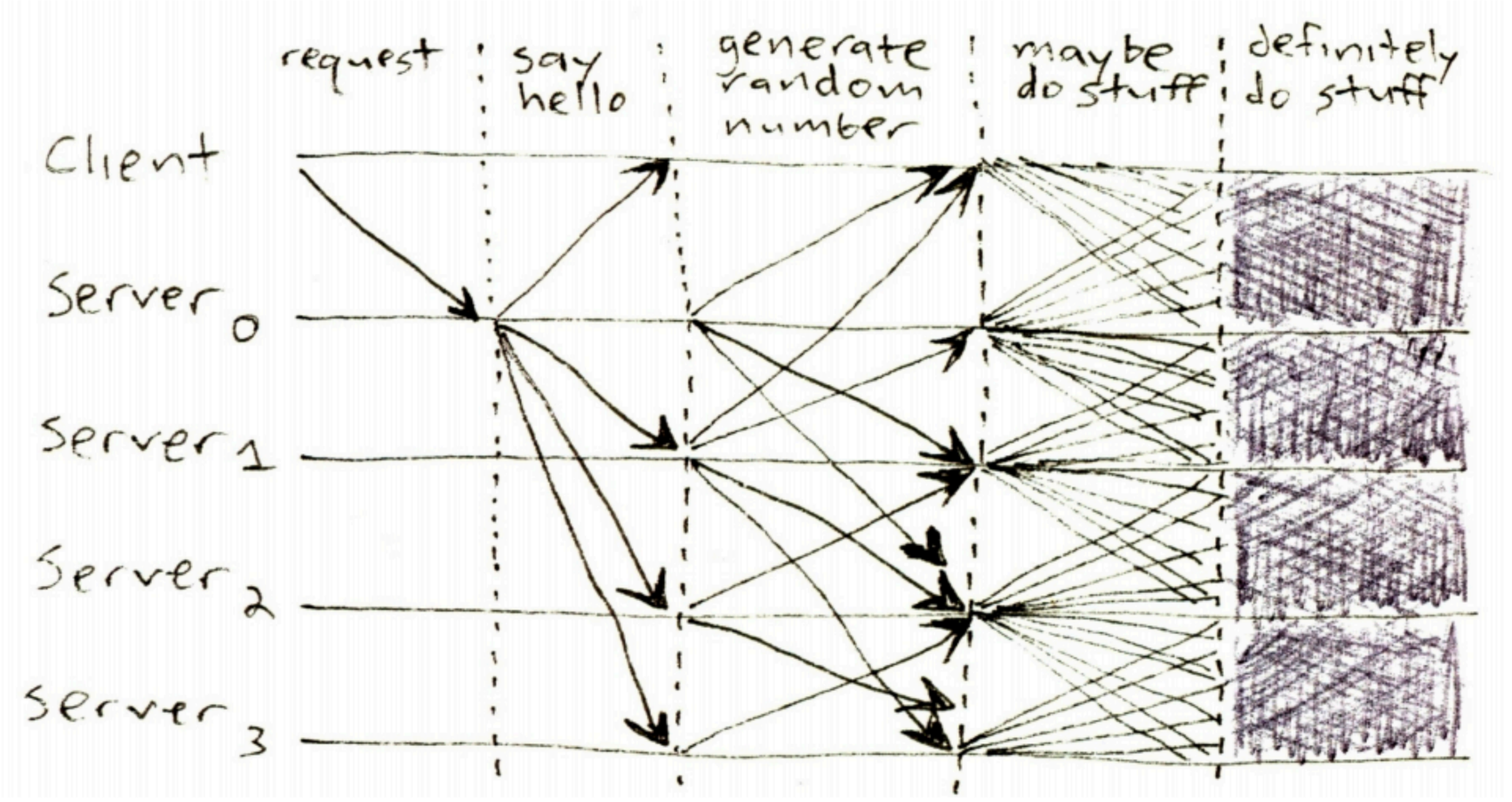
Many impossibility results

- 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

Why do these results matter?


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

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

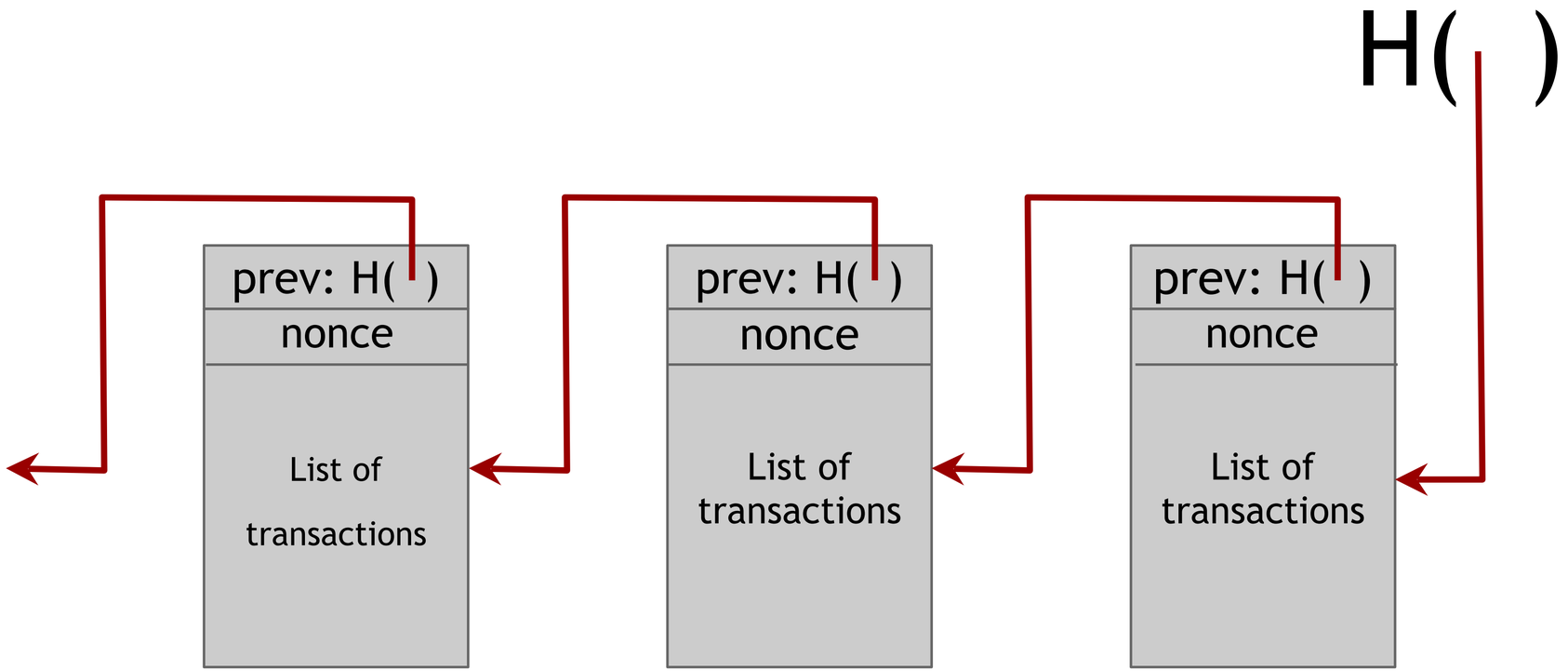
In each round, all nodes compete to solve a puzzle

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

Examples


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

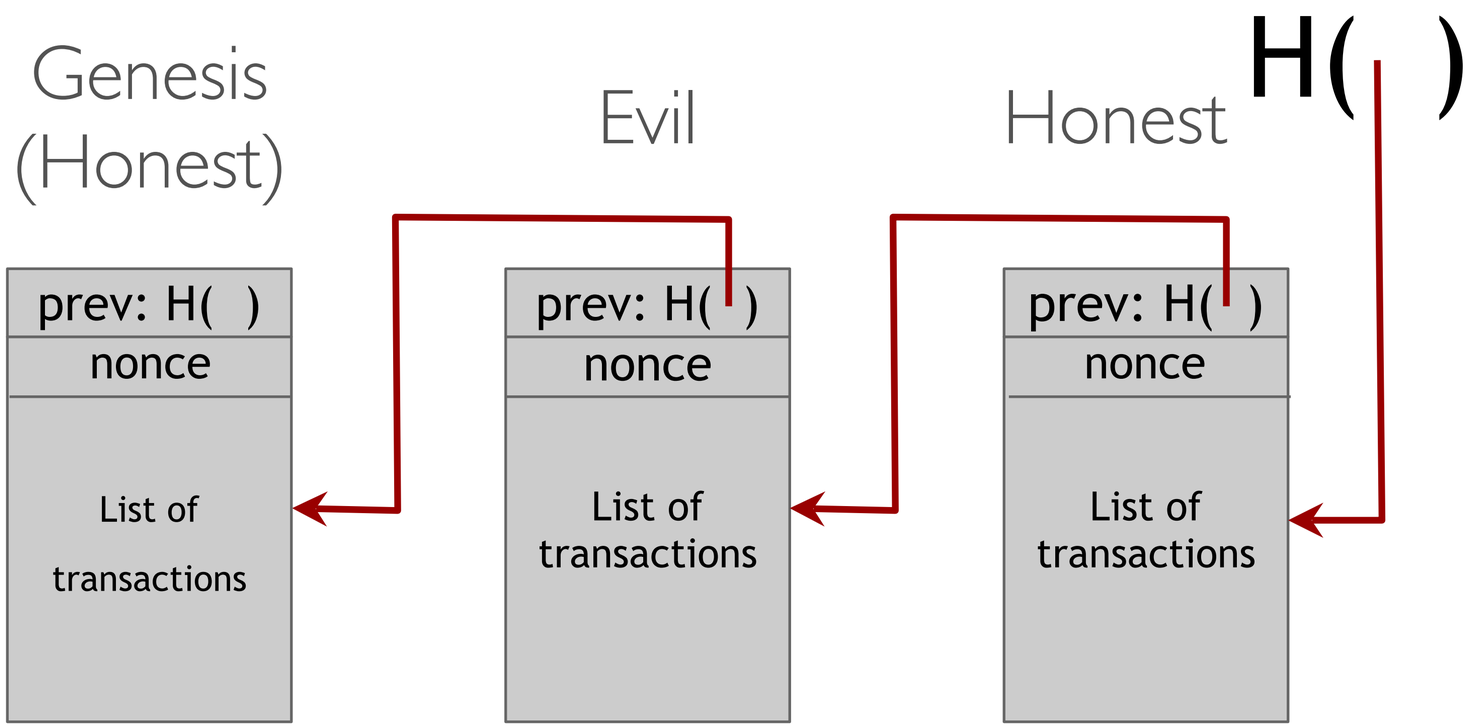
What happens if nodes ignore the block?

“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( )
prev: H( )
nonce
List of
transactions
prev: H( )
nonce
List of
transactions
prev: H( )
nonce
List of
transactions

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 average 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 average 10 minutes
- 2016 blocks * 10 minutes == 2 weeks
- Everyone in the network agrees on *T*
- *Hence must be a function of chain data*

This brings back a notion of time

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?

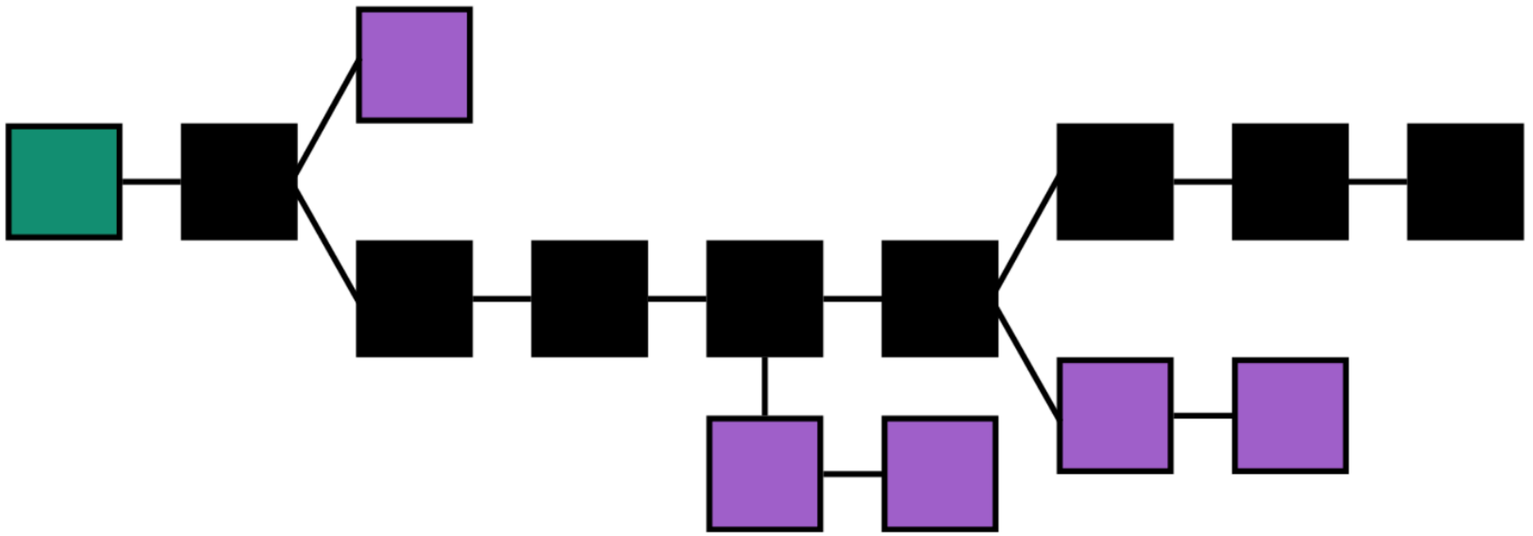


Image CC-BY-3 Theymos taken from the Bitcoin wiki

“Longest chain rule”

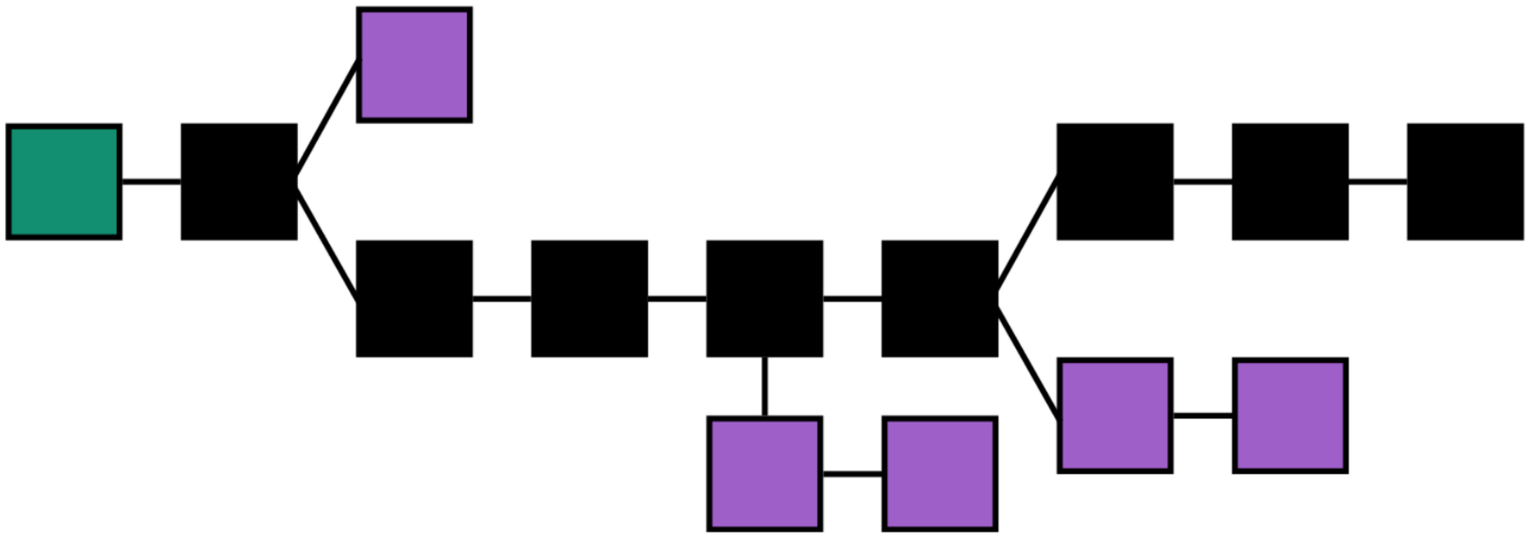


Image CC-BY-3 Theymos taken from the Bitcoin wiki

“Longest chain rule”

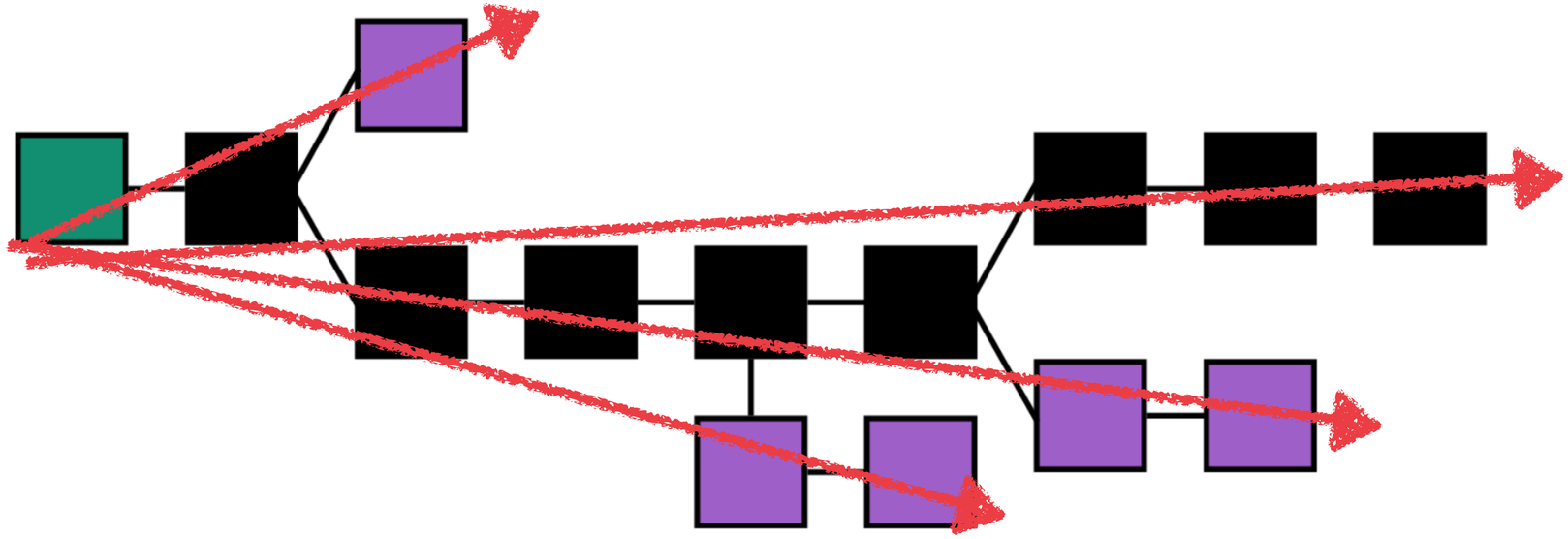


Image CC-BY-3 Theymos taken from the Bitcoin wiki

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