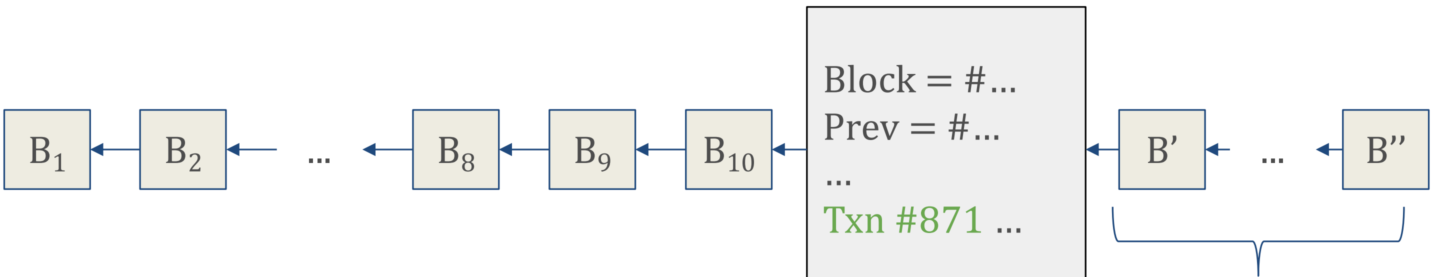
RECAP

View of someone who wants to make a transaction



Txn=#871	George	Anna	1 ₿	....

Wait a few blocks until you can say that the transaction is confirmed

WHY?

Want some assurance that this block will be on the longest chain in the long run!

PROOF OF WORK: RECAP

View of a miner

$B_1 \leftarrow B_2 \leftarrow \dots \leftarrow B_8$

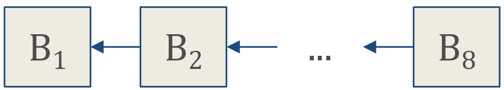
SHA256 (

Block = #8ae1…
Prev = B_8
….
Txn #123 = …
….
nonce

) = 0x0b39d9ca51f07fef3429ae15…

PROOF OF WORK: RECAP

View of a miner



SHA256 (

Block = #8ae1...
Prev = B_8
....
Txn #123 = ...
....
nonce'

) = 0x000000ef34244s1jd99a533g...

PROOF OF WORK: RECAP

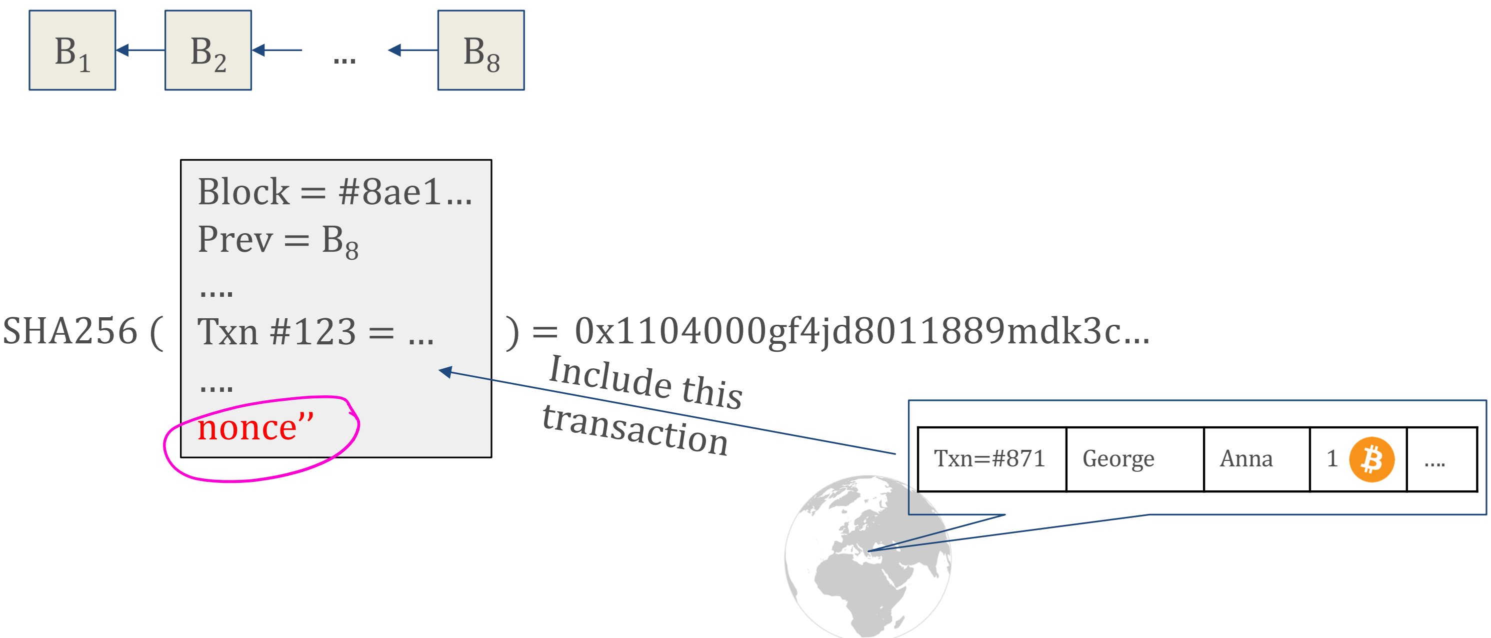
View of a miner

$B_1 \leftarrow B_2 \leftarrow \ldots \leftarrow B_8$

SHA256 (
Block = #8ae1…
Prev = B_8
….
Txn #123 = …
….
nonce’’
) = 0x1104000gf4jd8011889mdk3c…

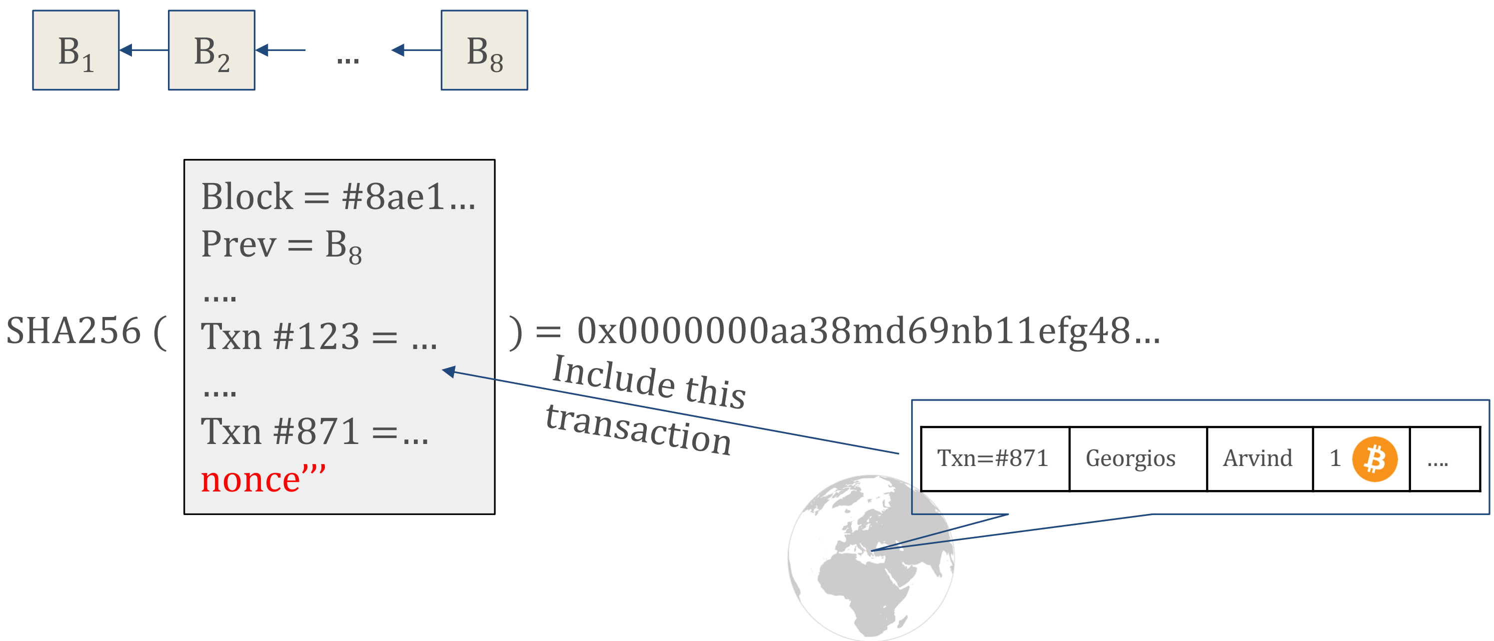
PROOF OF WORK: RECAP

View of a miner



PROOF OF WORK: RECAP

View of a miner



PROOF OF WORK: RECAP

View of a miner

$$B_1 \leftarrow B_2 \leftarrow \dots \leftarrow B_8 \leftarrow B_9$$

You lost the race

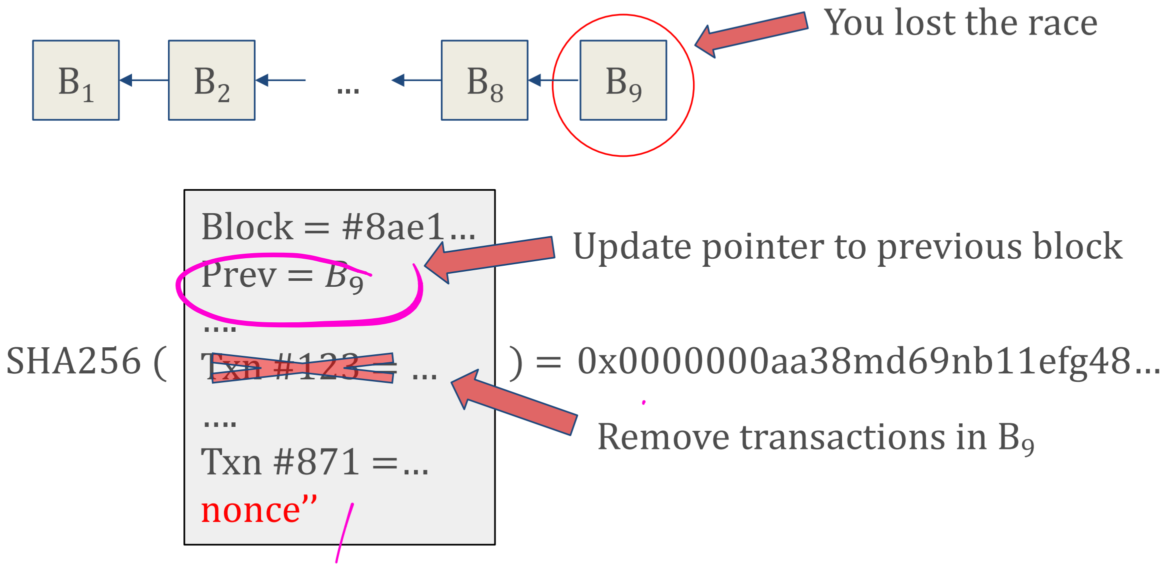
SHA256 (

Block = #8ae1…
Prev = B_8
….
Txn #123 = …
….
Txn #871 =…
nonce”

) = 0x0000000aa38md69nb11efg48…

PROOF OF WORK: RECAP

View of a miner



PROOF OF WORK: RECAP

View of a miner

$B_1 \leftarrow B_2 \leftarrow \dots \leftarrow B_8 \leftarrow B_9$

You lost the race

SHA256 (

Block = #8ae1…
Prev = B_9
….
~~Txn #123 =~~ …
….
Txn #871 =…
nonce

) = 0x0000000aa38md69nb11efg48…

Update pointer to previous block

Remove transactions in B_9

Keep trying!

RECAP OF BITCOIN

- **Transactions:** At any time, any buyer b can generate a transaction to pay d BTC to seller s.
- **Block:** A block consists of
  - A set of transactions
  - A cryptographic hash of the previous block (pointer to previous block
  - An ID of the miner for this block
  - A nonce.
- A set of properly signed transactions is **valid** if no account ever overspent its limit.
- A block is valid if
  - It points to a valid block.
  - All transactions on the chain to B are valid.
  - SHA256(nonce|| info in block) has k leading zeros.

RECAP OF BITCOIN II

- **Mining:** the process of extending the blockchain from some block B.
- Longest Chain Protocol (for miners):
  - Choose B to be the block furthest from the root, tie-breaking in favor of the first block you heard about.
  - Include all valid transactions you've heard about.
  - As soon as valid block created, announce it to the network.
- Miners are paid for creating valid blocks with freshly minted Bitcoins and with transaction fees.
- Difficulty of the puzzle is adjusted every 2016 blocks with the objective of making it so that a block takes 10 minutes to make in expectation.

KEY IDEA

- Trust the ledger that has the most "computational work" put into it.

- Ensure that fraudulent transactions/conflicting ledgers would require an infeasible amount of computation to create.

BITCOIN

- Is a mechanism.

- Question for us: are there beneficial deviations that can help a miner earn more than his fair share of rewards?

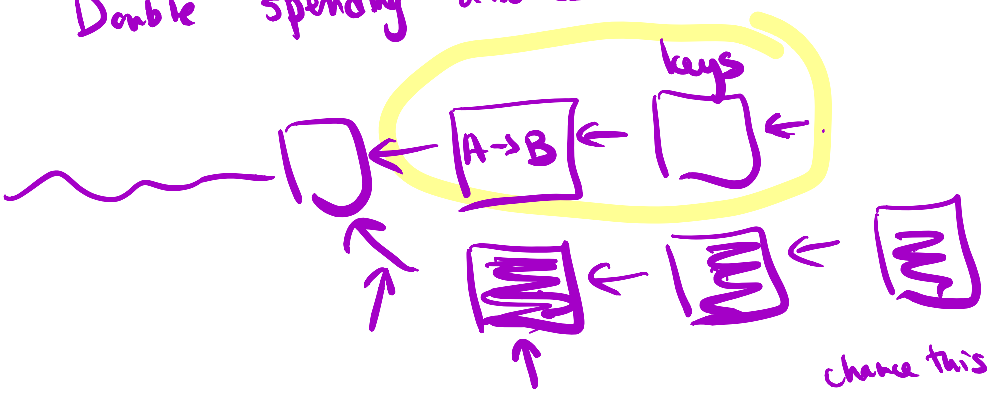
difficulty adjustment:

longest chain gets extended by block on avg every 10 mins.

What can a miner do?

- choose any block they know about to mine on.
- deliberately fork, "dishonest tie-breaking"
- hide a block once they find it.
- include any valid transactions you want.

Double spending attacks



my mining power is α

chance this happens is α^3

$\alpha > 51\%$

Selfish Mining.

Mining game:

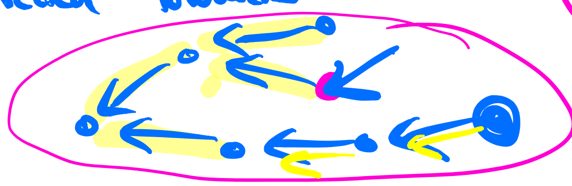
Set of miners

x_m fraction of mining power that miner m has

$\sum_m x_m = 1$

At all times, each miner is aware of tree directed towards root

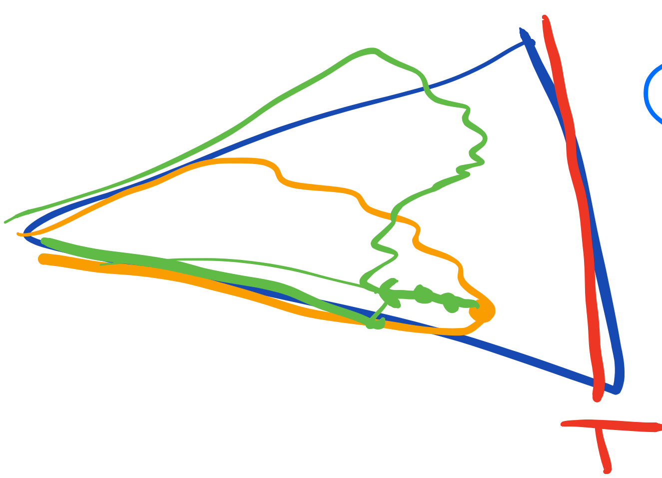
G_m [graph/tree miner m is aware of]



nodes represent blocks.

at each time t, a random miner is selected to mine a block. Miner m is selected w/ prob x_m

at each time step, each miner m can pick any node v in G_m & broadcast path from v to root.



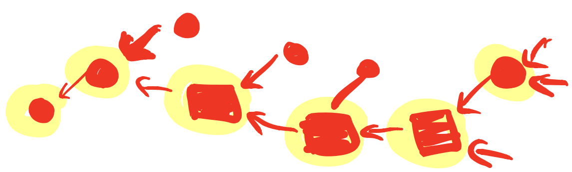
Fix miner m
$x_m = \alpha$

Payoff to a miner is $\lim_{T \to \infty} \frac{\text{\# of blocks they've mined on the longest public chain at time } T}{T}$

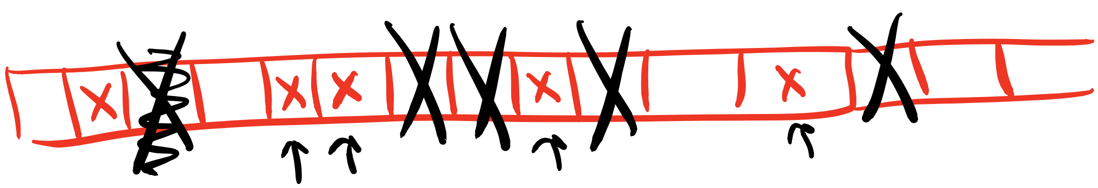
m

Assumption: if miner m is in a tie with any other block for longest chain, everyone else mines on m's chain.

miner m will mine on longest chain, breaking ties in favor of his own blocks, but will only broadcast a block he has found if $\exists$ another block at same distance from root



every block



reward fraction of his blocks on longest chain.

$T\alpha$ blocks in T time steps.

kills $T\alpha$ blocks on longest chain

$$\frac{T\alpha}{T(1-\alpha)} = \boxed{\frac{\alpha}{1-\alpha}} > \alpha$$

$\alpha = \frac{1}{3}$

$$\frac{\frac{1}{3}}{\frac{2}{3}} = \frac{1}{2}$$

G_m his view

G public view

$h_m =$ height(G_m)
length of longest chain

$h =$ height(G)

Assume that all ties lost by m
This improves rewards for m for
any $\alpha > .3$

Miner m strategy
– always work on longest chain in G_m
in choosing which to mine.
– Break ties in favor of his own blocks

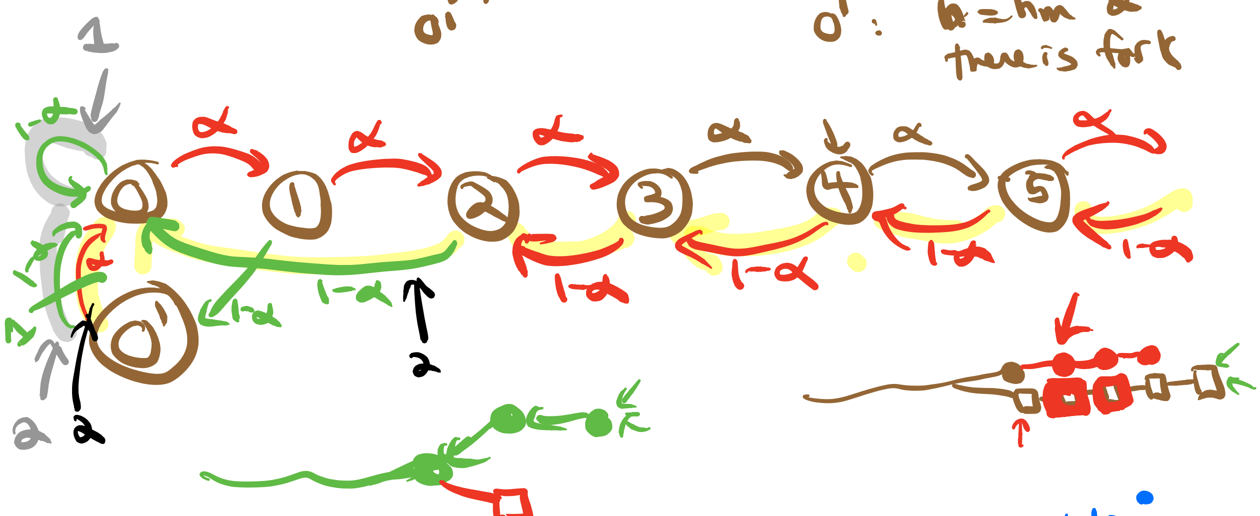
– Block announcing strategy
honest
If another miner finds a block at height h, then
– if $h_m \geq h+2$, announce h
– if $h_m = h+1$, announce h & $h+1$
– if $h_m = h$ announce h. Moreover, if m finds next block immediately announce it.

Markov Chain has set of states S
$\forall$ state u,
$q_{uv} =$ Prob next state is v | current state is u

$\sum_v q_{uv} = 1$

state $i = 0, 1, 2, \ldots$
$0'$

state i means $h_m = h + i$
$0'$: $h = h_m$ & there is fork



Let p_i be long run (stationary) prob of being in state i

$p_0 = p_0(1-\alpha) + p_{0'} + p_2(1-\alpha)$

$p_{0'} = p_1(1-\alpha)$

$p_1 = p_0\alpha$

$p_i = p_{i-1}\alpha + p_{i+1}(1-\alpha) \quad i \geq 2$

∞

stationary distn: long-run prob of being in each state

long run prob $\boxed{p_v \text{ of being in state } v}$

$\forall v \quad p_v = \sum_{u \in S} p_u q_{uv}$ $\quad (q_{uv})_{u,v \in S}$

$\sum_u p_u = 1$

$p_0' + \sum_{i=0} p_i = 1$

selfish reward on transition from x to y

Selfish payoff : $\sum_{x \in S} p_x \sum_{y \in S} q_{xy} S_{xy}$

honest reward on transition from x to y

Honest payoff $\sum_{x \in S} p_x \sum_{y \in S} q_{xy} H_{xy}$

we'll count a block when it is first announced & guaranteed to be on longest chain

$$\frac{\text{Selfish payoff}}{\text{Selfish payoff} + \text{Honest payoff}}$$

exp payoff of selfish miner w/ mng power α assuming loses all ties is

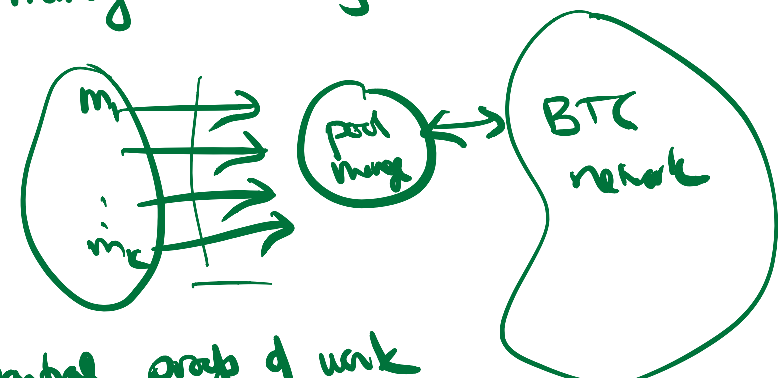
$$\frac{4\alpha^2(1-\alpha)^2 - \alpha^3}{1-\alpha(1+\alpha(2-\alpha))} \overset{?}{>} \alpha$$

true when $\alpha \geq 0.3$

$$\frac{\alpha}{1-\alpha} > \alpha$$

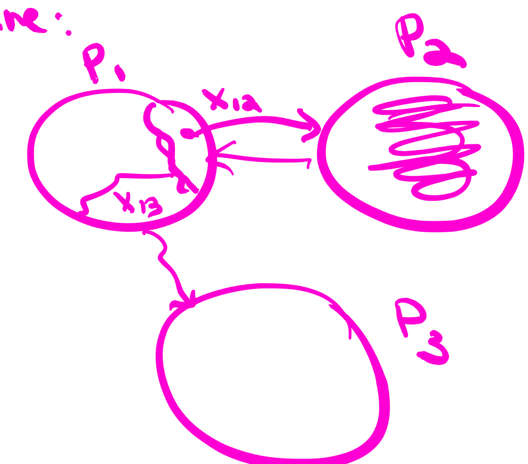
The Miner's Dilemma.

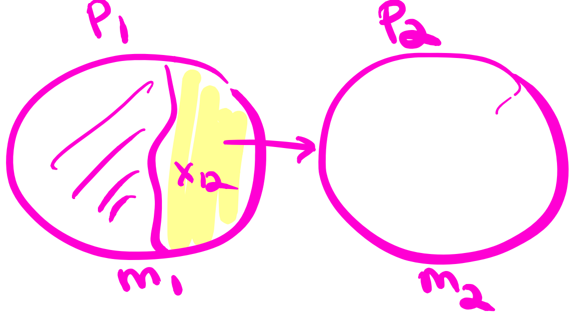
mining pools used to reduce the variance.
pool manager that joins Bitcoin as a single miner



use partial proof of work
to figure out how to split rewards among miners

The pool game:





$m = m_1 + m_2$

block rewards $R_1 : \dfrac{m_1 - x_{12}}{m - x_{12}}$

$R_2 : \dfrac{m_2}{m - x_{12}}$

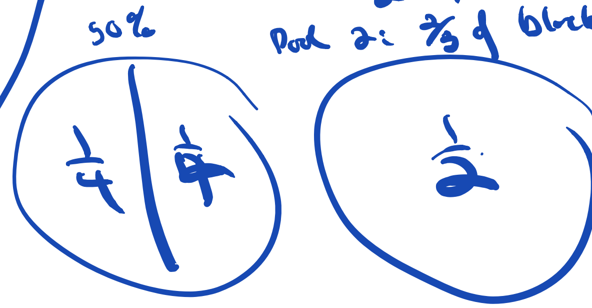
$r_1 = \dfrac{R_1 + x_{12} r_2}{m_1}$

$r_2 = \dfrac{R_2}{m_2 + x_{12}}$

Pool 1 attacks Pool 2 w/ $\frac{1}{2}$ of its total power

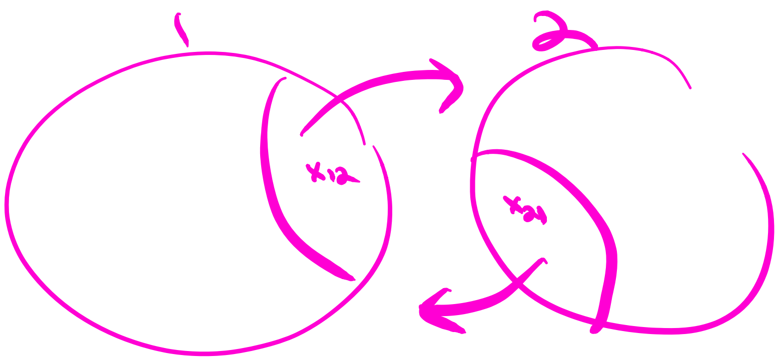
Pool 1: $\dfrac{\frac{1}{4}}{\frac{1}{2}+\frac{1}{4}} = \dfrac{1}{3}$

Pool 2: $\frac{2}{3}$ of block rewards



Pool 1 also gets $\dfrac{\frac{1}{4}}{\frac{1}{4}+\frac{1}{2}} = \dfrac{1}{3}$

Pool 1 makes $\frac{1}{3} + \frac{1}{3} \cdot \frac{2}{3} = \frac{5}{9}$



NE as long as both pools < 80%

	pool 2 doesn't	pool 2 does
pool 1 doesn't attack	(a, b)	$(a-2\varepsilon, b+2\varepsilon)$
pool 1 does	$(a+2\varepsilon, b-2\varepsilon)$	$(a-\varepsilon, b-\varepsilon)$