Fair rewards for more decentralised attesters

A DAG-based reward mechanism

Devcon SEA — 15/11/2024

Barnabé Monnot

Robust Incentives Group (RIG), Ethereum Foundation Research

Why we need decentralised attesters

Validators as attesters come to consensus on the chain.
Thousands of votes are cast every round.

We **need** these votes to remain **truthful**.
We **need** to **secure** the attesters' voice.

Our current mechanisms are too brittle.
We present here an alternative.

Breaking the balance of power

Joint work with Roozbeh Saranche, Ertem Nusret Tas, Caspar Schwarz-Schilling, Bart Preneel

A simple commitment attack

Commitment attack

Deploying a smart contract “**warping**” the incentives of other validators to do the correct thing.

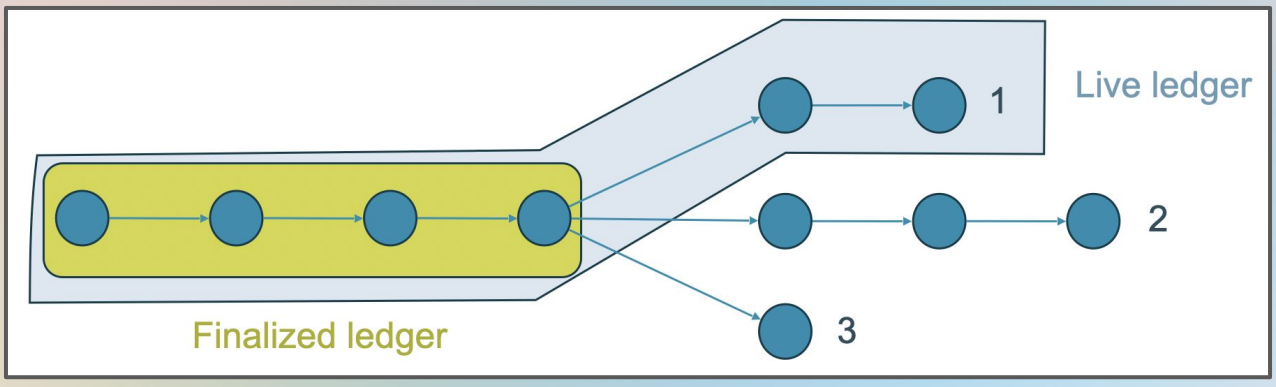
- Attacker commits to course of action.
- Victims must adjust behaviour/actions in reply.
- Coordinates victims towards attacker’s preferred outcome.

Interesting questions regarding the credibility of the commitment… but we don’t lack ways of being more credible :)

Gasper 101

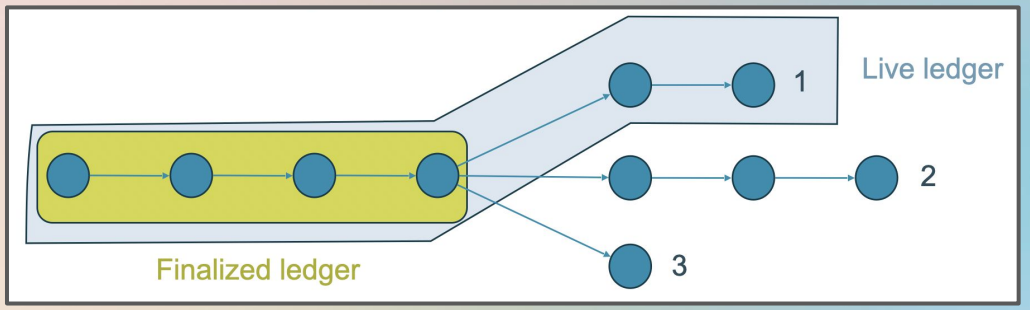
Ethereum's "Gasper" consensus mechanism has two components:

- FFG => **Finality** service
- LMD-GHOST => **Availability** service



Validator duties

To build the chain, a selected proposer makes a block.
The block contains user transactions and attester votes.
Attester votes have **FFG data** (what to finalise?)
and **LMD-GHOST data** (where is the head?).

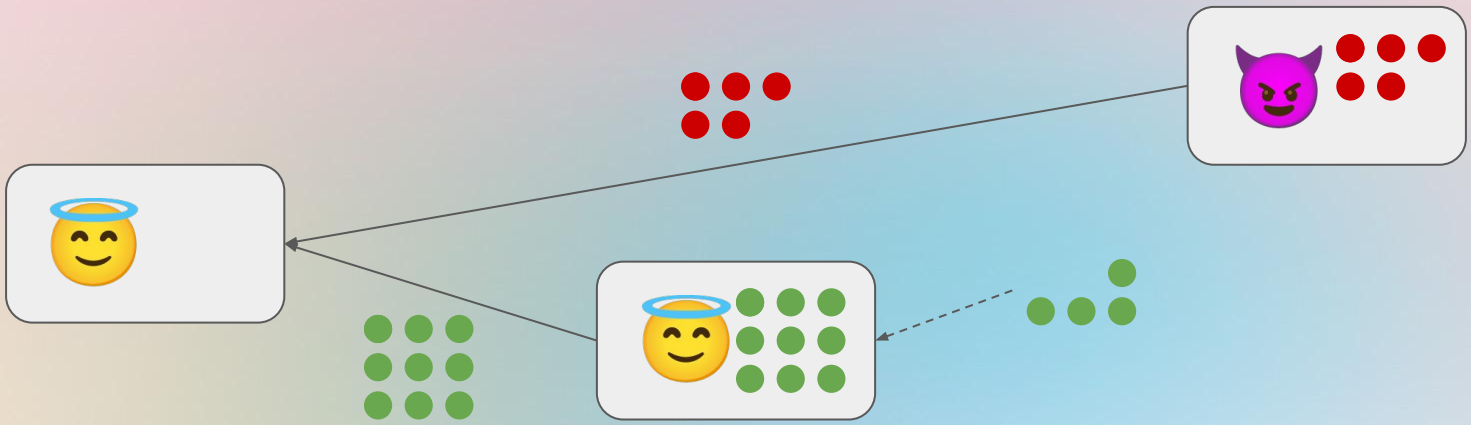


Good case

- Proposer 1 makes a block
- Attesters cast vote on the block
- Proposer 2 makes a block, includes attester vote

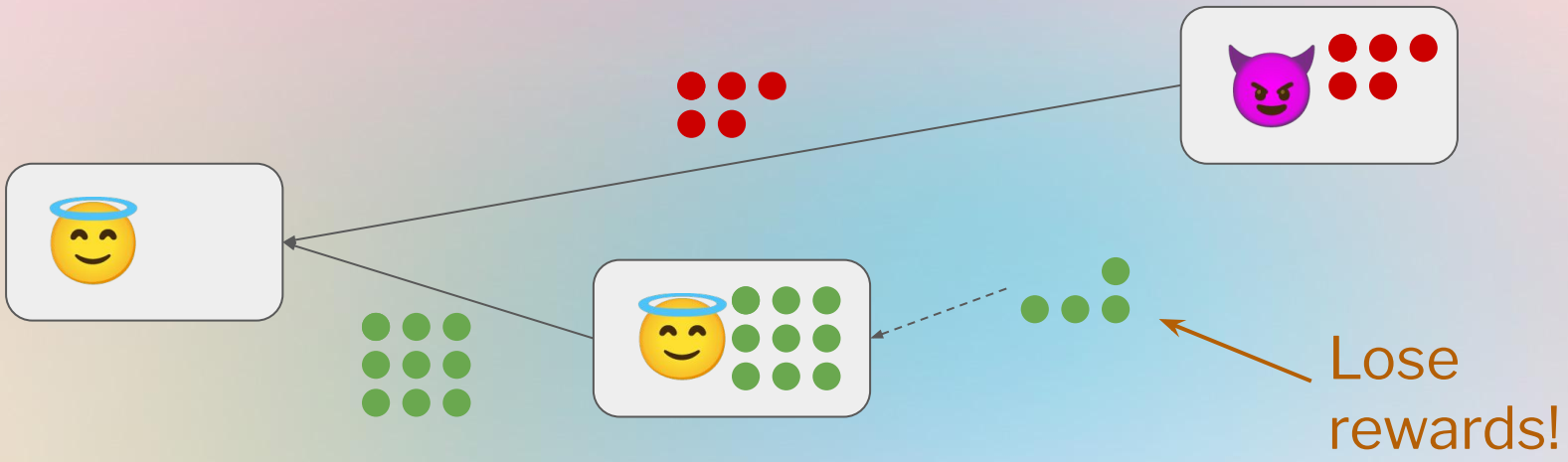
Proposer threat

The proposer threatens to ignore an attester vote if the attester doesn’t vote for the proposer’s chosen block.



Proposer threat

There is a Nash Equilibrium where attacker succeeds!



Extensions

In the paper we also discuss:

- Attacks over multiple blocks
  - With fixed attester sets (same players in all rounds)
  - With variable attester sets
- Attacks inspired by selfish mining attacks

Decentralising the proposer and fixing the rewards

Rewards in Ethereum Proof-of-Stake

Attesters are rewarded today when their head vote is:

- **Timely:** Included by the next proposer.
- **Correct:** Vote with the majority of attesters.

The timeliness constraint gives a lot of power to the next proposer!

Ensure timeliness without timely inclusion

The timeliness constraint gives a lot of power to the next proposer!

Strawman: Allow *any* proposer to include these votes.

… but how do we know that the votes were timely then?
We need the head votes ASAP!

Solution: Have attesters vote on the timeliness of other attesters!

~ Decentralise the role of the next proposer.

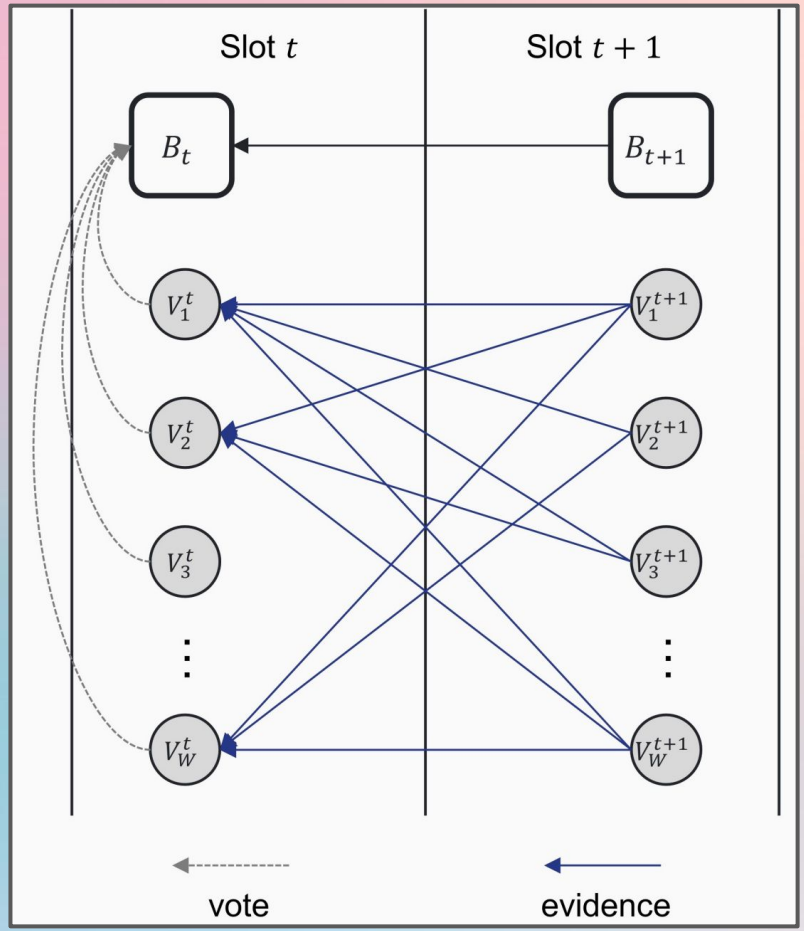
DAG-based votes

Attesters of slot *t*+1 vote on the votes of slot *t* attesters.

> “I have seen these votes arrive on time.”

These votes can be included **anytime**.

> Today, if block is missed, attesters lose reward :(



Performance

The DAG votes are a new object, creates overhead. But:

- In the good case (proposer is honest), then it’s not required, all votes are simply included.
- If proposer is missing or doesn’t include everything, then the DAG must be created.
- Worst-case is still reasonable.

~ think of it as insurance!

More on this in the paper.

Thank you!

https://rig.ethereum.org

see also “Breaking the Balance of Power: Commitment Attacks on Ethereum's Reward Mechanism”