Lecture 7: Bitcoin Liveness

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

Professor Pramod Viswanath
Princeton University

This lecture:
Liveness of the Bitcoin system
Chain Quality and Fairness

Recap

Last lecture: safety of the longest chain protocol, which means that once a block is confirmed (e.g., k-deep), then the probability of deconfirmation is very small (k large).

Safety is an important security property

But what if no block gets into the ledger, or blocks get in but some honest transactions don’t?

Liveness is an important security property: focus of this lecture

Observations

- The longest chain protocol cannot deadlock
    - Mining operation is very democratic and even a single honest miner with tiny hash power will eventually succeed in mining

- But that doesn’t guarantee liveness. There are two adversarial events.

- Notice the ledger is made up of blocks on the longest chain. It could happen that all blocks on the longest chain are adversarial

- A block mined by an adversary could be simply empty (censoring all transactions) or censor specific transactions

- Both these are fatal liveness attacks.

Chain growth

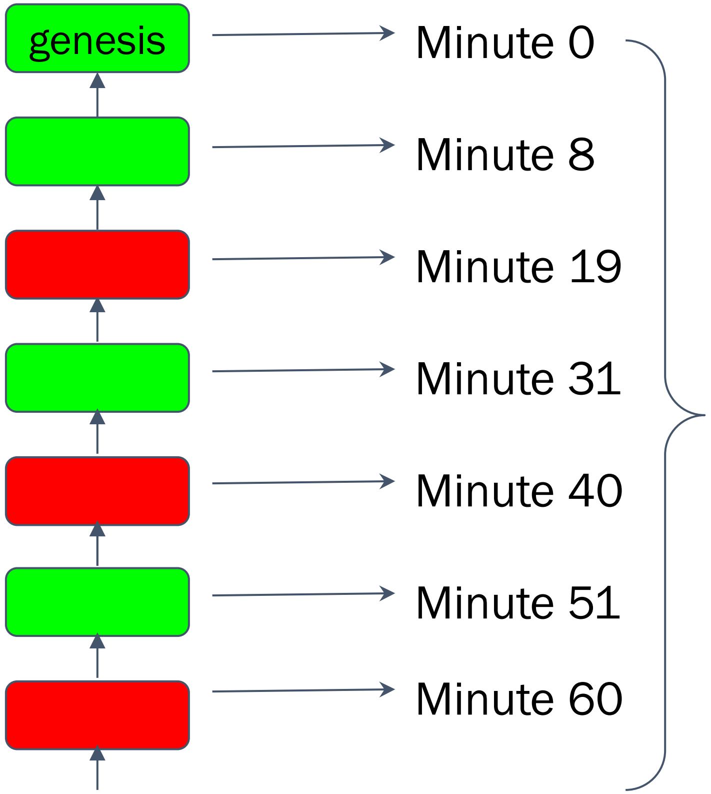
Chain growth (CG): rate of growth of the longest chain

Observation 1: $CG > 0$

- Adversary stays silent $CG = \frac{(1-\beta)\lambda}{1+(1-\beta)\lambda\Delta}$
- Adversary acts honest $CG = \frac{(1-\beta)\lambda}{1+(1-\beta)\lambda\Delta} + \beta\lambda$

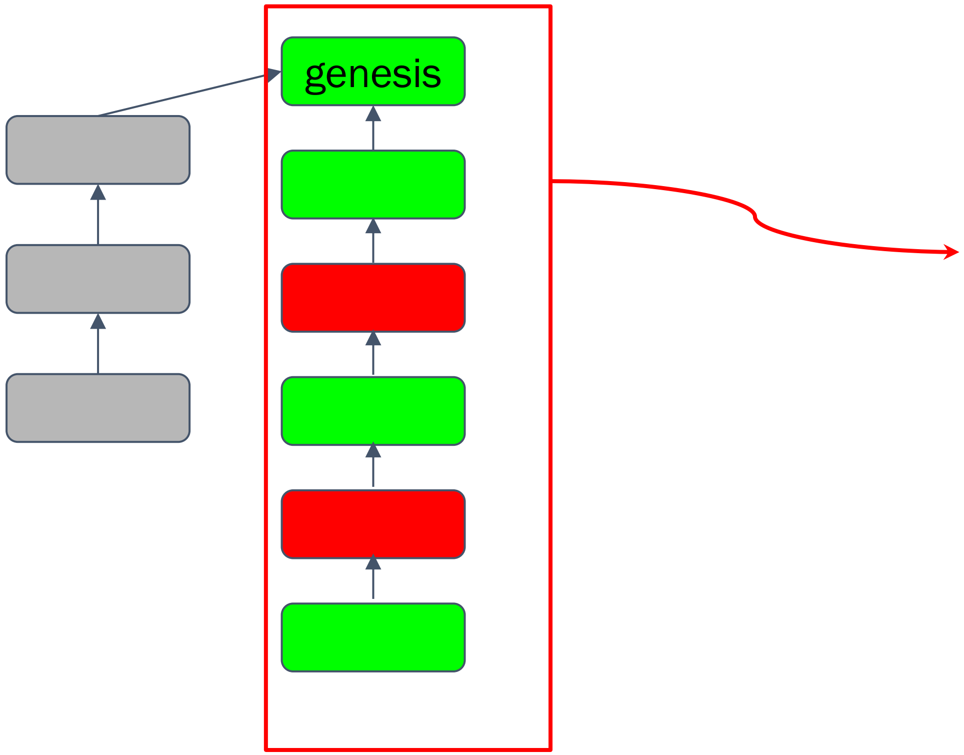
Claim: $CG > \frac{(1-\beta)\lambda}{1+(1-\beta)\lambda\varDelta}$

Chain growth



Chain growth is
1 block/10 minutes

Chain quality



4 honest blocks

2 adversarial blocks

In the **longest chain**

=> chain quality is 4/6 = 0.667

Chain quality (CQ): # of honest blocks in the longest chain/# of all blocks in the longest chain

Chain quality

Observation: we need $CQ > 0$ for liveness

$$CQ \geq \frac{CG * T - \beta\lambda T}{CG * T} = \frac{CG - \beta\lambda}{CG}$$

$$CQ > 0 \iff \frac{(1-\beta)\lambda}{1+(1-\beta)\lambda\Delta} > \beta\lambda$$

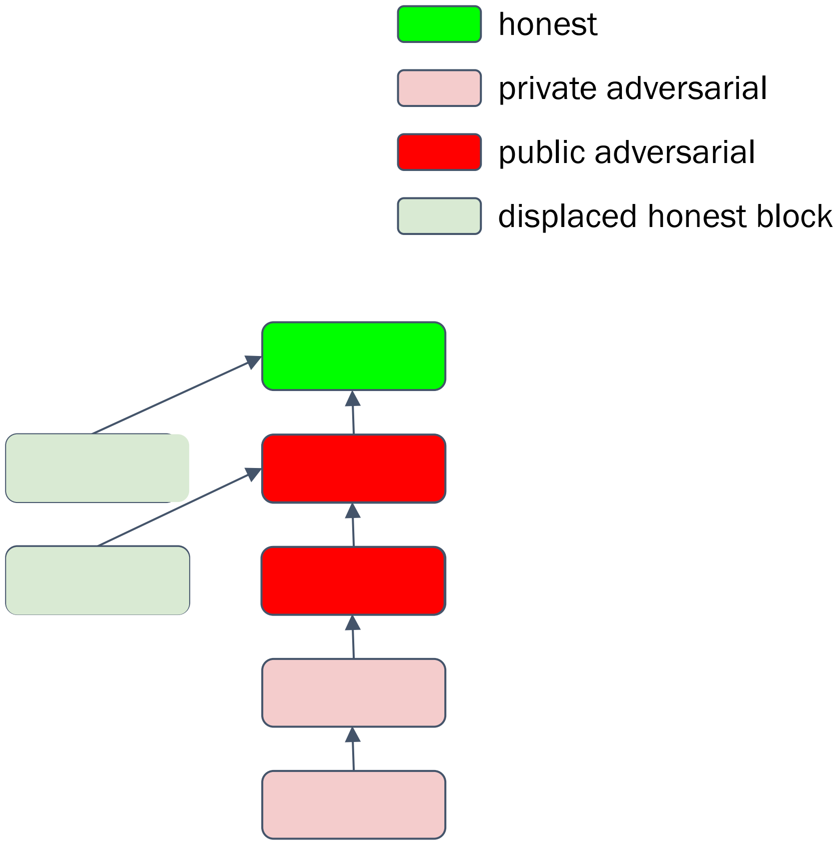
This is exactly the same condition for safety

Turns out the condition above is also necessary for liveness → Selfish mining

Selfish mining attack

1. The adversary always mines on the block at the tip of the longest chain, whether the chain is private or public. Upon successful mining, the adversary maintains the block in private to release it at an appropriate time.
2. When an honest miner publishes a block the adversary will release a previously mined block at the same level (if it has one).

* Adversary can break ties in its favor, so honest miners will mine on the adversarial block.



Chain quality and liveness

$$1 - \beta \geq CQ \geq \frac{\frac{(1-\beta)\lambda}{1+(1-\beta)\lambda\Delta} - \beta\lambda}{\frac{(1-\beta)\lambda}{1+(1-\beta)\lambda\Delta}}$$

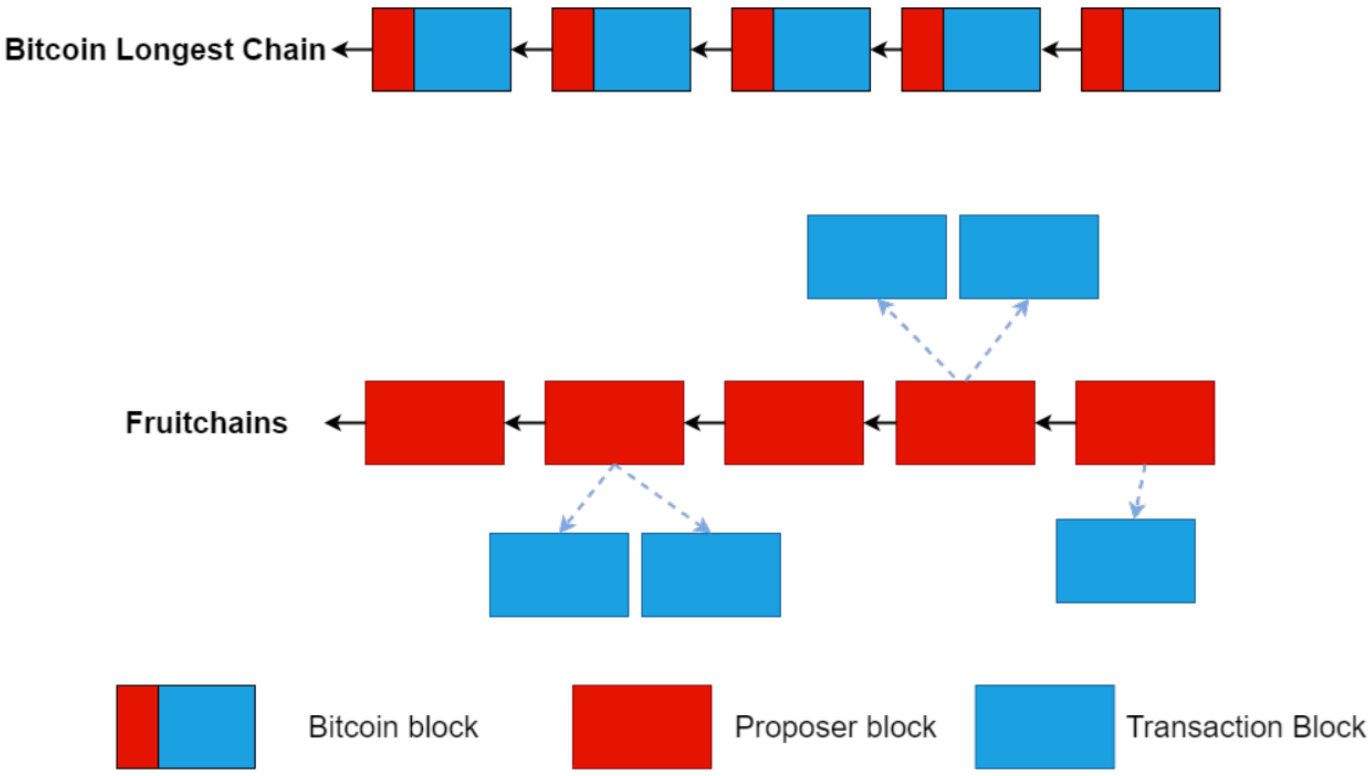
Chain quality quantifies liveness and **fairness**

The most fair reward distribution occurs when number of honest blocks on the longest chain is proportional to honest hash power $(1-\beta)$, i.e., $CQ = 1 - \beta$,

This is not happening in the longest chain protocol (due to selfish mining).

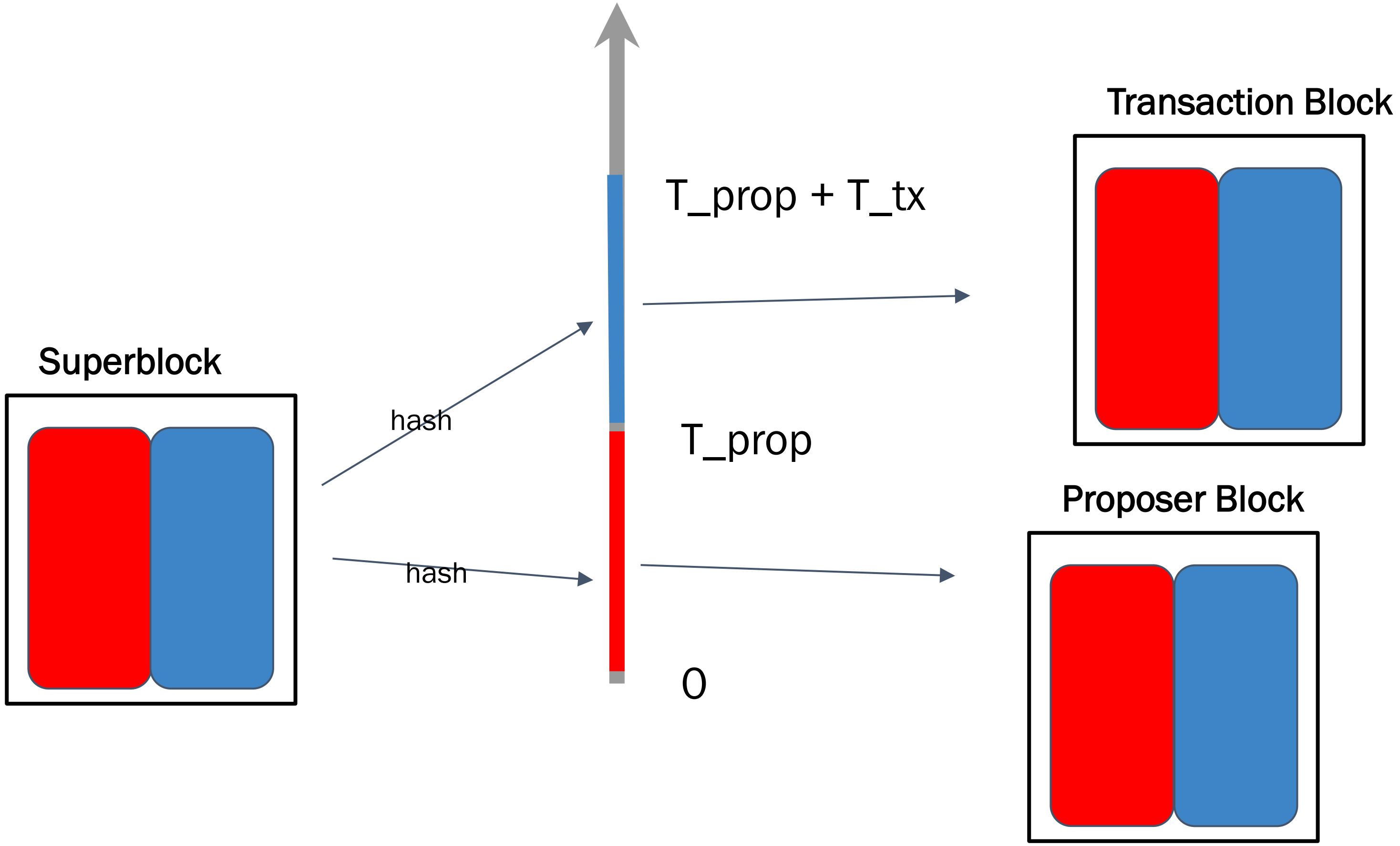
Fruitchains

Main idea: separate transactions (& their rewards) from blocks in the longest chain

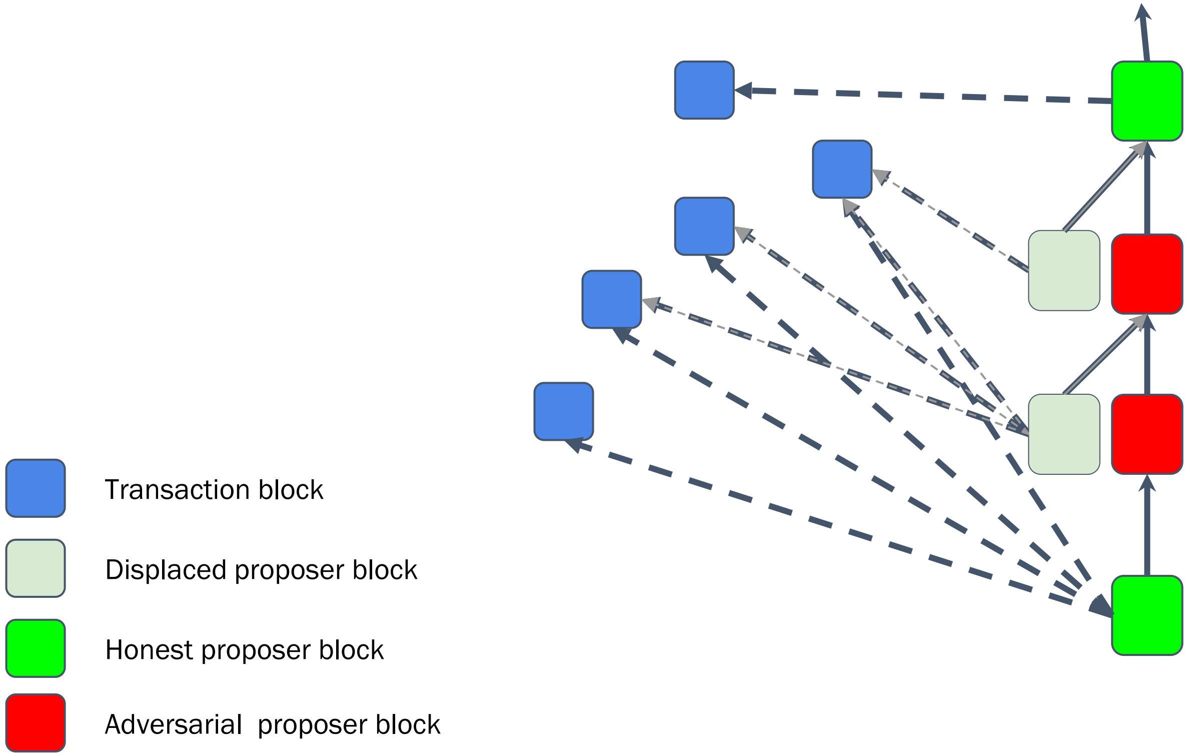


Cryptographic Sortition

How to do PoW for both types of blocks simultaneously?



Optimal chain quality



🟦 Transaction block

⬜ Displaced proposer block

🟩 Honest proposer block

🟥 Adversarial proposer block

Fraction of honest transaction blocks **in the longest chain** → Fraction of honest transaction blocks **overall** → $\mathrm{CQ} \geq \frac{(1-\beta)\lambda_{\mathrm{tx}}}{(1-\beta)\lambda_{\mathrm{tx}} + \beta\lambda_{\mathrm{tx}}} = \boxed{1-\beta.}$

Short time scale optimal CQ

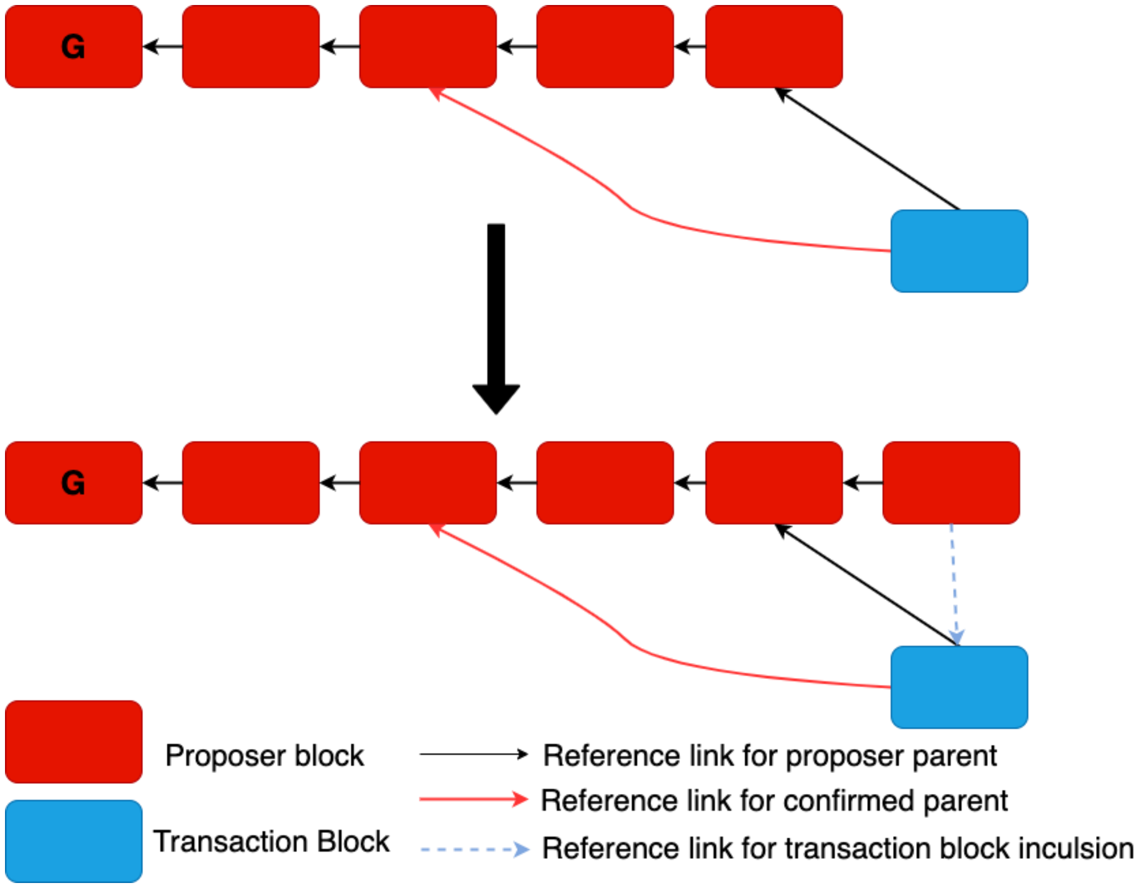
Block withholding attack:

- An attacker keep successfully mined transaction blocks private
- Then it suddenly release a large number of them at the same time, thereby creating a very high fraction of adversarial transaction blocks in some small segment of the proposer chain

Resolution: by requiring that a transaction block should “hang” from a confirmed proposer block which is not too far from the proposer block which includes it.

Short time scale optimal CQ

- During mining, each transaction block refers to a recently stabilized/confirmed proposer block (called confirmed parent)
- Recency condition: a transaction block B is recent with respect to a proposer chain C if the confirmed parent of B is a block that is at most R deep in C, where R is a recency parameter.
- Proposer blocks only include recent transaction blocks


G
G
Proposer block
Transaction Block
Reference link for proposer parent
Reference link for confirmed parent
Reference link for transaction block inculsion