Lecture 9: Scaling Latency

https://web3.princeton.edu/principles-of-blockchains/

Professor Pramod Viswanath
Princeton University

This lecture:
Improving latency of Bitcoin

Three Modules

- Bitcoin (lectures 2-7)
- Scaling Bitcoin (lectures 8-14)
- Beyond Bitcoin (lectures 15-20)

Scaling Bitcoin

- Scaling Bitcoin (lecture 8-14)
- L1 Scaling: Improve Bitcoin performance while still retain basic structure of the longest chain protocol
  - Throughput (#8)
  - Latency (#9)
  - Storage & compute (#10) -- Sharding
  - Energy (#11) – Proof of Stake
- L2 Scaling: Improve performance via an “overlay” on Bitcoin
  - Payment Channels (#12)
  - Data Availability (#13)
  - Rollups: Optimistic and Cryptographic proofs of compressed ledgers (#14)

Bitcoin latency

Time from when a transaction was broadcast until the transaction is confirmed in the ledger

- τ_1: Time from when a transaction was broadcast until the transaction is put into a mined block B
- τ_2: Time from when the transaction was put into a mined block B until block B is k-deep in the longest chain

$$\tau = \tau_1 + \tau_2$$

τ_2 is the real bottleneck, depends on how large k is.

Bitcoin latency

Assume low forking ($\lambda\Delta \ll 1$),

$$\tau = \frac{k}{(1-\beta)\lambda}$$

Depth of blocks

Block arrival rate

From Lecture 6, error probability

$$\epsilon = e^{-ck}$$

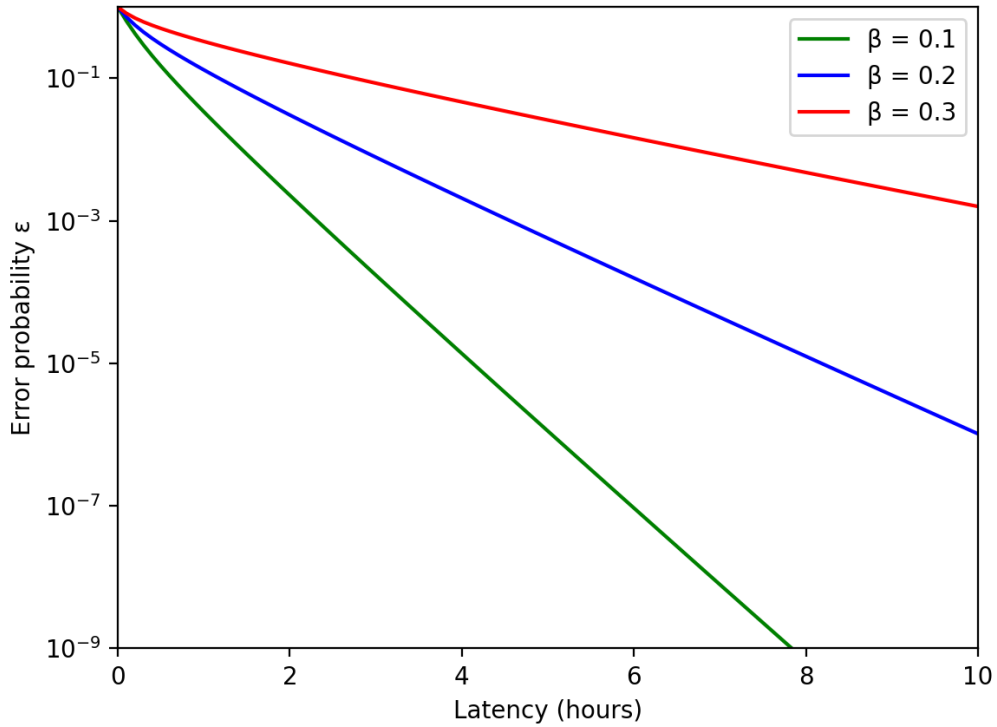
$$\tau = \frac{\frac{1}{c}\log(\frac{1}{\epsilon})}{(1-\beta)\lambda} = O(\frac{1}{\lambda}\log(\frac{1}{\epsilon}))$$

Latency and security are coupled

Bitcoin latency

$$\tau = O(\frac{1}{\lambda}\log(\frac{1}{\epsilon}))$$

Bitcoin: $\frac{1}{\lambda}$ = 10 minutes



Improve Bitcoin latency

Only way to improve latency is to

- reduce k; but this reduces security
- Increase λ; but this also reduces security

Ethereum: $\frac{1}{\lambda}$ = 15s; $k = 100$

- latency = 25 minutes
- Way better than Bitcoin performance; improvement simply by picking better parameters.

Improve Bitcoin latency

Question: can we make relatively small changes to the longest chain protocol and PoW mining while scaling latency?

Key Requirement:

- Do not want latency to depend on security level
- Decouple security from latency

Prism

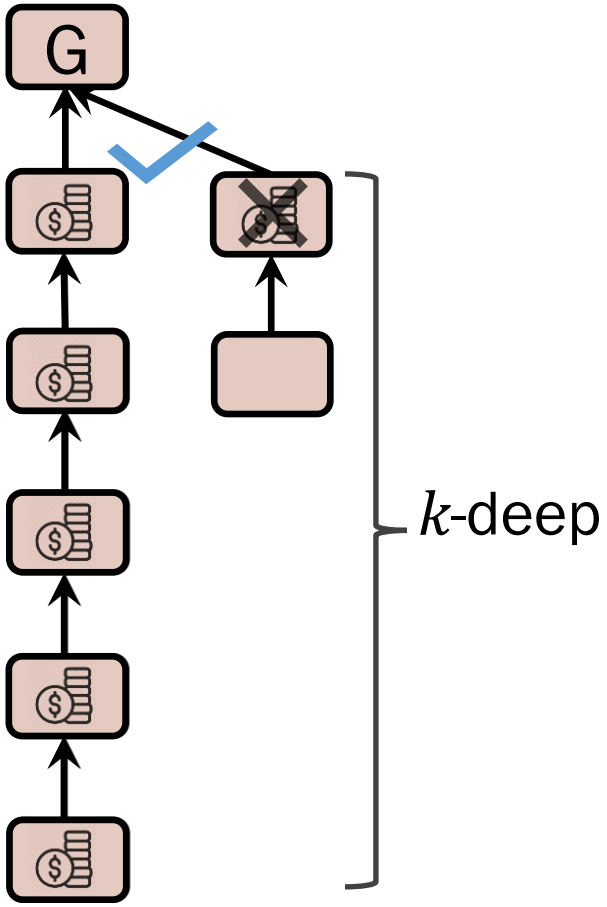
Prism achieves optimal latency

- **Decoupling principle**: separate performance from security

- Prism 1.0 achieves optimal throughput; last lecture

Decoupling voting

k-deep confirmation rule is a form of voting



Satoshi's Table

```
q=0.3
z=0     P=1.0000000
z=5     P=0.1773523
z=10    P=0.0416605
z=15    P=0.0101008
z=20    P=0.0024804
z=25    P=0.0006132
```

~~1 deep~~ => .45

25 deep => 0.0006

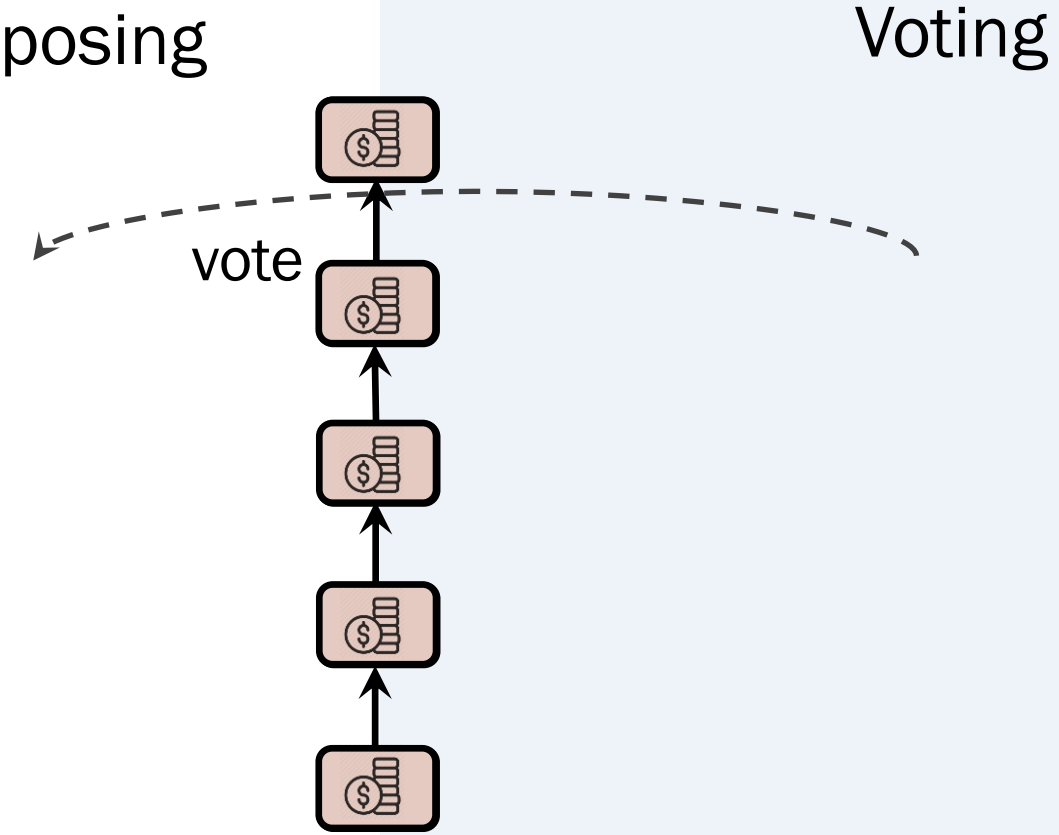
Can think of one block = one vote underneath B

k-deep = k votes in sequence

Really need k large to sample the miners

Bitcoin → Deconstruct

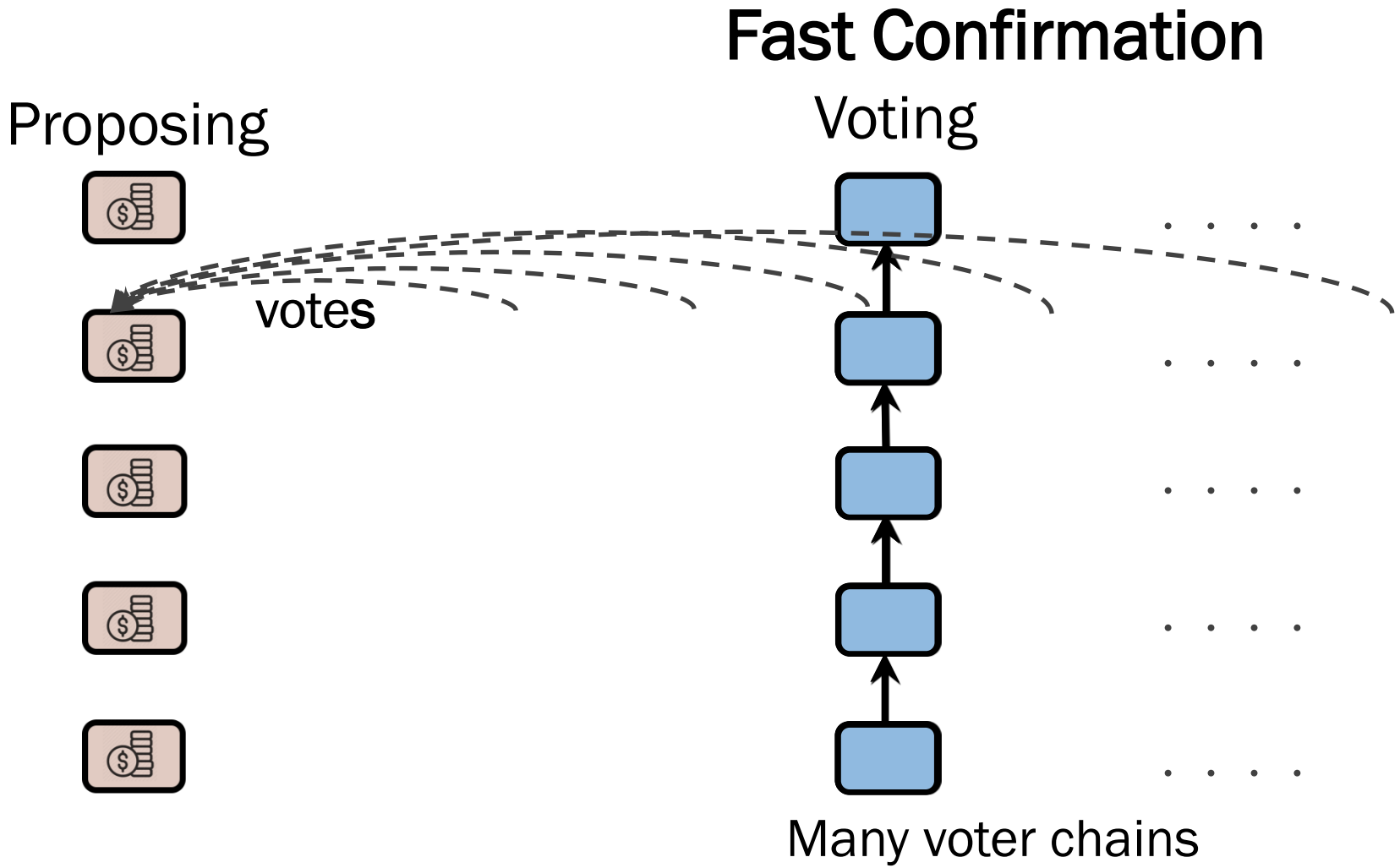
Ledger construction



1. Select votes along longest voter chain

2. Order the proposer blocks by votes

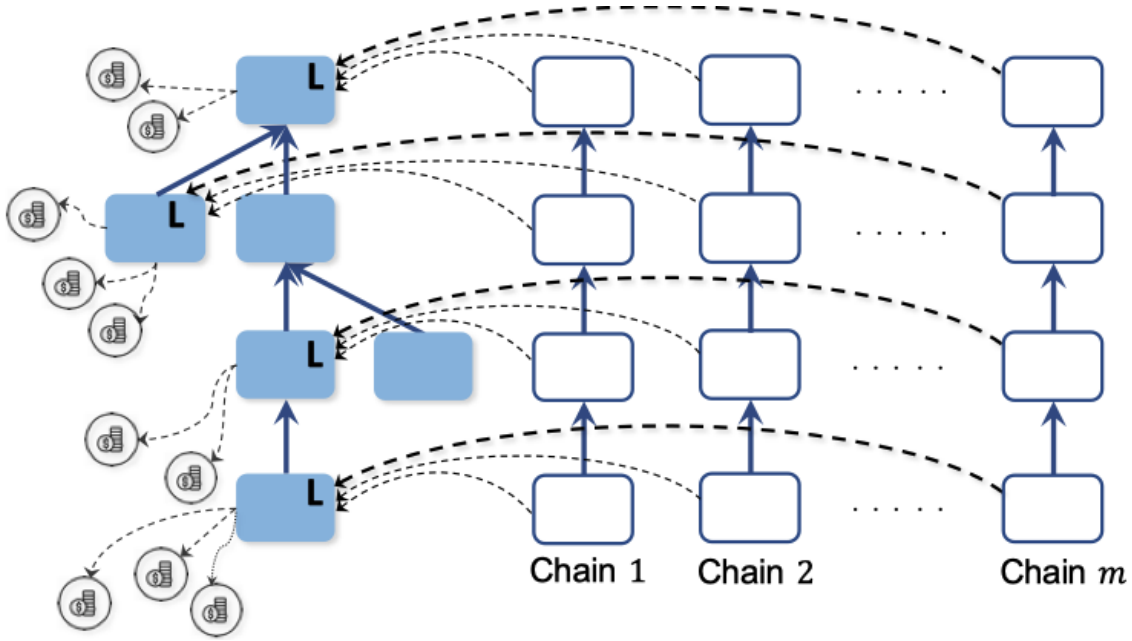
Bitcoin → Deconstruct →Prism



Ledger Construction: For each level choose the proposer block with **maximum** votes

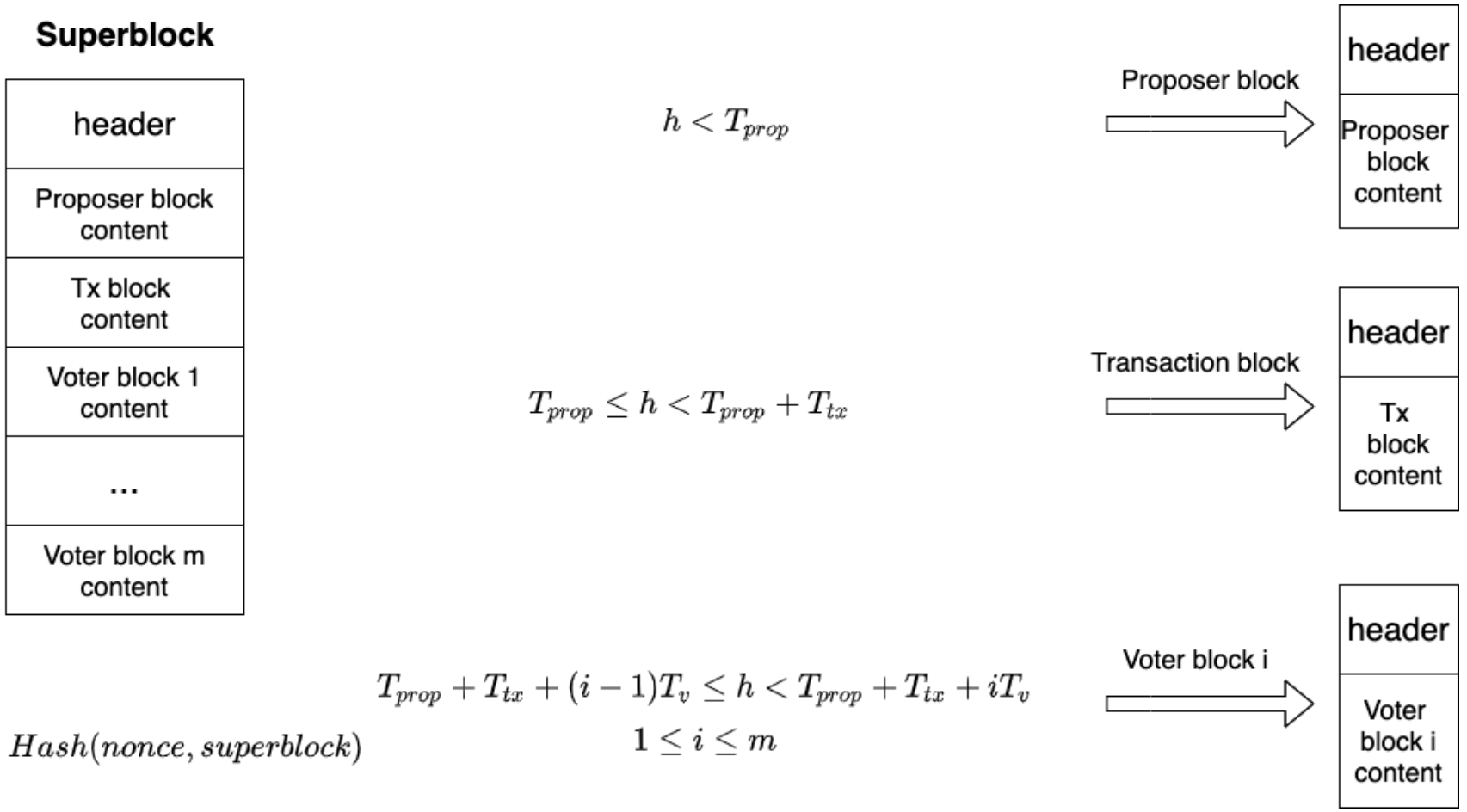
Prism

- Proposal rule: longest chain
- Voting rule:
  a) each voter chain votes for one and only one proposer block at each level
  b) each voter block votes for all the proposer levels that have not been voted by its parent.
- Mining rule: honest miner picks to be proposer/voter/transaction block at random



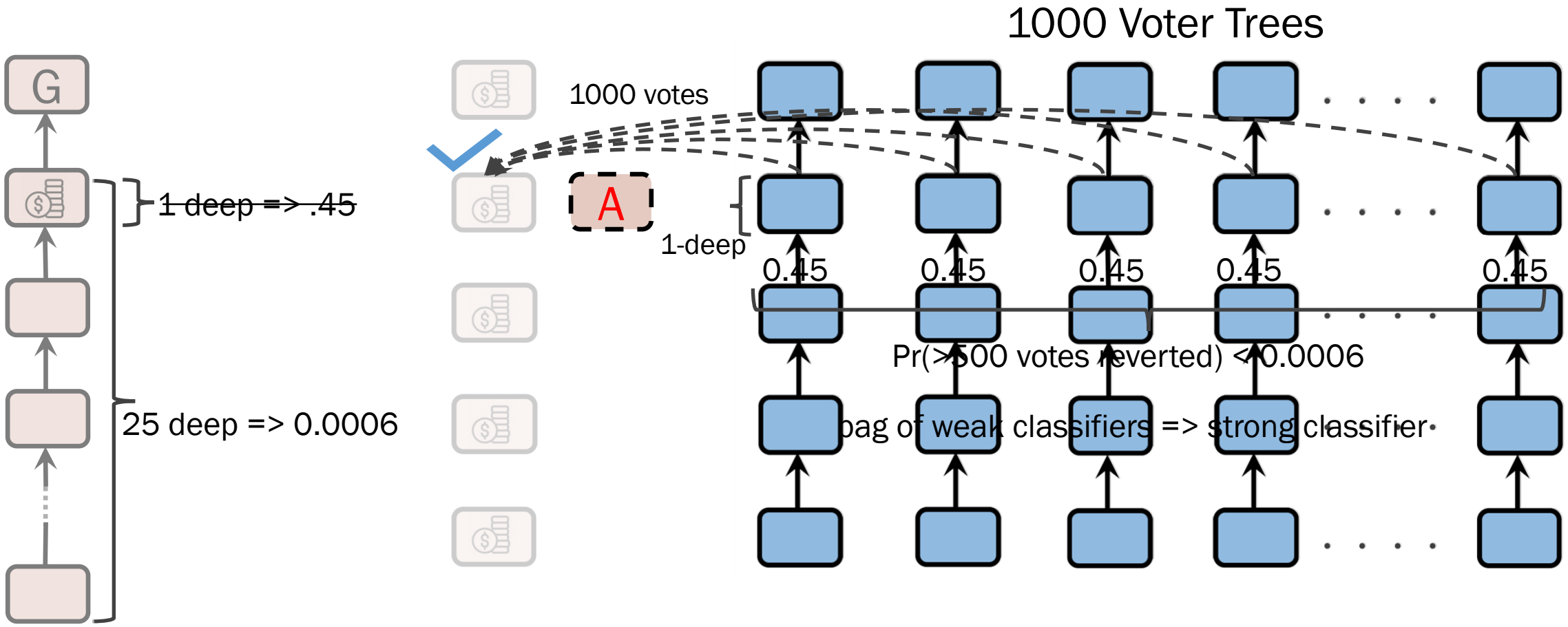
Cryptographic sortition

How do you prevent adversary from m focusing its mining power on a specific type of blocks or on a specific voter chain?



Fast confirmation

Bitcoin



Ledger Construction: For each level choose the proposer block with **maximum** votes