CS251 Fall 2022

(cs251.stanford.edu)

Incentives and Accountability in Consensus: Proof-of-Stake

Ertem Nusret Tas

Recap of the Last Lecture

- Sybil Attack
- Sybil Resistance: Proof-of-Work, Proof-of-Stake, and Proof-of-Space.
- Bitcoin and Nakamoto Consensus
- Consensus in the Internet Setting
- Security for Bitcoin: Nakamoto’s Private Attack and Forking

Incentives in Bitcoin

How does Bitcoin *incentivize* miners to participate in consensus and mine new blocks?

- Block rewards
- Transaction fees

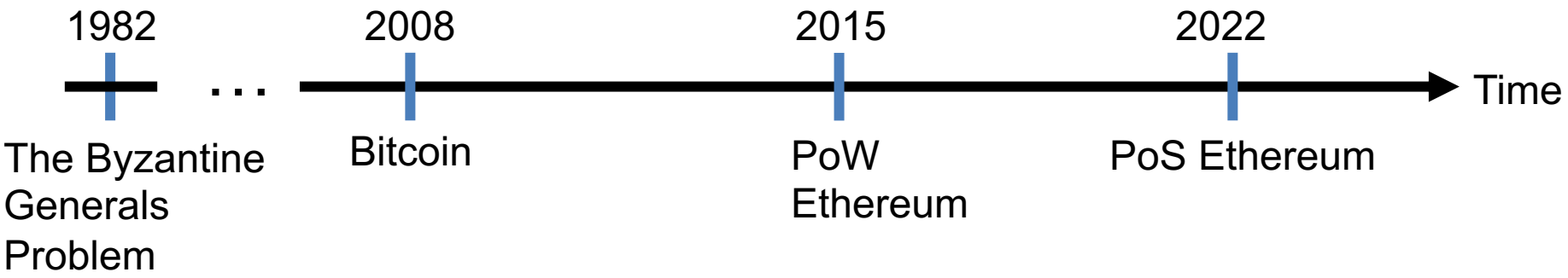
How does a miner capture these rewards?

- The first transaction in a Bitcoin block is called the **coinbase transaction**.
- The coinbase transaction can be created by the miner.
- Miner uses it to collect the block reward and the transaction fees.

Can these *incentives* guarantee *honest* participation?

- Not necessarily!
- Selfish mining attack!
- (See the optional slides if interested in the details.)

From Bitcoin to Proof-of-Stake



Open Participation
- Dynamic availability
- Sybil resistance

Block rewards (carrot)

PoS Ethereum:
Open Participation
- Dynamic availability
- Sybil resistance

Block rewards (carrot)
Finality and accountable safety
Slashing (stick)

The Byzantine Generals Problem (1982)
Bitcoin: A Peer-to-Peer Electronic Cash System (2008)
Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform. (2015)
Combining GHOST and Casper (2020)

A few words on Proof-of-Stake

In a Proof-of-Stake protocol, nodes lock up (i.e., stake) their coins in the protocol to become eligible to participate in consensus.

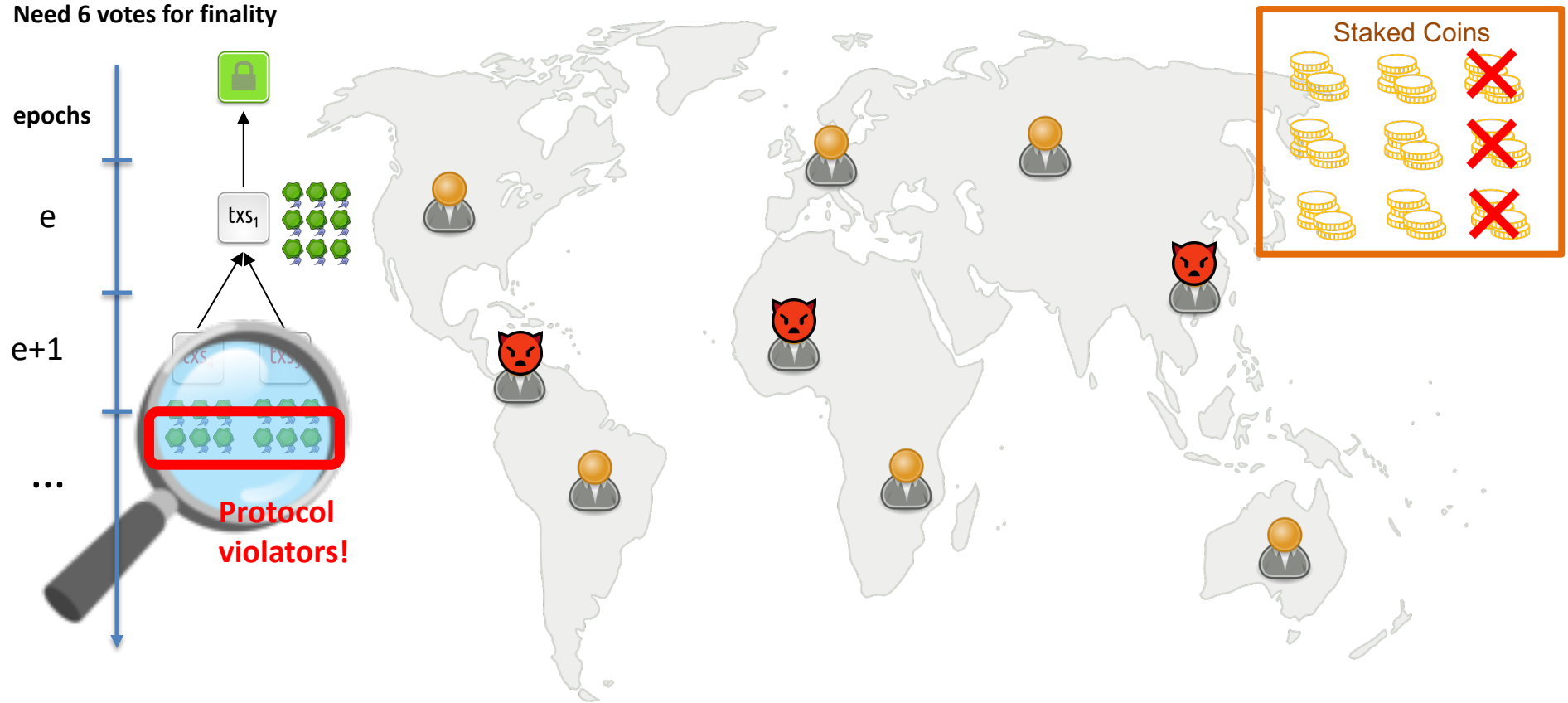
The more coins staked by a node…

- Higher the probability that the node is elected as a leader (recall Streamlet).
- Larger the weight of that node's vote.

If the node is caught doing an adversarial action (like voting for two conflicting blocks), it can be punished by burning its locked coins (stake)! This is called *slashing*.

Thus, in a Proof-of-Stake protocol, nodes can be held *accountable* for their actions (unlike in Bitcoin, where nodes do not lock up coins).

A few words on Proof-of-Stake



Accountable Safety

In a protocol with resilience of *n/3:*

- The protocol is secure (safe & live) if there are less than *n/3* adversarial nodes.
- **Example:** Streamlet under partial synchrony has resilience of *n/3.*

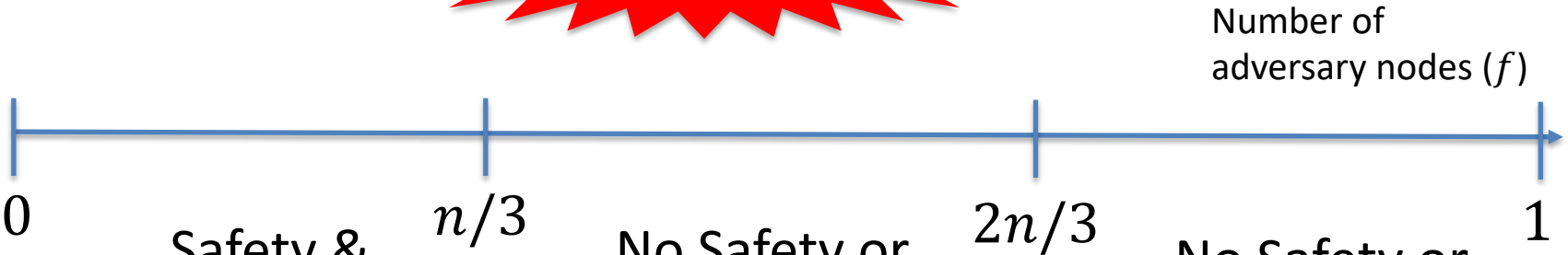
In a protocol with *accountable safety resilience* of *n/3:*

- The protocol is secure if there are less than *n/3* adversarial nodes.
- If there is ever a safety violation, all observers of the protocol can provably identify (i.e., catch) *n/3* adversarial node as protocol violators.
- No honest node is ever identified (no false accusation).
- **Examples:** PBFT, Tendermint, HotStuff, VABA…

Casper the Friendly Finality Gadget. (2017)
BFT Protocol Forensics (2021)

Accountable Safety

Accountable safety is a stronger notion than just security.



Resilience of n/3	Safety & Liveness ☺	No Safety or Liveness ☹	No Safety or Liveness ☹
Accountable safety resilience of n/3	Safety & Liveness ☺	• No liveness ☹ • If safety is violated, catch and punish adversarial nodes ☺	

Another Property of PoS: Finality

- Most accountably safe protocol examples we have seen satisfy safety and liveness under partial synchrony.
  - This means these protocols preserve safety during periods of asynchrony (before GST).

- We say that a protocol provides *finality* if it preserves safety during periods of asynchrony.
  - **Example:** Streamlet provides *finality*.

- Interestingly, in *most* protocol providing *finality*, transactions can be *finalized* much faster than they can be *confirmed* in Bitcoin.
  - No need to wait for k=6 blocks (1 hour)!

Holy Grail of Internet Scale Consensus

- We want Sybil resistance: Proof-of-Work or Proof-of-Stake…
- We want dynamic availability so that…
  - Transactions continue to be confirmed and processed even when there is low participation, e.g., due to a world-wide catastrophe.
- We want finality and **accountable safety** so that…
  - Finality: There cannot be safety violations (double-spends) during asynchrony.
  - **Accountable safety:** Nodes can be held accountable for their actions.
- Let's focus on having dynamic availability and finality for now…

Holy Grail of Internet Scale Consensus

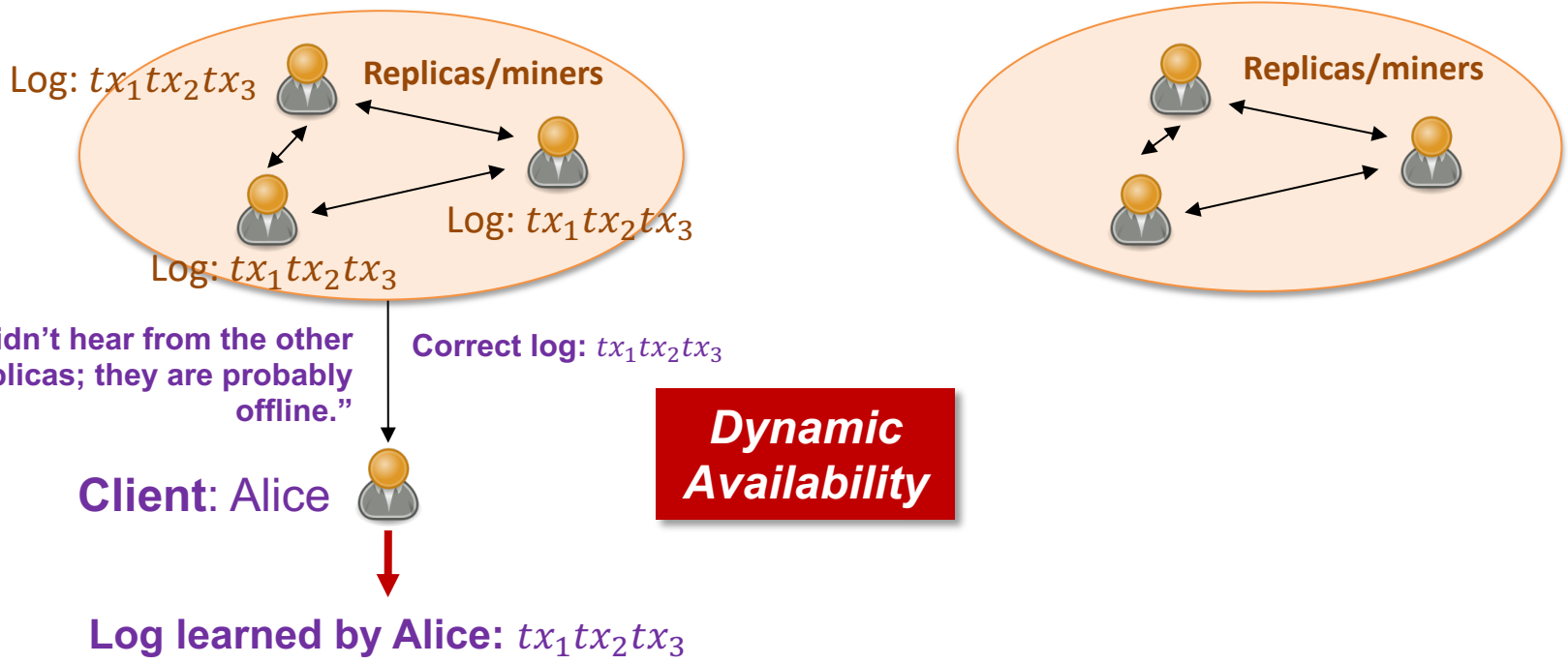
Is there a SMR protocol that provides both dynamic availability and finality?

No!

Blockchain CAP Theorem

Blockchain CAP Theorem

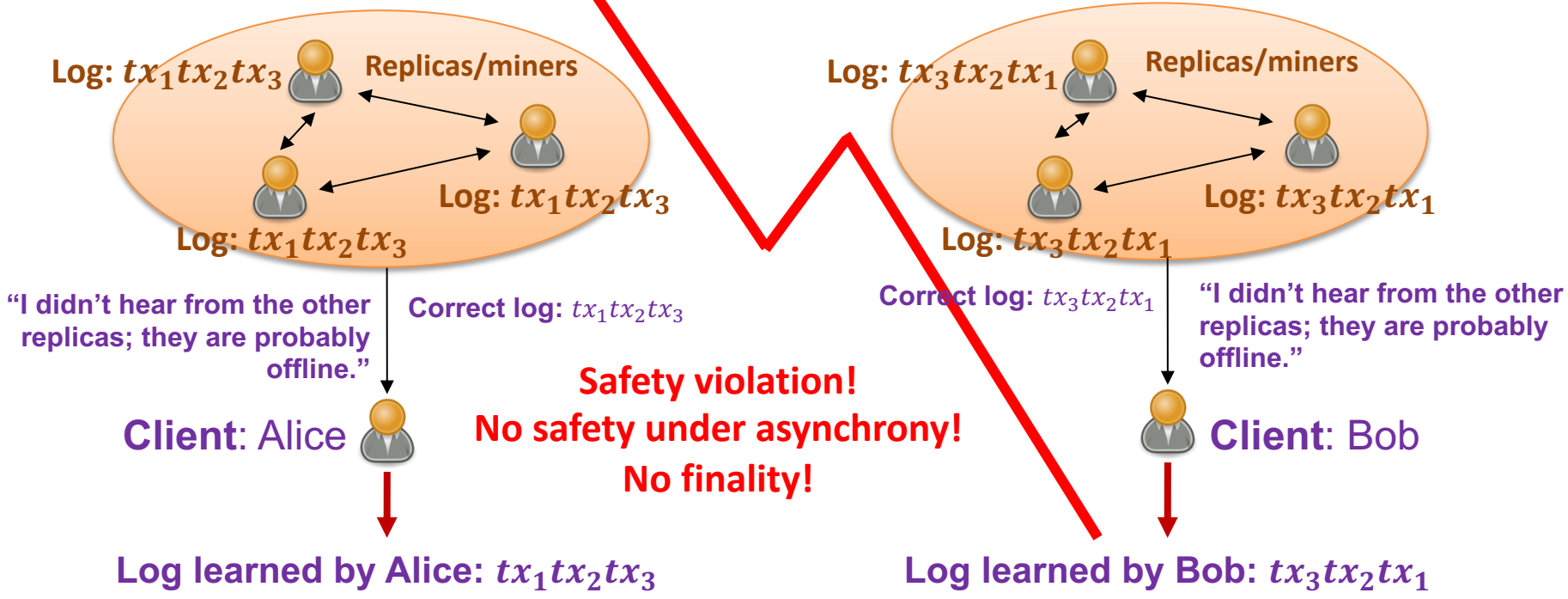
For contradiction, suppose our SMR protocol has both dynamic availability and finality.



Resource Pools and the CAP Theorem (2020)

Blockchain CAP Theorem

For contradiction, suppose our SMR protocol has both dynamic availability and finality.



Resolution: Nested Chains

Single chain: tx_1, tx_2, tx_3, …

Impossible!

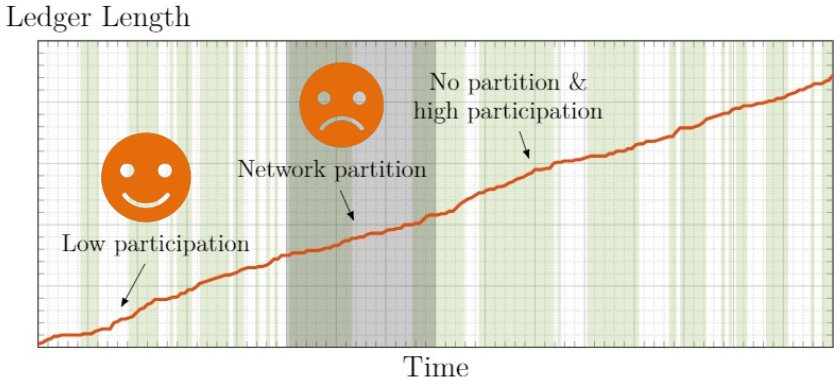
- Finality: Safe under asynchrony
- Dynamic availability: Live under dynamic participation

Finalized chain	Available chain
• Prefix of the available chain. • Safe under asynchrony. • Live once the network becomes synchronous and if enough nodes are online.	• Safe and live under synchrony and dynamic participation.

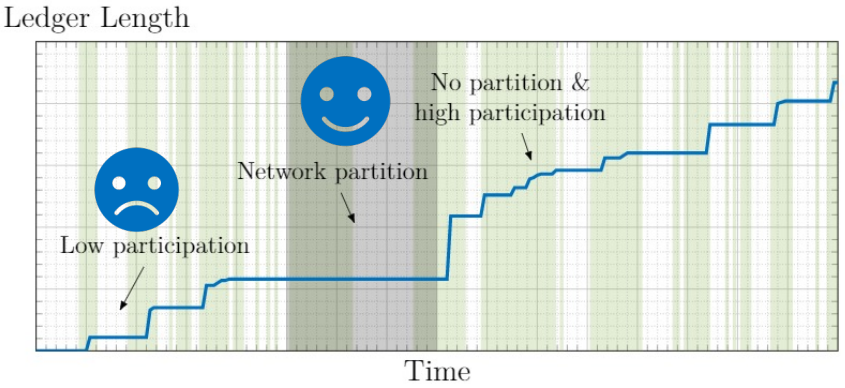
Client chooses better guarantee

Ebb-and-Flow Protocols: A Resolution of the Availability-Finality Dilemma (2020)

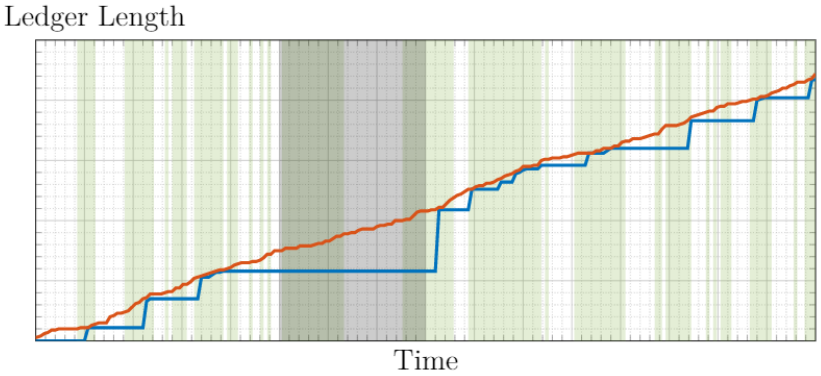
Resolution: Nested Chains



Available chain



Finalized chain



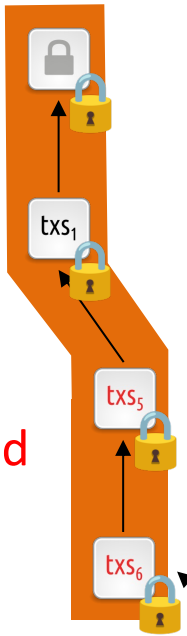
How to obtain the nested ledgers?

- The available chain is determined by a protocol, denoted by Π_{ava}, that satisfies dynamic availability (e.g., a protocol running Nakamoto Consensus).
- The finalized chain is determined by a *checkpointing* protocol, denoted by Π_{fin}, that satisfies security under partial synchrony.
  - **Examples:** Casper FFG, Grandpa, Afgjort, Accountability Gadgets…
- The chain *confirmed* by Π_{ava} is the available chain.
- Π_{fin} occasionally checkpoints blocks within the available chain.
- Prefix of the *last checkpoint* constitutes the finalized chain.

Casper the Friendly Finality Gadget. (2017)
Afgjort: A Partially Synchronous Finality Layer for Blockchains (2020)
GRANDPA: a Byzantine Finality Gadget (2020)
The Availability-Accountability Dilemma and its Resolution via Accountability Gadgets (2021)

How to obtain the nested chains?

Available and finalized chains



Always extend the last checkpoint!!

A

C

B

D

Dynamic Availability

Checkpointing Protocol

Propose blk “txs5”

C Votes “txs5”

B Votes “txs5”

D Votes “txs5”

Propose blk “txs6”

A Votes “txs6”

C Votes “txs6”

D Votes “txs6”

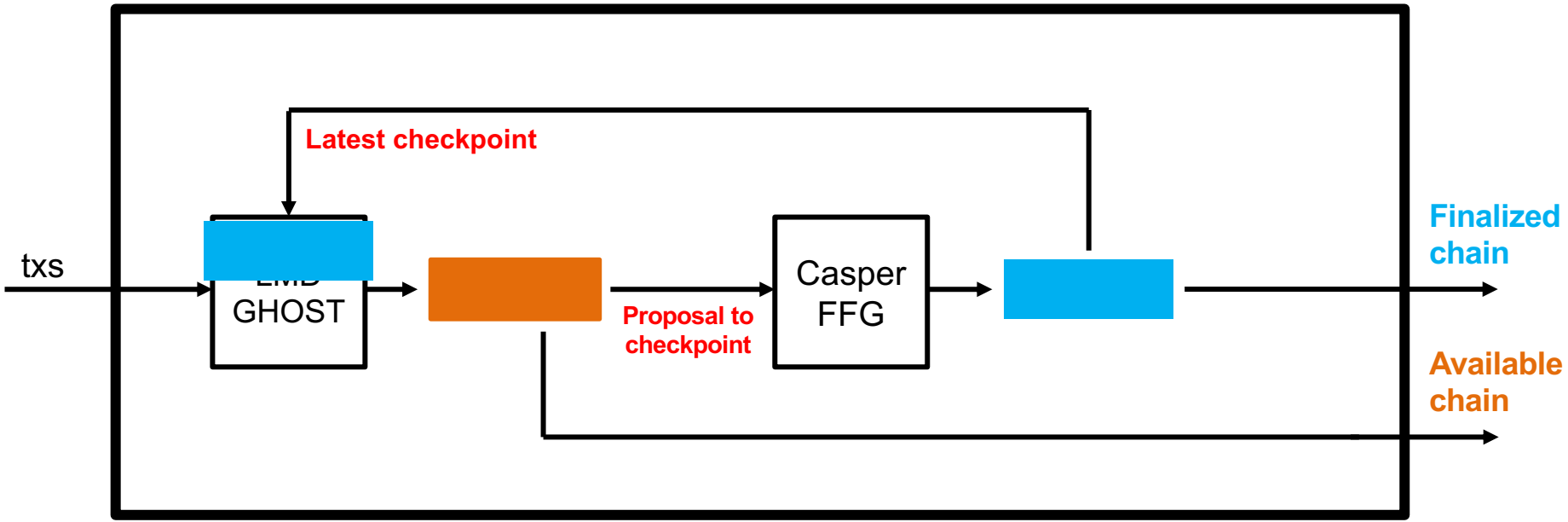
Finality: Thanks to votes, checkpoints are safe even under asynchrony.

PoS Ethereum

Consists of

- An available chain, which is determined by the protocol **LMD GHOST (Latest Message Driven - Greedy Heaviest Observed Subtree).**
  - The available chain provides dynamic availability.
- A finalized chain, which is determined by a *checkpointing* protocol called **Casper FFG (Casper the Friendly Finality Gadget).**
  - The finalized chain provides finality: safety under asynchrony.
- Besides finality, the finalized chain of PoS Ethereum provides **accountable safety:**
  - When there is a safety violation on the finalized chain, all observers of the protocol can provably identify *f* adversarial nodes as protocol violators, and no honest node.

PoS Ethereum



END OF LECTURE

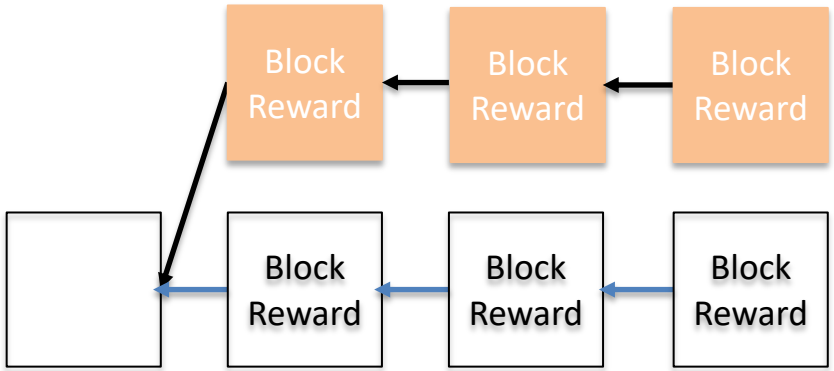
Next lecture: interesting scripts,
wallets, and how to manage crypto assets

Optional Slides

Slides going forward is optional material and investigate the Selfish Mining Attack.

Selfish Mining Attack (Optional)

Attacker keeps its blocks private until sufficiently many honest blocks are mined.
It then publishes the hidden blocks to ‘reorg’ the honest blocks.



Majority is not Enough: Bitcoin Mining is Vulnerable (2013)

Selfish Mining Attack (Optional)

Suppose you hold β fraction of the mining power.

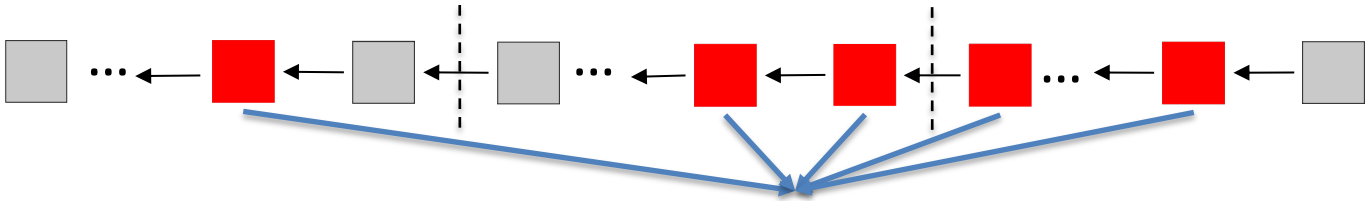
If you behave honestly, mining on the tip of the longest chain in your view and broadcasting your blocks as soon as they are mined…

You mine $\sim\beta$ fraction of the blocks.

You earn $\sim\beta$ fraction of the block rewards over Bitcoin's lifetime.

Note that the total amount of block rewards over Bitcoin's lifetime is fixed!

Selfish Mining Attack (Optional)



β fraction: adversary's blocks

Total fraction on the longest chain: 1

Remaining $1-\beta$ fraction: honest miners' blocks

Selfish Mining Attack (Optional)

If you do selfish mining…

You kick out $\sim\beta$ fraction of the mined blocks out of the longest chain.

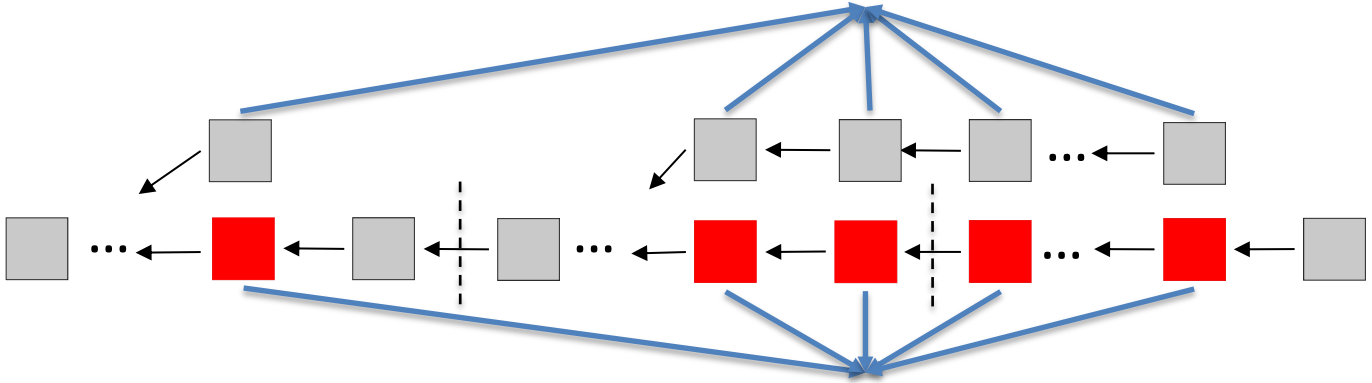
$\sim 1-\beta$ fraction of the mined blocks are in the longest chain.

You have mined $\sim \frac{\beta}{1-\beta}$ of the blocks in the longest chain.

You earn $\sim \frac{\beta}{1-\beta} > \beta$ fraction of the block rewards over Bitcoin's lifetime!

Selfish Mining Attack (Optional)

β fraction: honest miners' blocks displaced by the adversary's blocks



β fraction: adversary's blocks

Total fraction on the longest chain: $1-\beta$

Remaining $1-2\beta$ fraction: honest miners' blocks that were not displaced by the adversary's blocks

Selfish Mining Attack (Optional)

Chain quality (fraction of honest blocks in the longest chain) of Bitcoin $\leq \frac{1-2\beta}{1-\beta}$

Is it possible to make Bitcoin incentive compatible and increase chain quality to β?

Yes!

Examples: Fruitchains (ε-Nash equilibrium), Colordag (ε-sure Nash equilibrium)

Fruitchains: A Fair Blockchain (2017)
Colordag: An Incentive-Compatible Blockchain (2022)