Speeding Up Bitcoin

David Tse

Padovani Lecture

NASIT 2021

June 25, 2021

Part I:
Introduction

Trust: a basic human need

- large-scale
- decentralized
- permissionless

Basis of trust

- An important basis of trust is a common immutable record of history that everyone can agree on.
- A challenge of achieving decentralized trust is how to maintain this record of history without a central authority.
- The heart of decentralized trust is a distributed consensus problem.

Decentralizing trust: two breakthroughs

2008

Bitcoin: A Peer-to-Peer Electronic Cash System

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

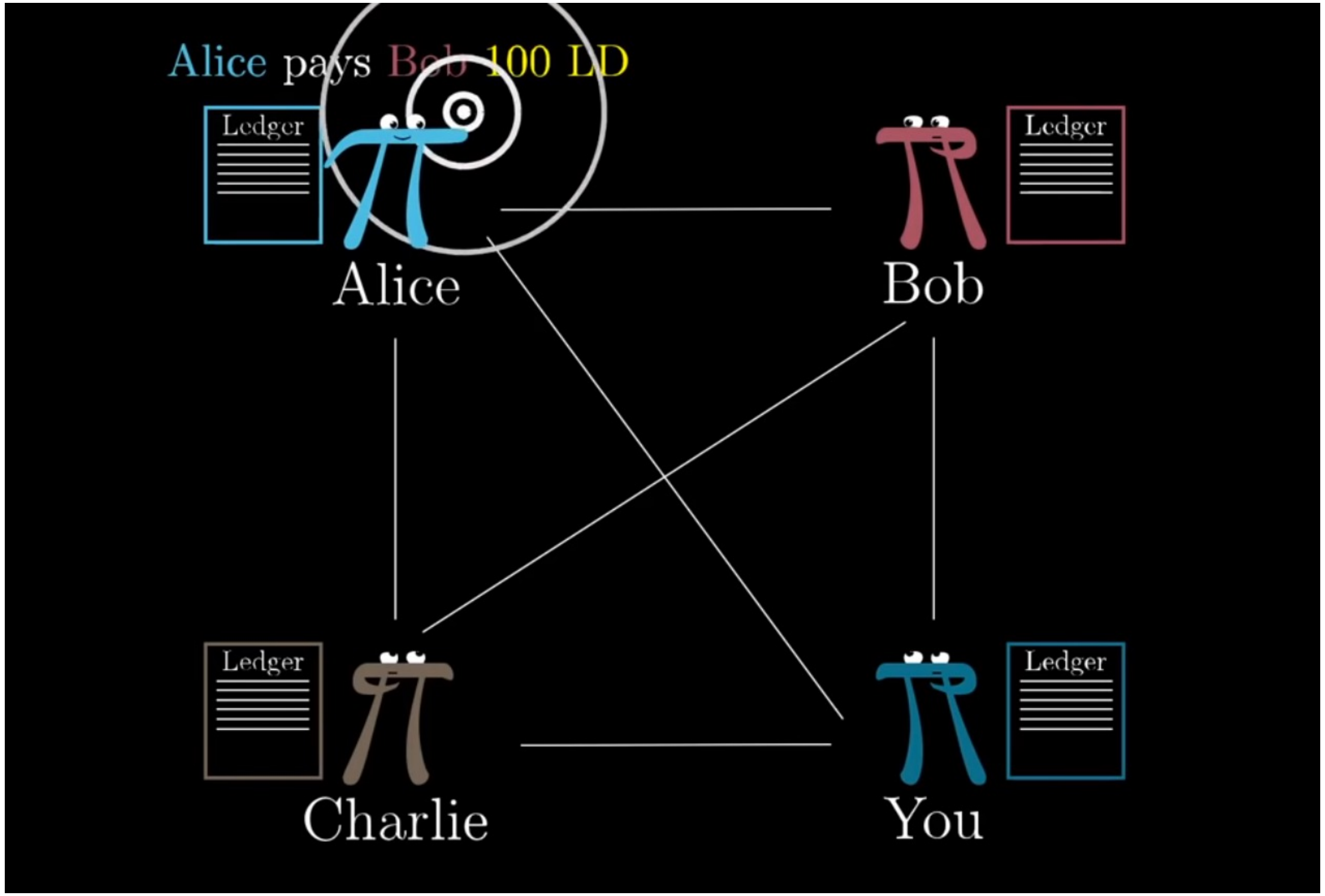
First system to achieve large-scale permissionless consensus

2013

Ethereum: The Ultimate Smart Contract and Decentralized Application Platform

Broaden from payments to other applications

Bitcoin: a decentralized ledger



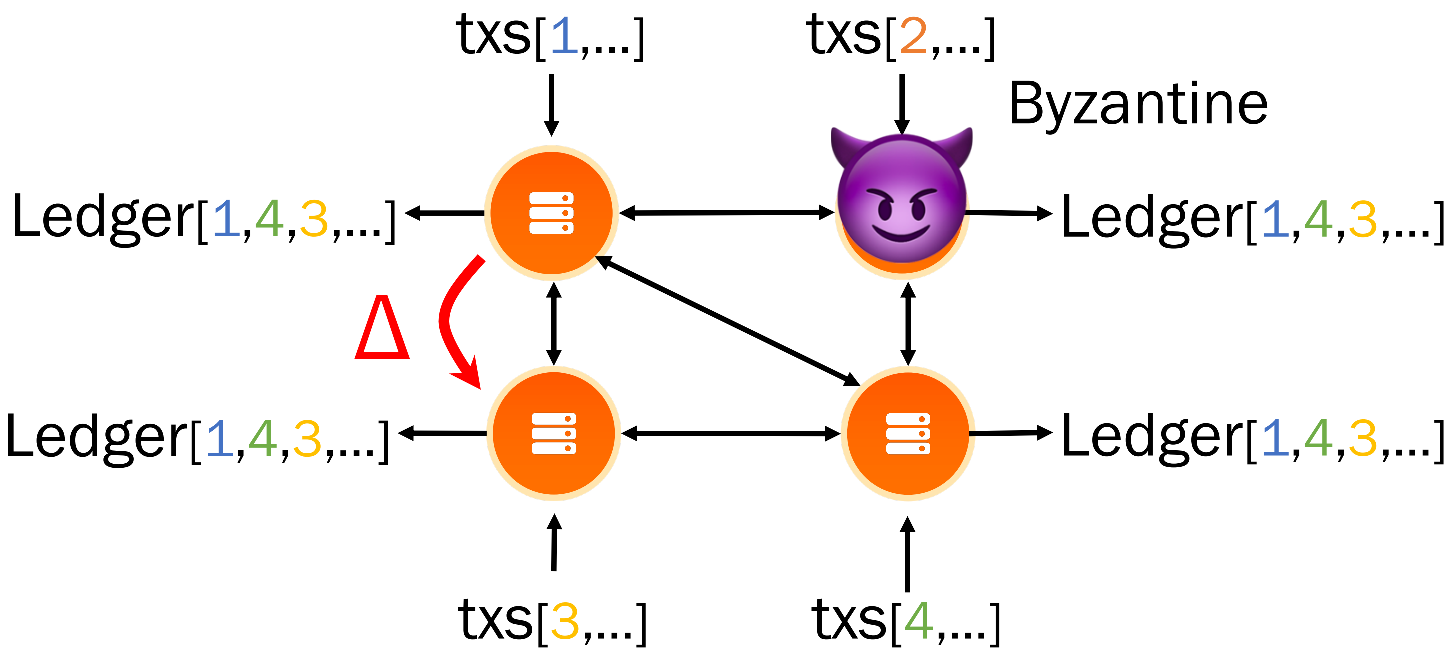
"But how does bitcoin actually work?", Youtube

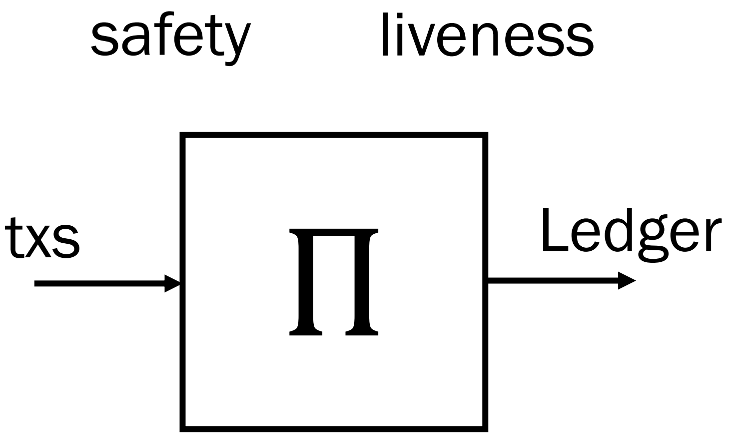
Core problem Bitcoin solved: consensus

- A new data structure: blockchain
- A new consensus protocol: proof-of-work longest chain protocol

The Byzantine consensus problem

Lamport, Pease, Shostak 1980, 1982





State machine replication

Typical theorem: a consensus protocol is safe and live when no more than f out of n nodes are Byzantine.

What's new about Bitcoin?

- Traditional consensus protocols are designed for a closed environment with a fixed set of permissioned nodes.

- Bitcoin is designed for an Internet-scale open environment where any node can join or leave at any time.

Permissionless dynamic participation

Theorem: Bitcoin is safe and live when no more than 50% of the online compute power is adversarial.

Bitcoin: Pros and Cons

Pros:

- permissionless
- dynamic participation
- Extremely simple protocol

Cons:

- high consumption of energy (~ Sweden)
- low transaction throughput (7 transactions per second)
- high confirmation latency (hours)
- No accountability
- Insecure under network partition.

Questions we will answer

- How does Bitcoin work?

- How do we formalize safety and liveness and how do we prove that Bitcoin is secure?

- Why does Bitcoin have very bad latency?

- How to speed up Bitcoin while keeping its security properties?

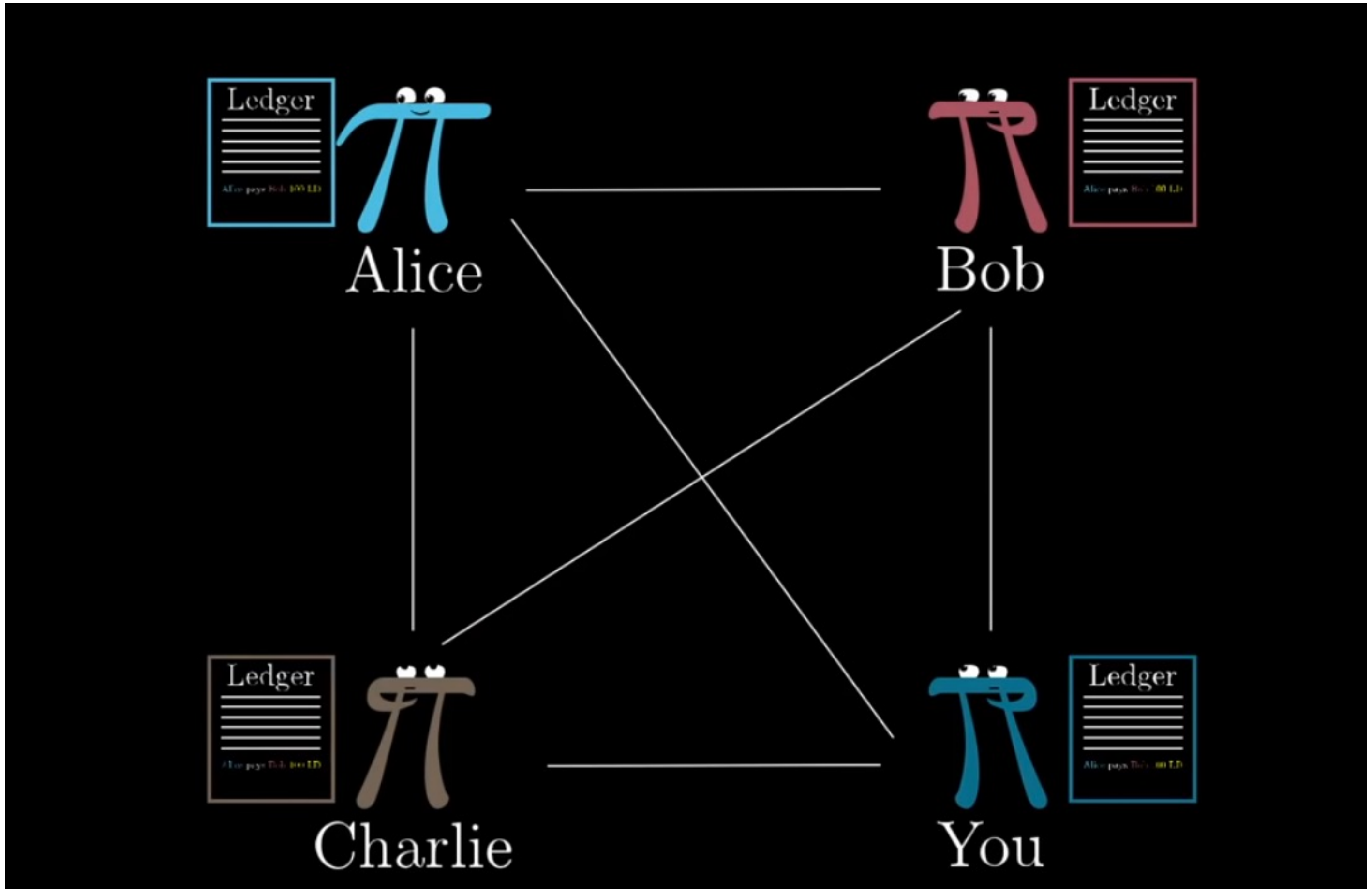
Part II:
Bitcoin and its Security

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.

Ledger



Data integrity and data agreement

- Data integrity: Data is legit.
  - Solved by digital signatures.

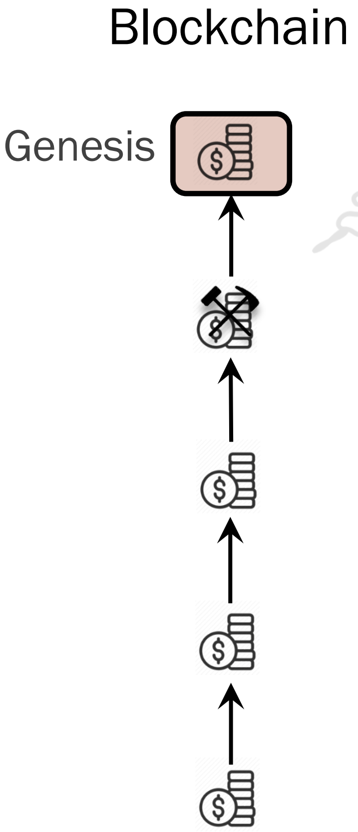
- Data agreement: among all nodes and across time.
  - This is the “double spending” problem and is solved by a consensus protocol.

Protocol

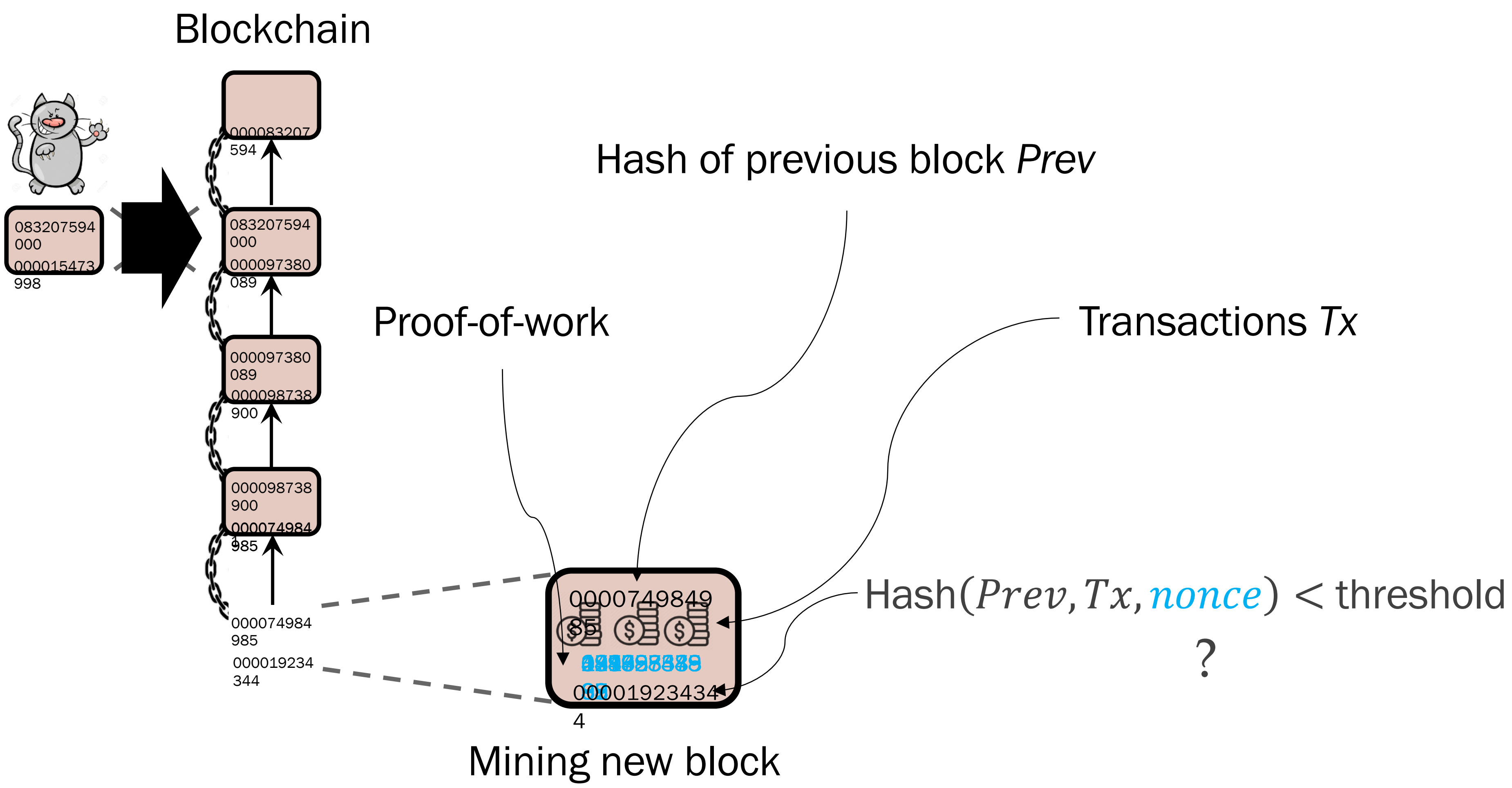
- Mining rule (encoder)
- Confirmation rule (decoder)

Mining rule

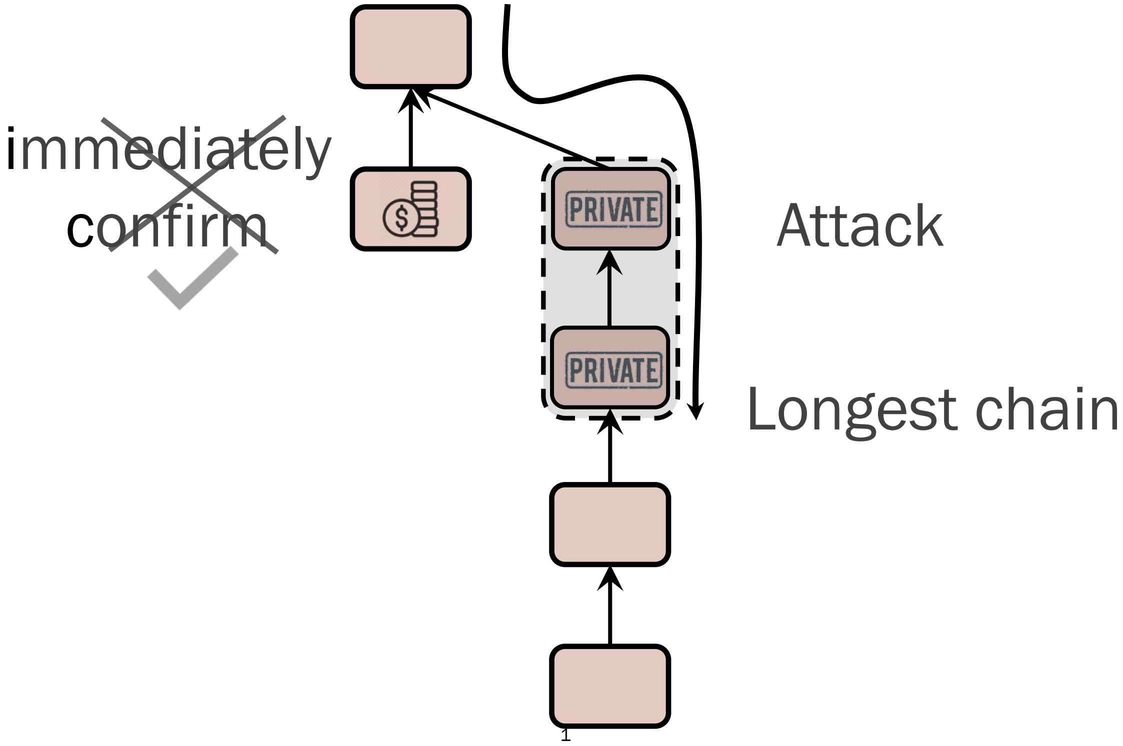
- Mining on the longest chain
- Poisson arrivals of blocks
- Average mining rate λ = 1 block per 10 min



Proof-of-Work



Instant confirmation

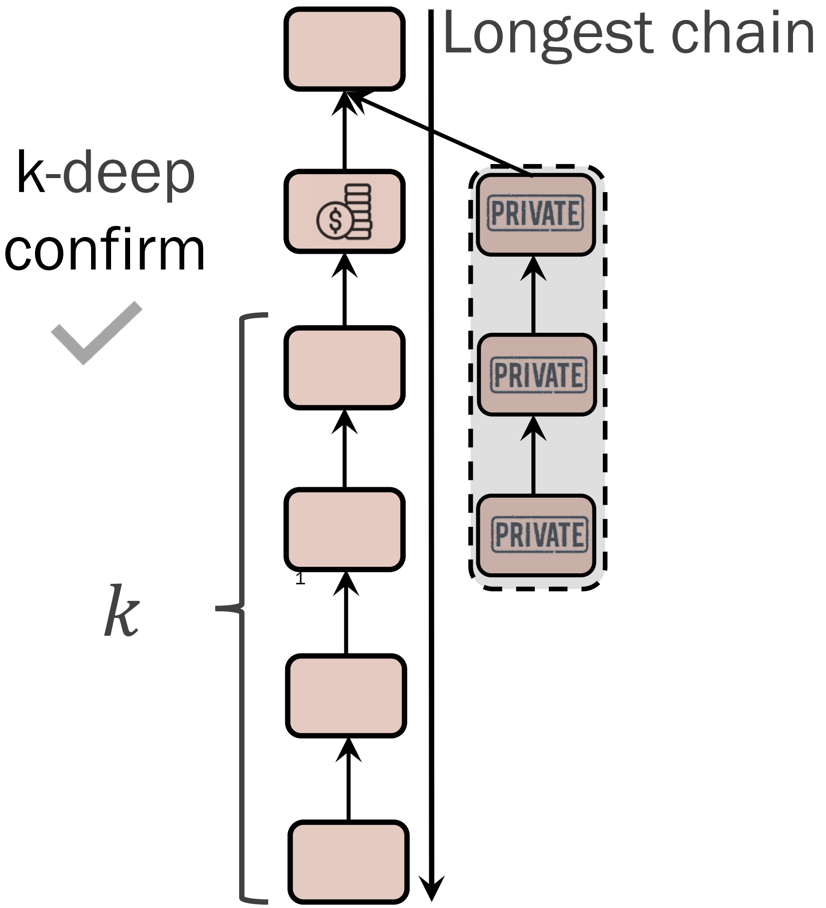


k-deep confirmation

30% adversary power

k=0 $\varepsilon = 1.0000000$
k=5 $\varepsilon = 0.1773523$
k=10 $\varepsilon = 0.0416605$
k=15 $\varepsilon = 0.0101008$
k=20 $\varepsilon = 0.0024804$
k=25 $\varepsilon = 0.0006132$
k=30 $\varepsilon = 0.0001522$
k=35 $\varepsilon = 0.0000379$
k=40 $\varepsilon = 0.0000095$
k=45 $\varepsilon = 0.0000024$
k=50 $\varepsilon = 0.0000006$

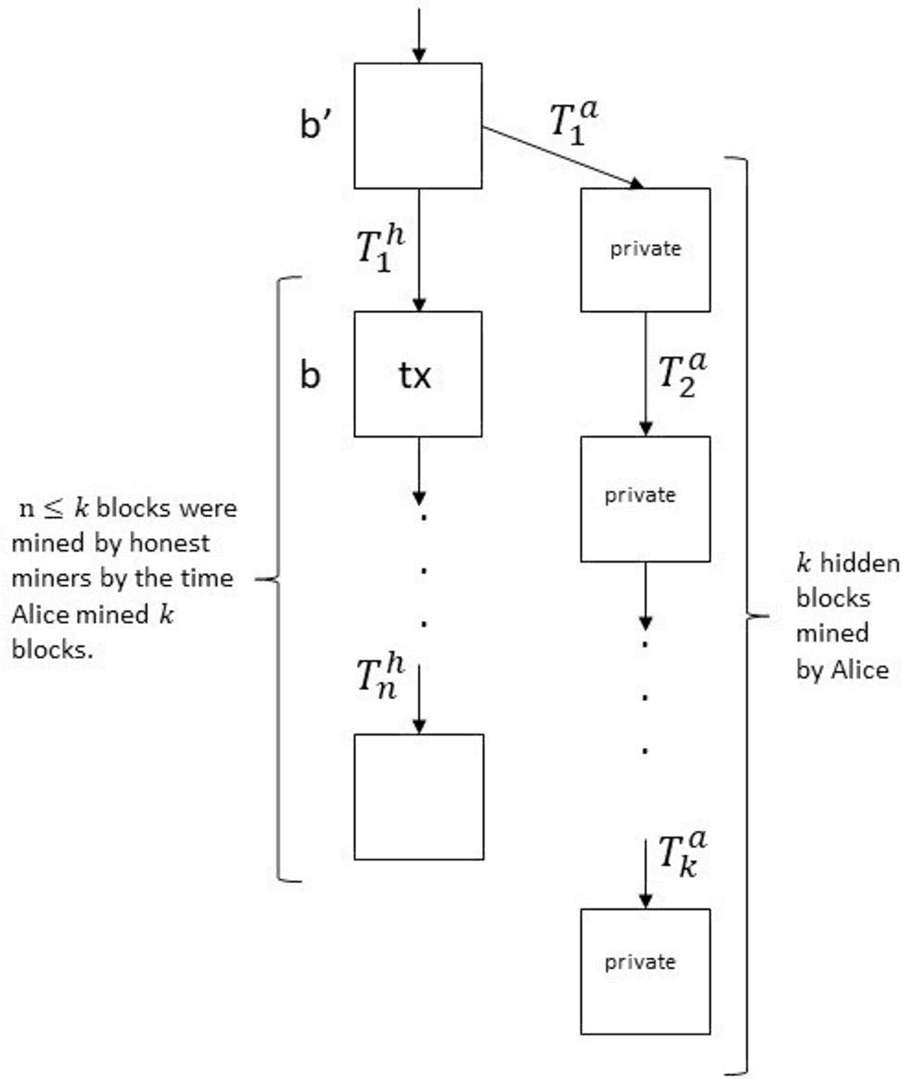
Nakamoto's table



Notations

- total mining rate λ
- honest mining rate λ_h
- adversarial mining rate λ_a
- adversarial fraction β
- network delay bound Δ. (assumed 0 for now)

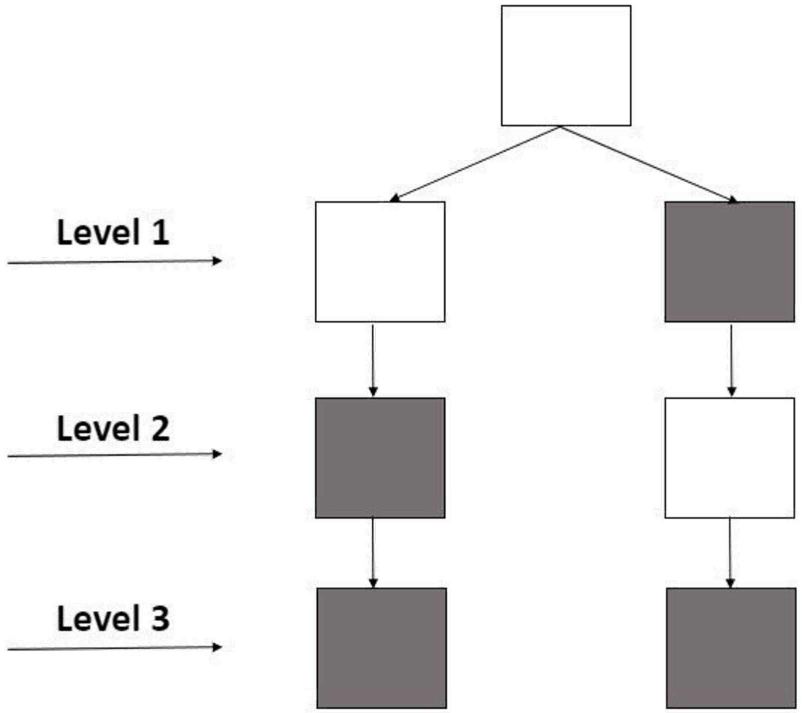
Private attack analysis



Safety

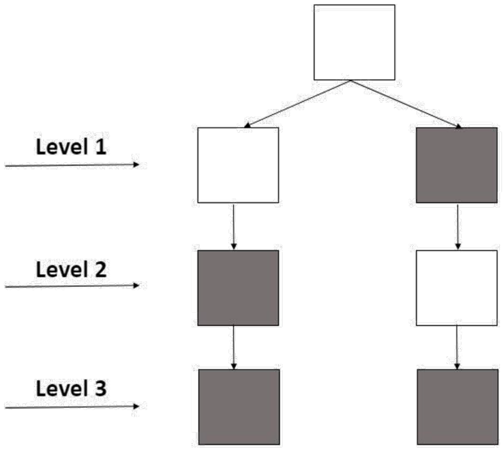
- A block b is safe if once it is confirmed, it remains on the ledger in the view of any node at any future time……..regardless of the adversary's attack.
- What we showed is safety with high probability……..under a specific attack, the private attack.
- What about other attacks?

Example: balance attack

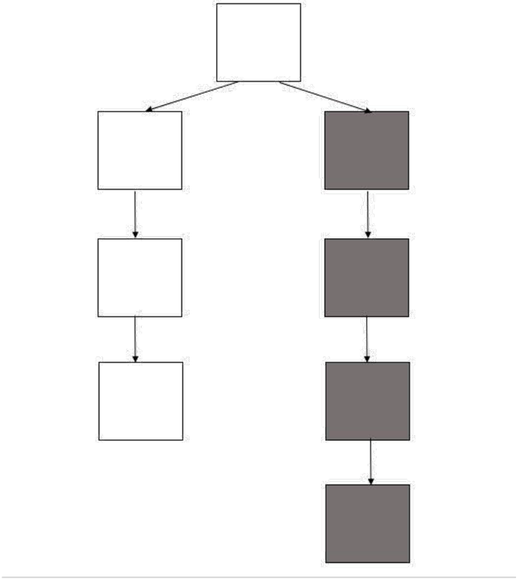


Safety analysis (for level 1 block)

any successful attack



private attack



Safety theorem for Bitcoin

Theorem:

Each Bitcoin block is safe with probability of confirmation error going to zero exponential in k if $\lambda_a < \lambda_h$

Is Safety Enough?

- What happens if no honest blocks are confirmed?

- We need liveness.

- A protocol is live if a non-zero fraction of honest blocks are confirmed.

Bitcoin: chain growth and chain quality

Bitcoin: liveness theorem

Theorem:

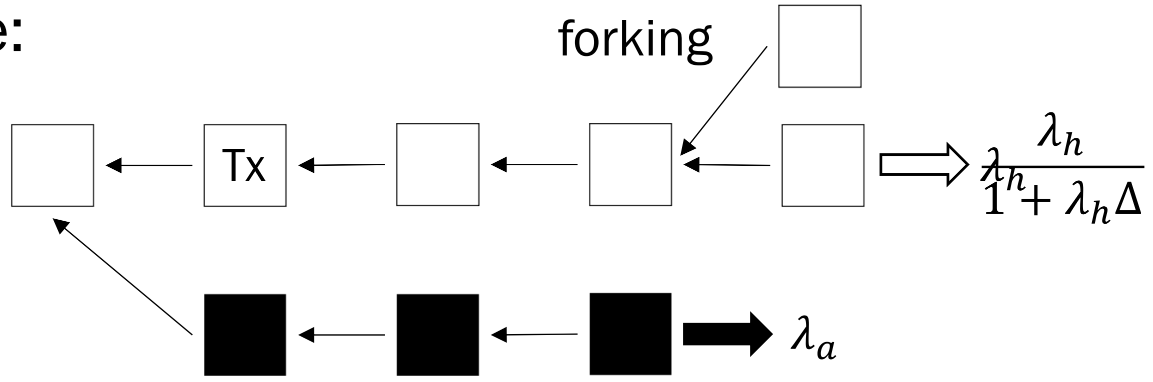
Bitcoin is live if $\lambda_a < \lambda_h$

Network delay

- So far we have assumed communication of blocks happen instantaneously.
- But real networks have delays.
- Synchronous model: communication of all blocks is delayed by at most Δ seconds.

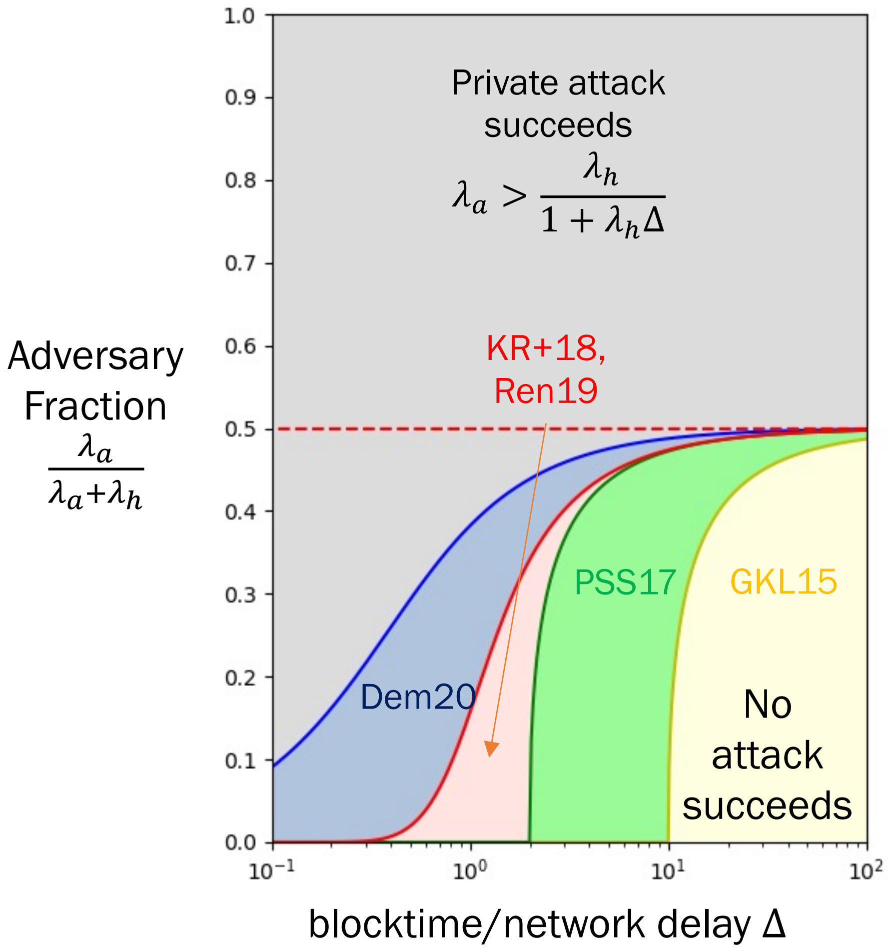
Private attack analysis

The race:



Private attack succeeds if $\lambda_a > \frac{\lambda_h}{1+\lambda_h\Delta}$

What about all attacks?



Part III:
Speeding up Bitcoin

V. Bagaria, S. Kannan, D.T., G. Fanti, P. Viswanath, "Prism: Deconstructing the blockchain to approach physical limits ", ACM CCS'19.

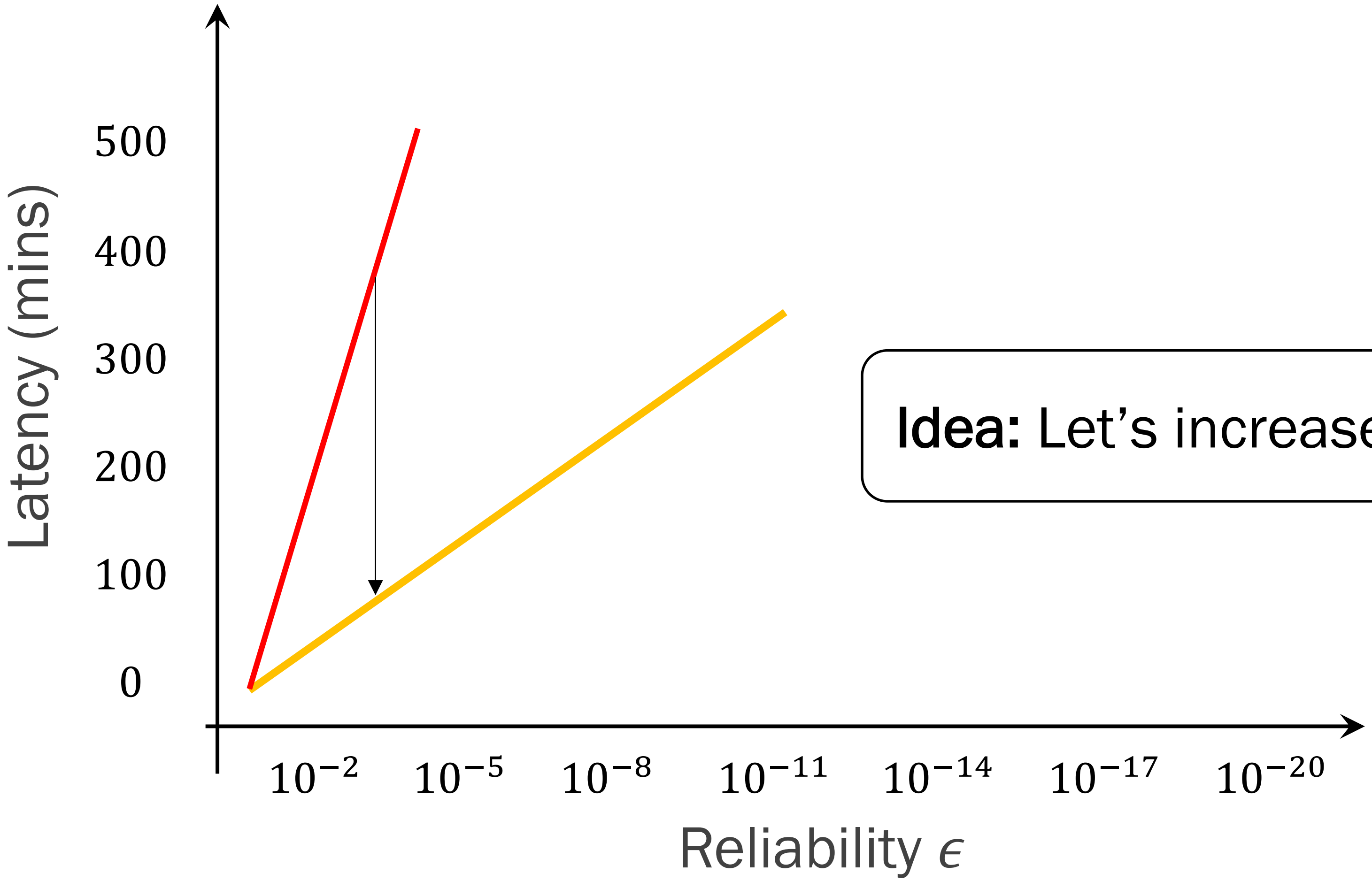
Confirmation latency

30% adversary power

k=0 $\varepsilon = 1.0000000$
k=5 $\varepsilon = 0.1773523$
k=10 $\varepsilon = 0.0416605$
k=15 $\varepsilon = 0.0101008$
k=20 $\varepsilon = 0.0024804$
k=25 $\varepsilon = 0.0006132$
k=30 $\varepsilon = 0.0001522$
k=35 $\varepsilon = 0.0000379$
k=40 $\varepsilon = 0.0000095$
k=45 $\varepsilon = 0.0000024$
k=50 $\varepsilon = 0.0000006$

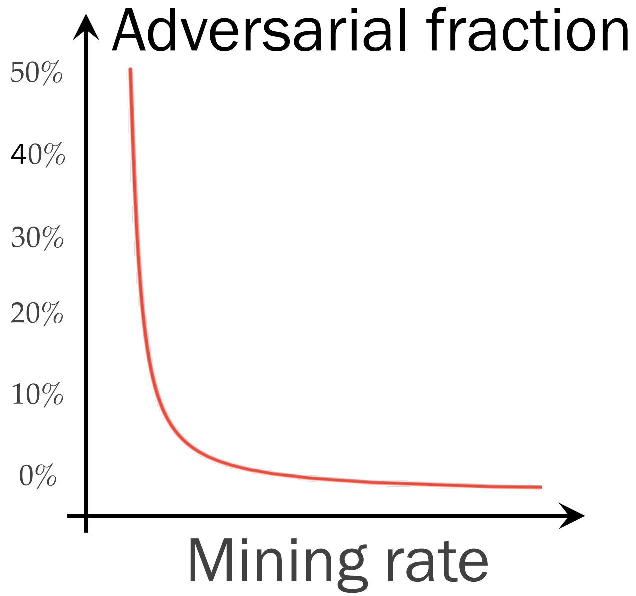
Nakamoto's table

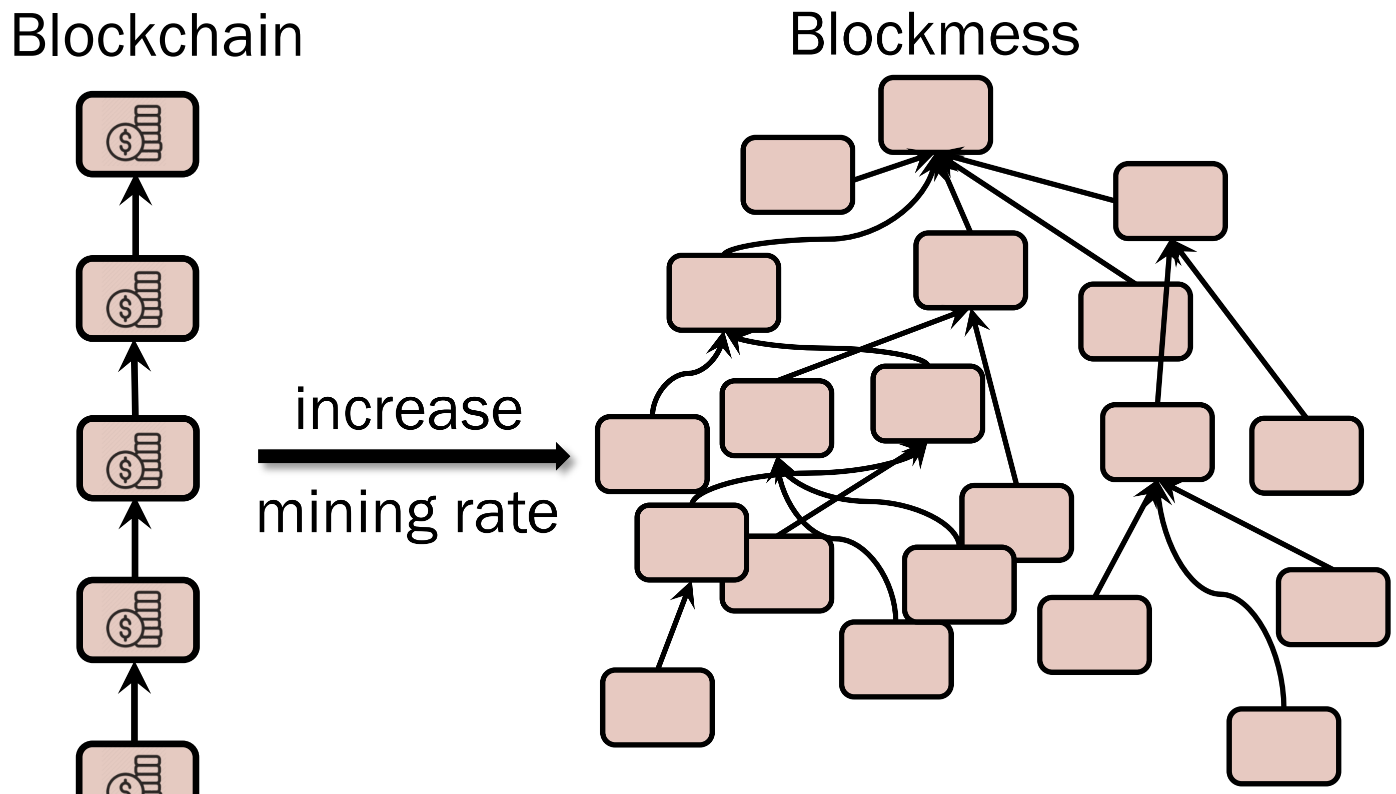
Mining rate
λ = 1 block/10 min



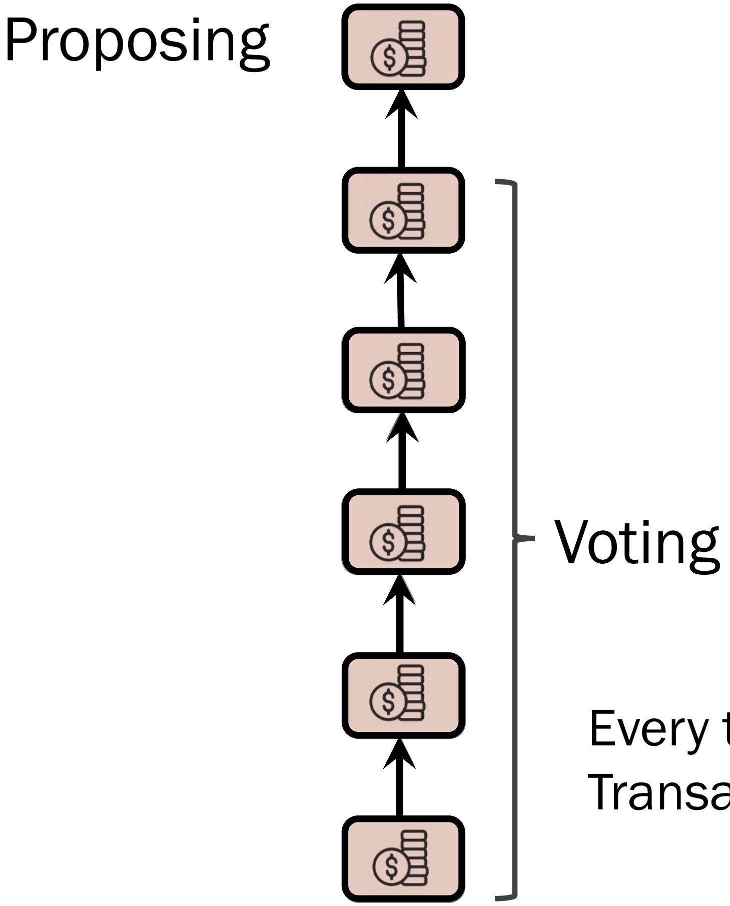
Idea: Let's increase λ!

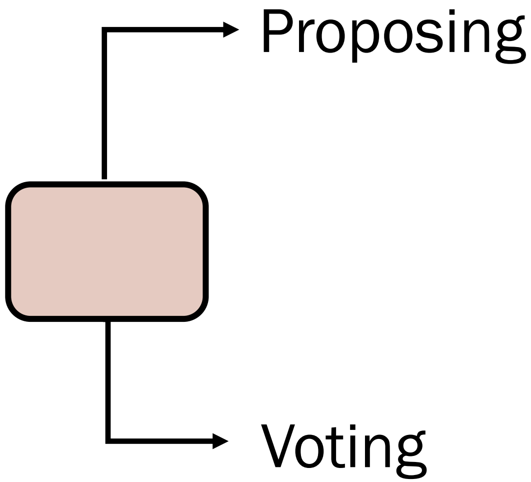
Scale the mining rate





2 roles of a Bitcoin block

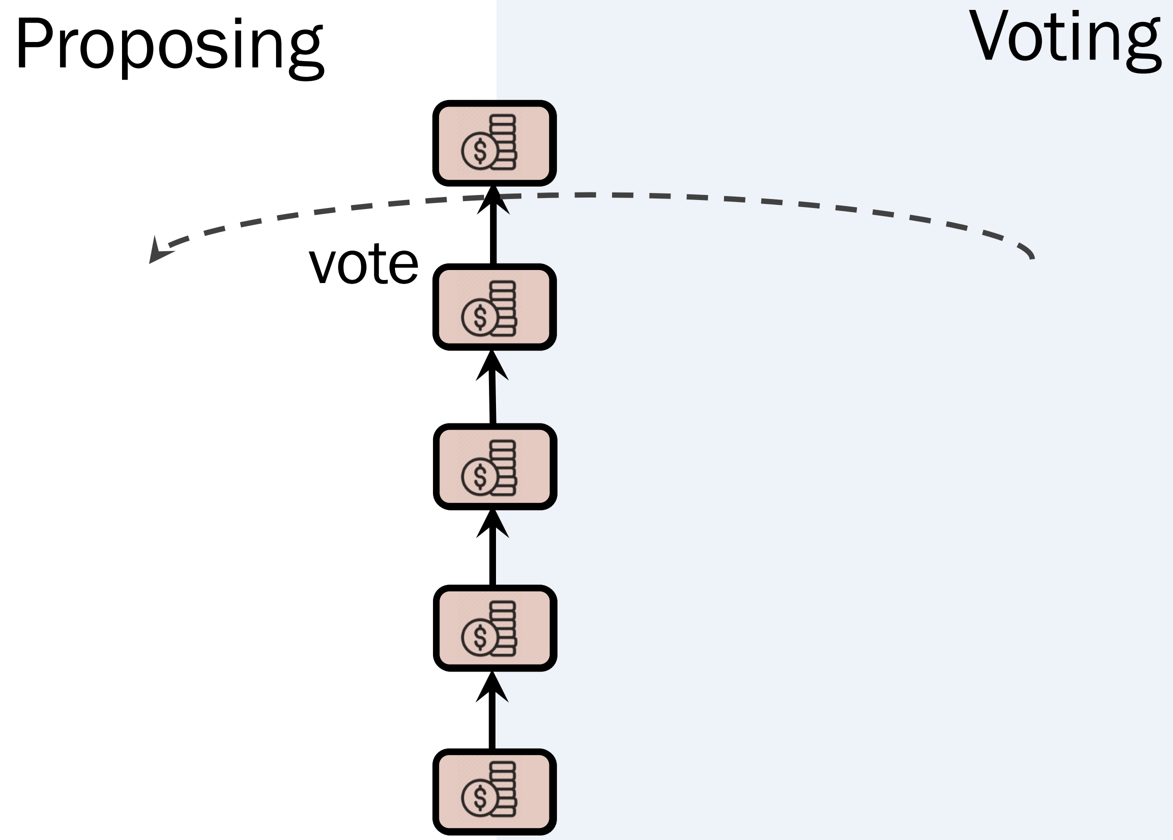




Every time a block is mined, it simultaneously propose new Transactions and vote for previous blocks.

Deconstruct Bitcoin, and scale.

Bitcoin → Deconstruct



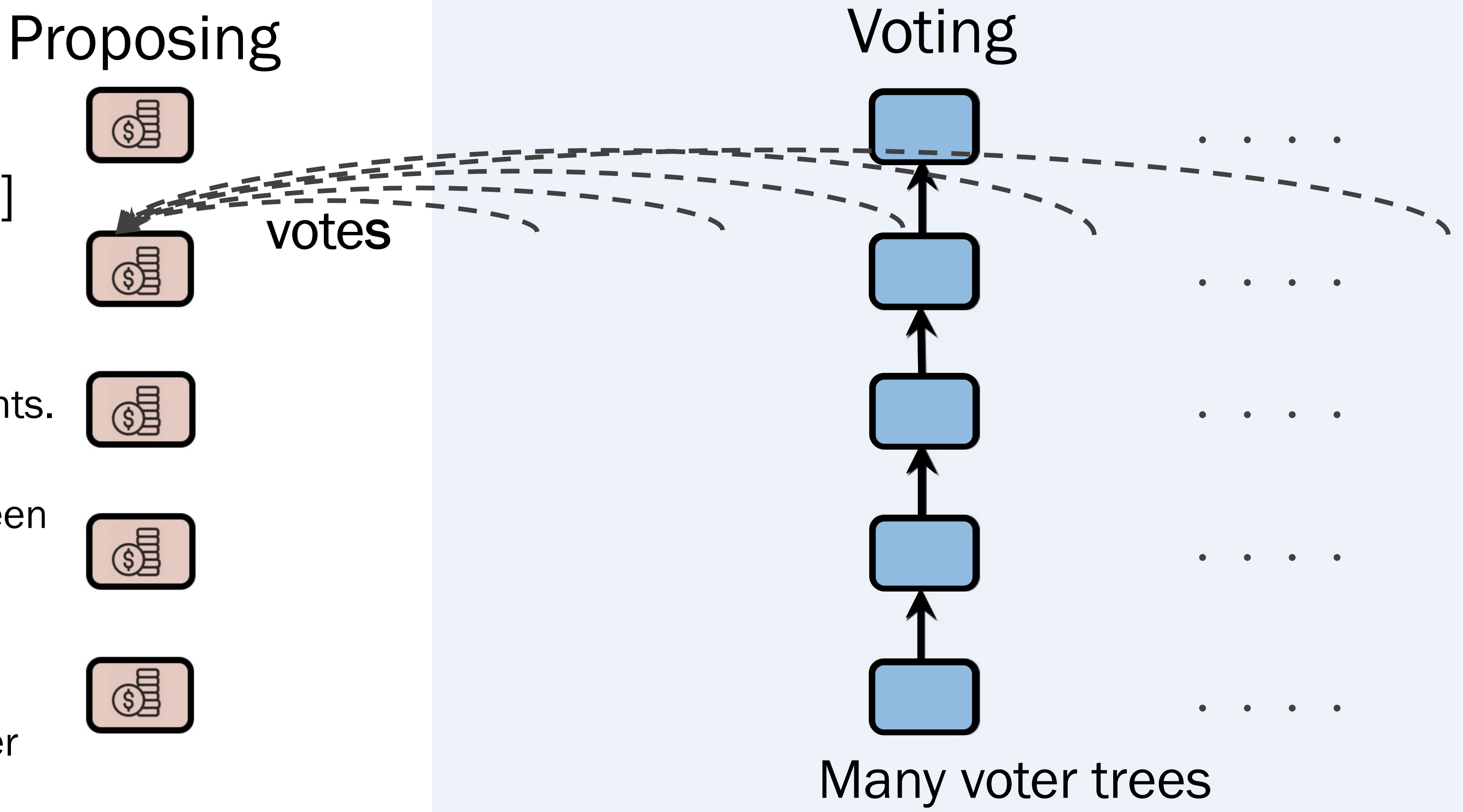
Bitcoin → Deconstruct → Scale

Many parallel PoW lotteries:

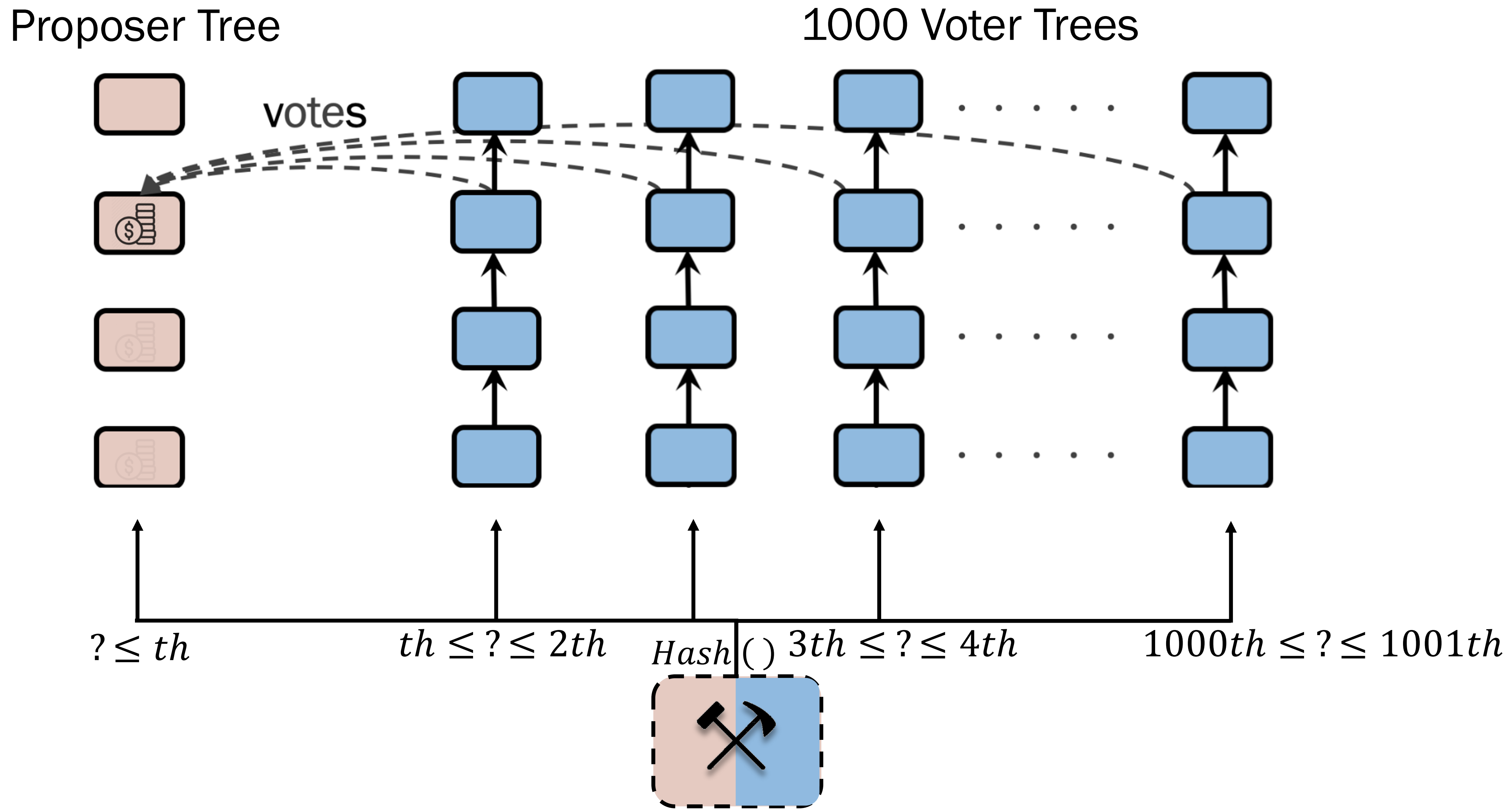
2-for-1 mining [GKL15,PS17]

Proposing and voting:

1. Segregated by proposer block heights.
2. Each voter tree votes for the first seen proposer block at each height,
3. Only votes from main chains count.
4. For each height choose the proposer block with most votes.



Prism: Mining



Theorems

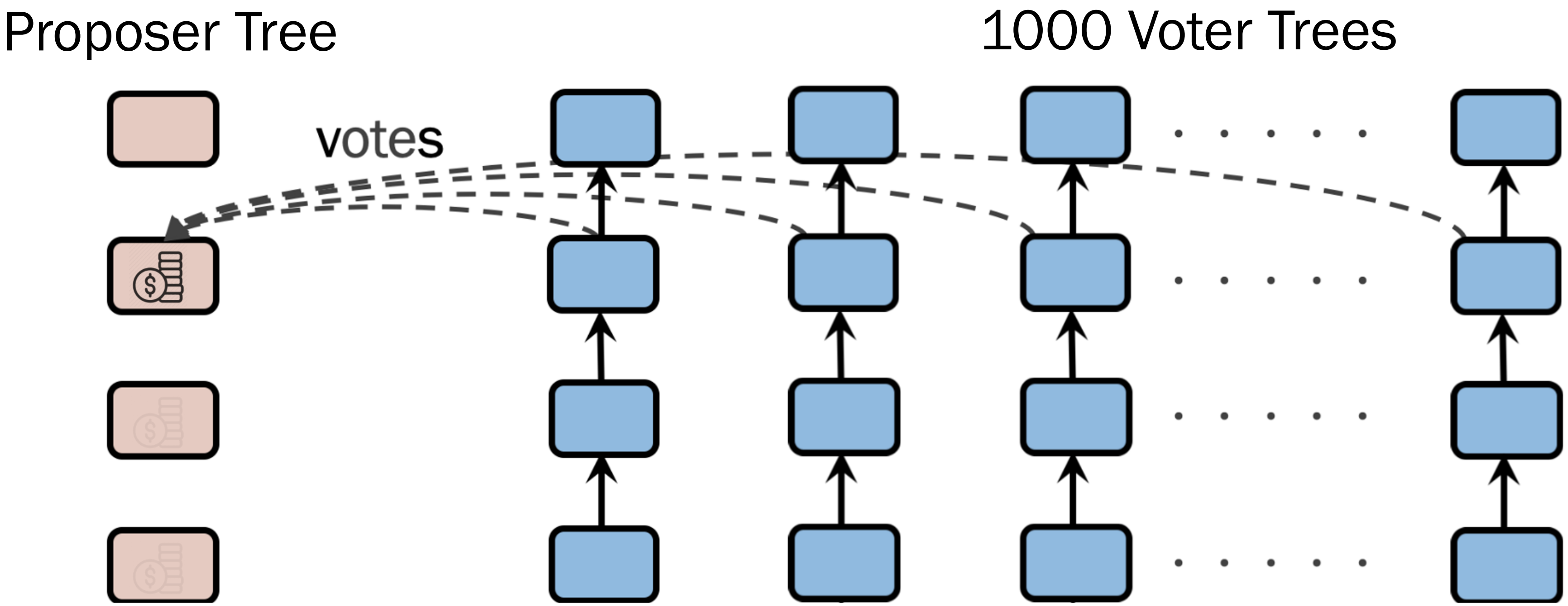
Security

Prism achieves safety and liveness against an adversary with less than 50% of total hash power.

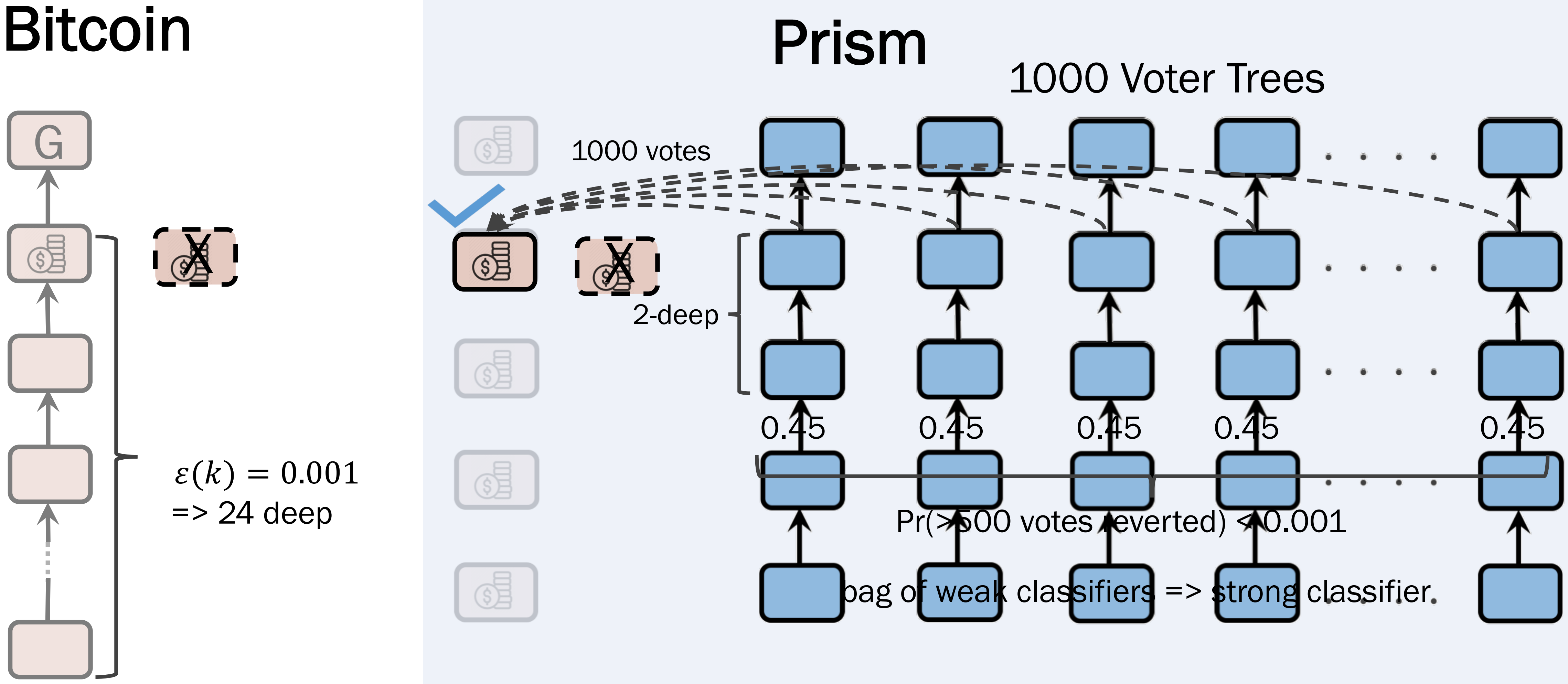
Normal-path Latency

With probability $1 - \epsilon(m)$, *Prism* confirms transactions with constant average latency, independent of m, # of voter chains.

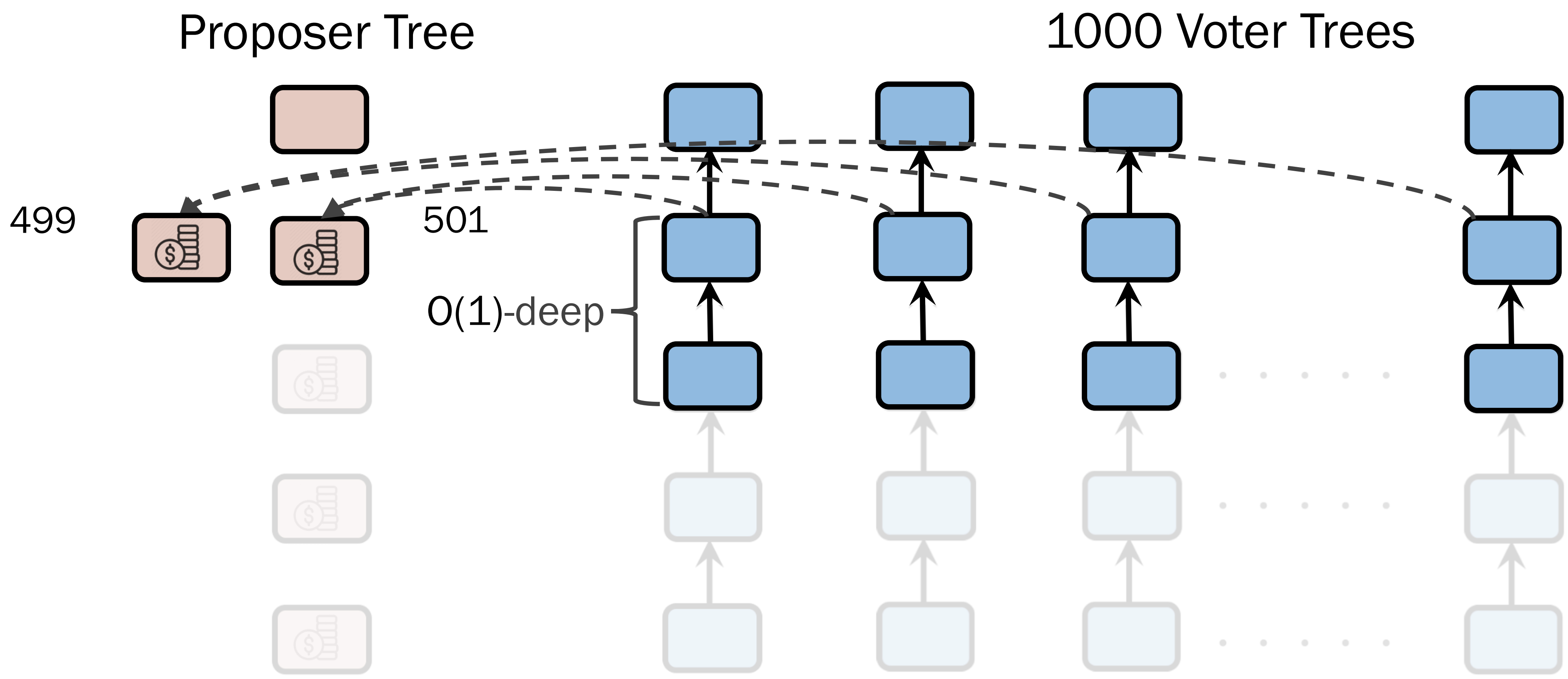
Prism: Safety and Liveness



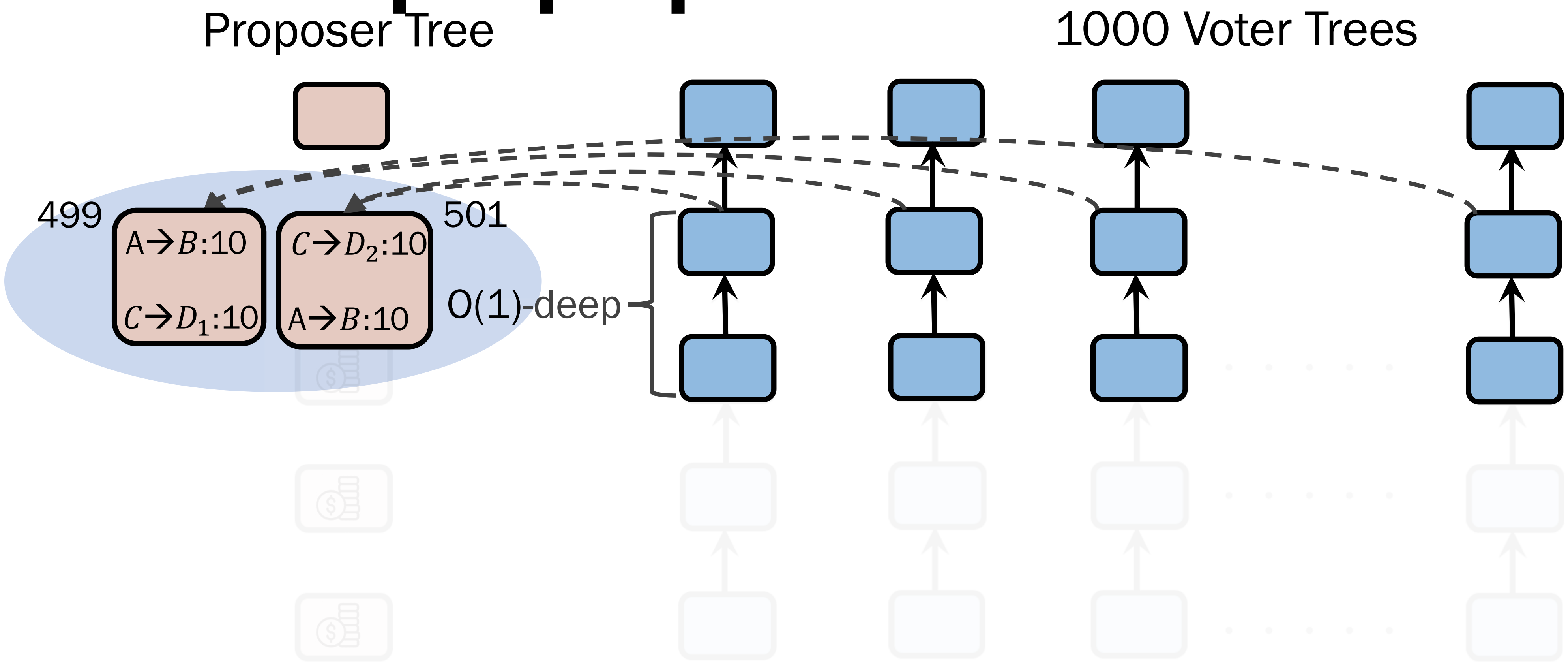
Prism: fast confirmation



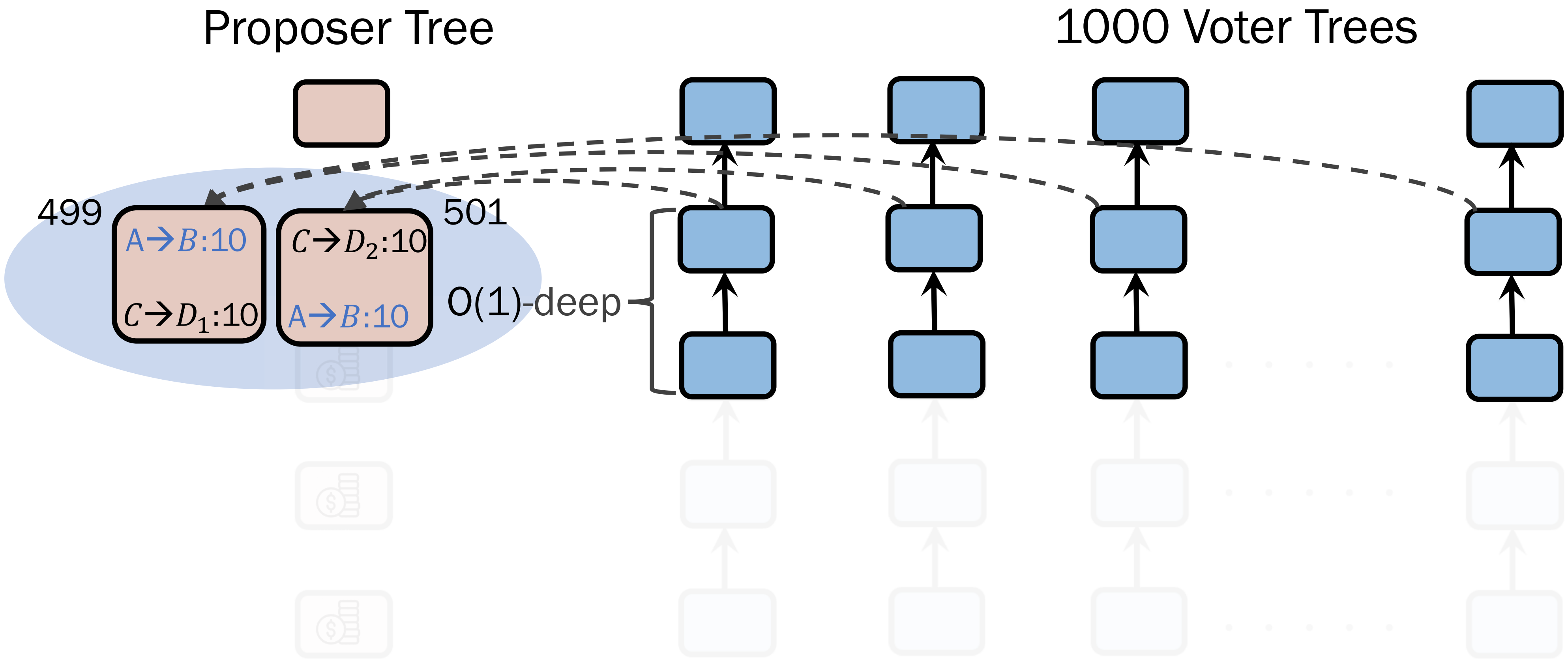
Multiple proposer blocks



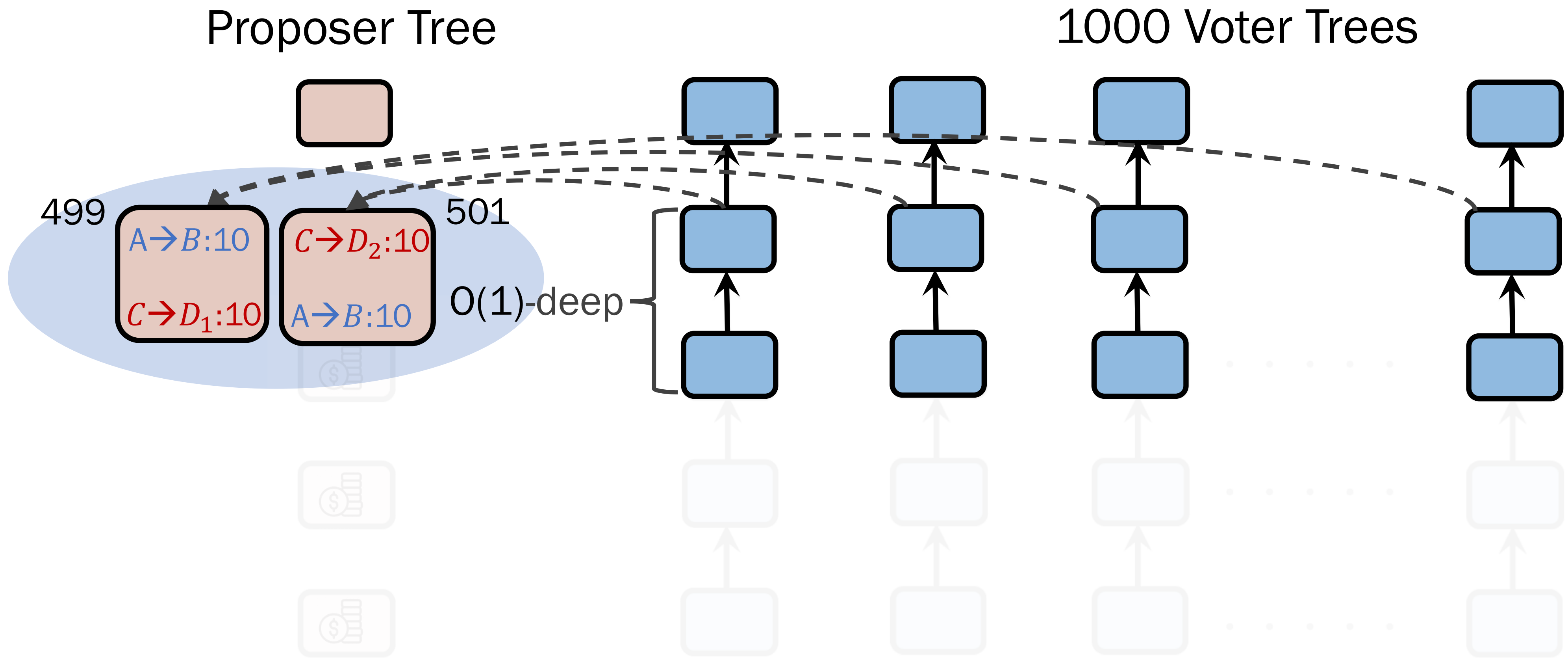
Multiple proposer blocks



Multiple proposer blocks

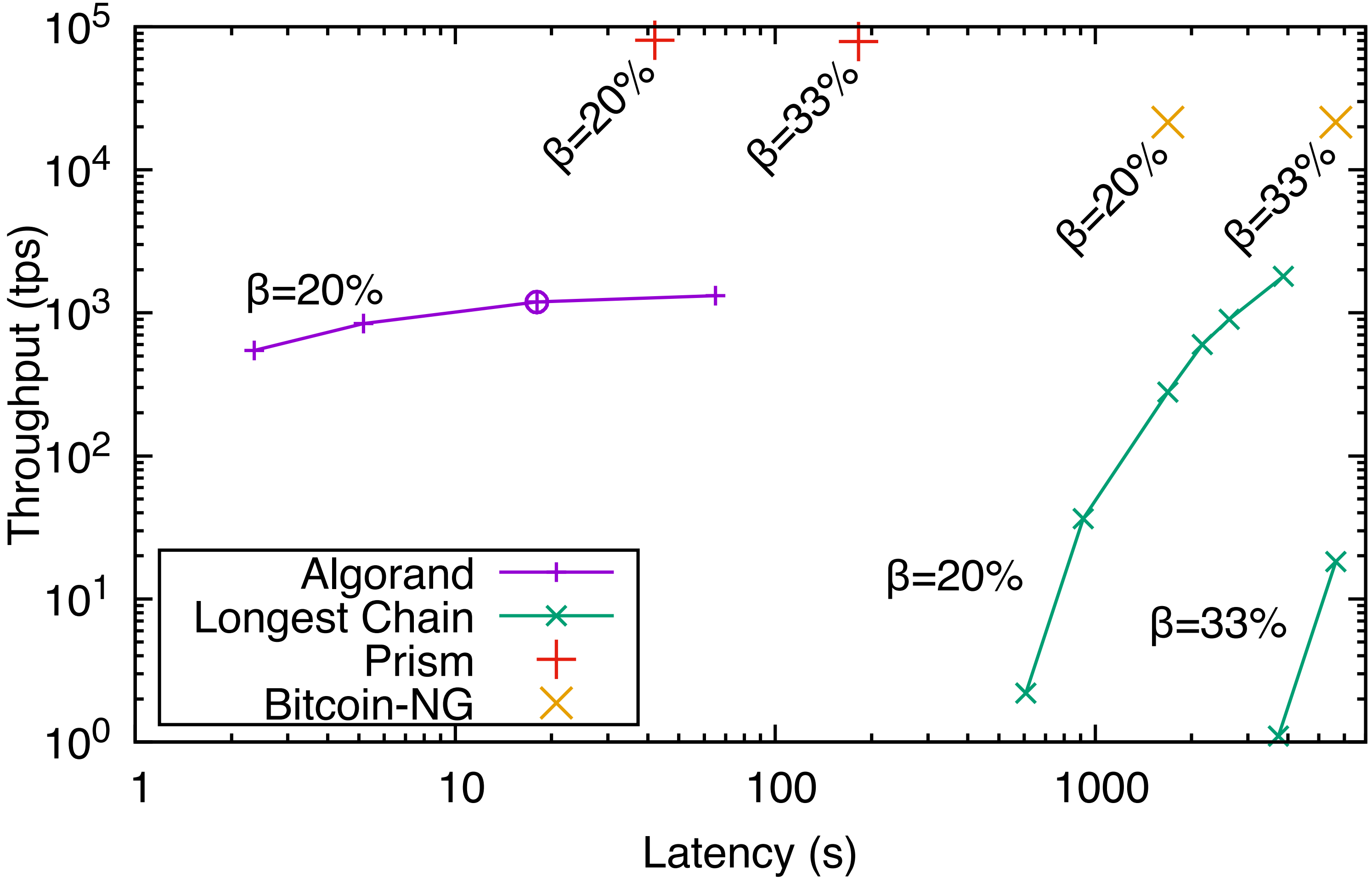


Multiple proposer blocks



Rust Implementation

4-regular topology of 100 EC2 c5d.4xlarge instances, 120ms delay, 400 Mbps bandwidth per link.



Yang et al, "Prism: Scaling Bitcoin 10,000 X", arXiv:1810.08092