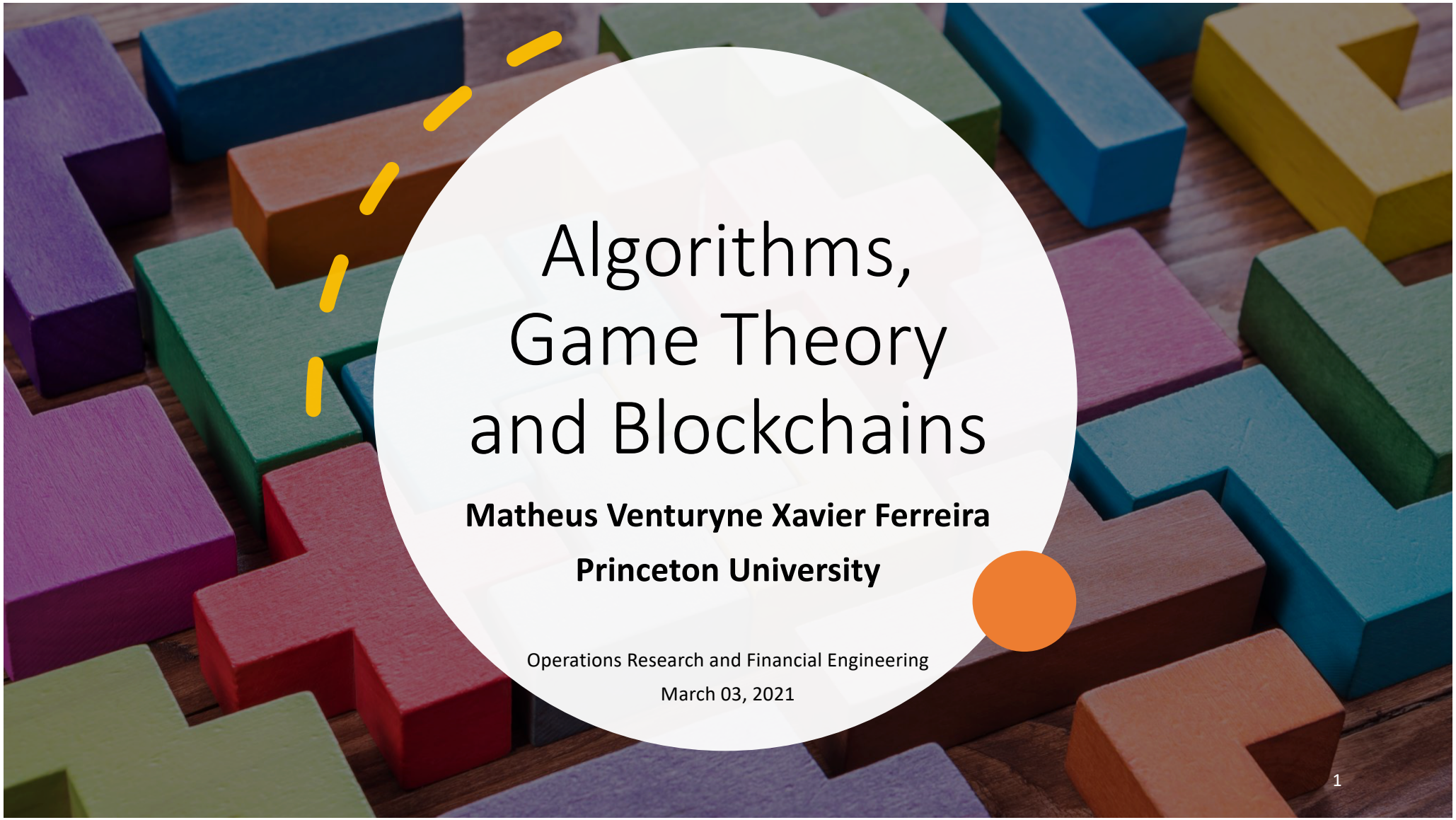


Algorithms, Game Theory and Blockchains

Matheus Venturyne Xavier Ferreira

Princeton University

Operations Research and Financial Engineering

March 03, 2021

Focus

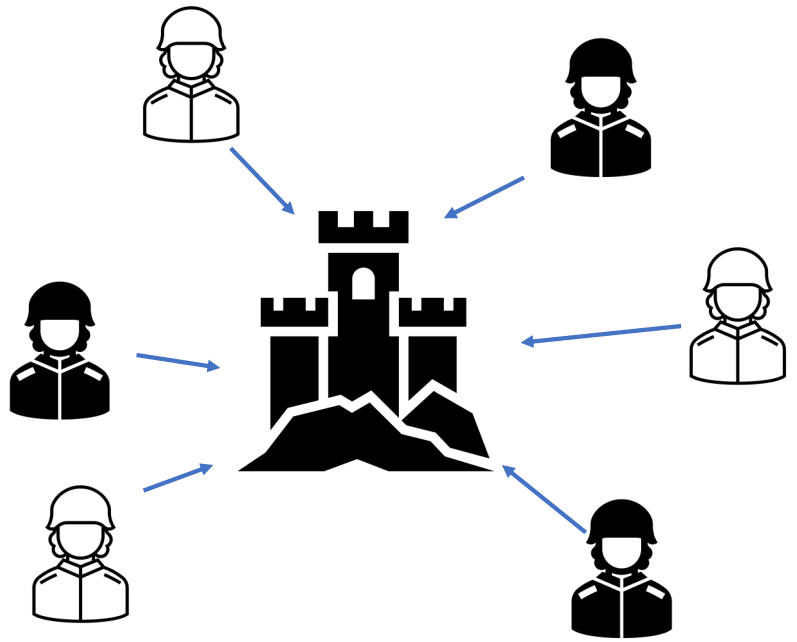
- Blockchains are mechanisms.
- Challenges of design blockchains that incentivize compliance.
- Analyzing existing mechanisms and propose new ones.
- **Not the focus:**
  - Use cases (why does cryptocurrency have value)?
  - Cryptographic security.
  - Network security.
  - “Malicious attacks’’.

Types of blockchains

- **Public and permissionless.**
  - Anyone can join, read, write.
  - Low scalability.
  - Example: Bitcoin, Ethereum, …
- Public and permissioned.
  - Anyone can join and read but need authorization to write.
  - Medium scalability.
  - Example: Ripple.
- Private.
  - Need permission to join, read and write.
  - High scalability.

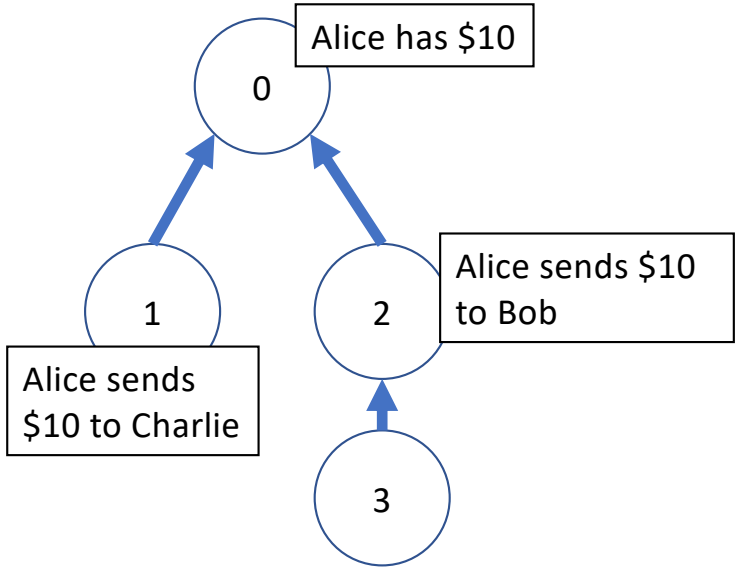
What problem does Bitcoin solve?

- Byzantine generals problem.
  - [LSP, 82] Requires 2/3 of generals to be loyal.
- Impossible on a public, permissionless network.
  - Bitcoin overcomes this impossibility by introducing monetary incentives and so far, has been successful.



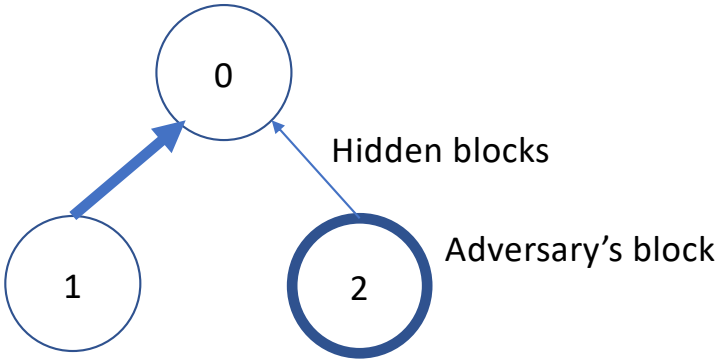
Blockchains are mechanisms

- Game with **users** and **miners**.
- **Users**
  - Submit transactions.
- **Miners**
  - Create blocks.
  - Each block stores at most m transactions.
  - Receives fresh coins as reward.
- Conflict resolution via the longest chain rule.
  - Intended behavior: **ALL** miners extend the longest path.



Assumptions

- What adversaries **cannot** do:
  - Change blocks, pointers, transaction, …
  - Steal someone else's block or coins.
  - **Takeaway**: you cannot break crypto.
- What adversaries **can** do:
  - Point to any block (**NOT** extend the longest path).
  - Hide a block (and broadcast it later).
  - Include any transactions in a block.
  - **Takeaway**: the algorithm designer cannot force miners to take any particular action.

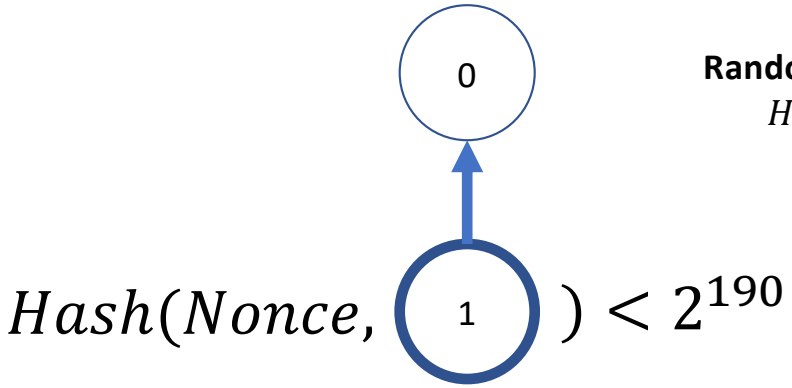


Outline

- **Mining games**.
  - Proof-of-work (PoW) [Eyal, Sirer 13][Sapirshtein, Sompolinsky, Zohar 16] [**F**, Weinberg 21].
  - Energy-efficient alternative: Proof-of-stake (PoS) [**F**, Weinberg 21].
- Transaction fee auctions.
  - Monopolistic pricing [Lavi, Sattath, Zoha 19][Yao 20].
  - Dynamic posted-pricing [**F**, Moroz, Parkes, Stern, 21].
- Mining pools.
  - Attacks on mining pools [Eyal 14].
- Difficult adjustment algorithm (DAA)
  - Bitcoin cash vs Bitcoin [Noda , Okumura, Hashimoto 20]
  - Chaotic pure equilibrium in PoW [Fiat, Karlin, Koutsoupias, Papadimitriou, 19].

Bitcoin's Proof-of-work

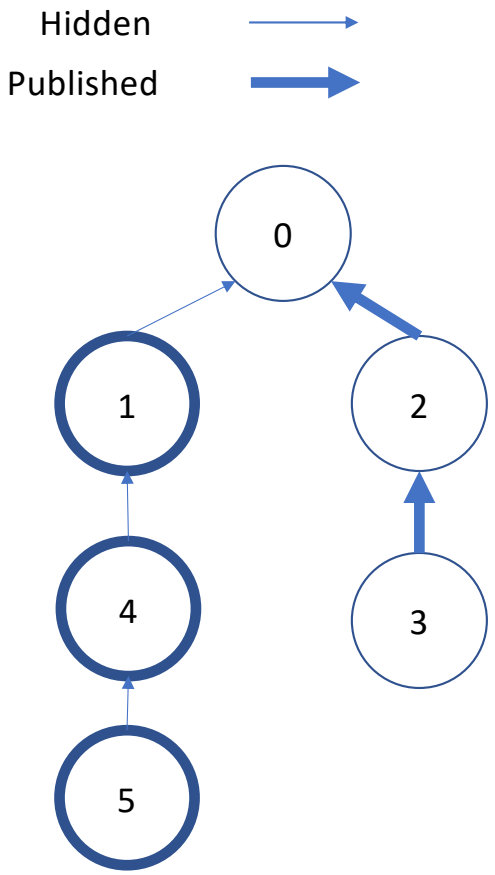
- Mining tournament.
  - First to solve the cryptopuzzle receives the **privilege** to create a block.
  - The creator of each block currently receives 6.25 Bitcoins.



Random oracle model: for all x, $Hash(x) \sim Unif(2^{256})$

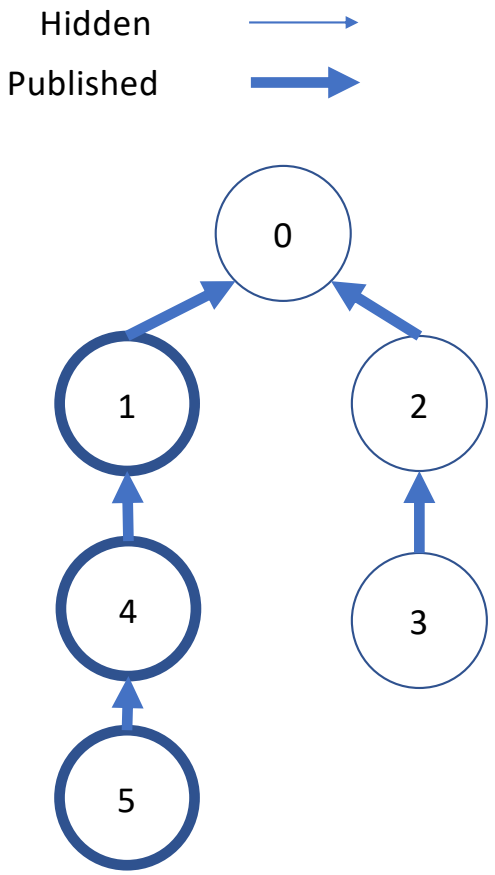
Model

- Setup
  - Miner $k \in [M]$, has mining power $x(k)$ and $\sum_k x(k) = 1$.
  - At all times, miner k is aware of directed tree $G(k)$.
- Game
  - Each time step t, sample miner k from distribution $x(1), x(2), \ldots, x(M)$ to create block t.
    - $k \in [M]$ creates node t with a single directed edge to any node in $G(m)$.
  - Each time step, every miner k can broadcast any v in $G(m)$.
  - Game stops when $G(m)$ has height H for some m.
- Payoff
  - $P(m) = \frac{\#\ nodes\ in\ the\ longest\ path}{H}$ (take $H \to \infty$).
  - **Sanity check:** because of Bitcoin's difficult adjustment H is proportional to time.



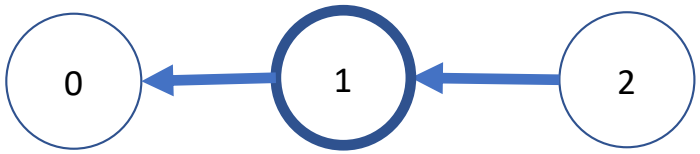
Model

- Setup
  - Miner $k \in [M]$, has mining power $x(k)$ and $\sum_k x(k) = 1$.
  - At all times, miner k is aware of directed tree $G(k)$.
- Game
  - Each time step t, sample miner k from distribution $x(1), x(2), \ldots, x(M)$ to create block t.
    - $k \in [M]$ creates node t with a single directed edge to any node in $G(m)$.
  - Each time step, every miner k can broadcast any v in $G(m)$.
  - Game stops when $G(m)$ has height H for some m.
- Payoff
  - $P(m) = \frac{\#\ nodes\ in\ the\ longest\ path}{H}$ (take $H \to \infty$).
  - **Sanity check:** because of Bitcoin's difficult adjustment H is proportional to time.



Honest mining

- Whenever miner $k \in [M]$ creates block t, point t to the block of highest height in $G(k)$. Publish block t immediately.

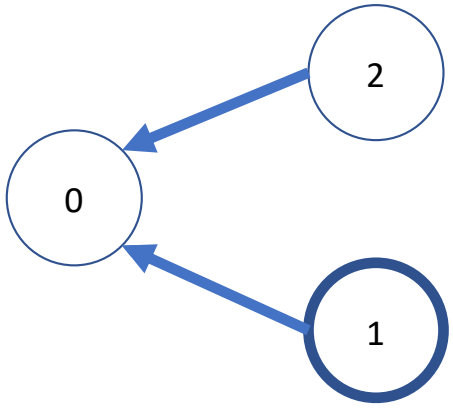


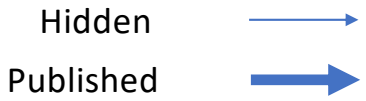
- [Nakamoto 08] Informally claimed that if $x(k) < 1/2$, honest mining is a **Nash equilibrium**.
- [Eyal, Sirer 14] [**Theorem**] Honest mining is not a Nash equilibrium if $x(k) > 1/3$ for some k.
- [SSZ16, **F**W 21][**Theorem**] Assuming $x(k) \leq 0.308$ for all $k \in [M]$, then honest mining is a **Nash equilibrium**.

Lucky Selfish Mining [Eyal, Sirer, 13’]

- Always tie breaks in your favor.
- Whenever you create a block, point to the longest chain and hide.
- Whenever they create and publish a block, publish one hidden block (if you have).

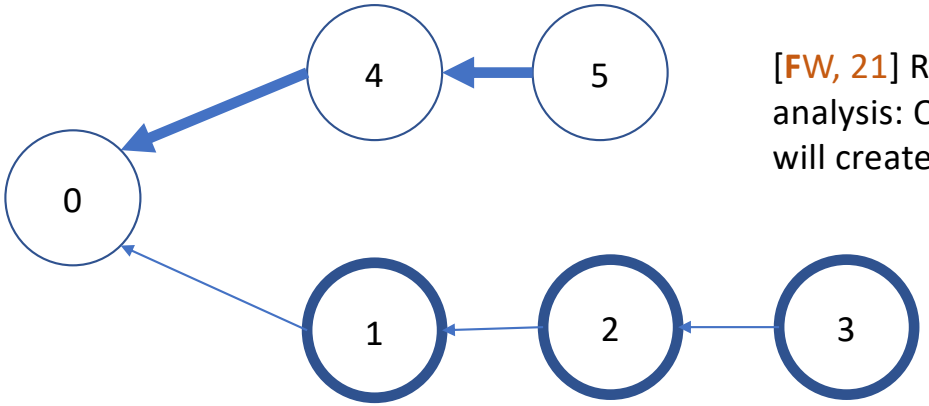
- **Payoff**: Assuming everyone else is honest mining, lucky selfish mining creates $x(k)/(1-x(k))$ fraction of blocks.





Unlucky selfish mining [Eyal, Sirer, 13’]

- Always tie breaks against you.
- Main idea: once you have a lead of 2 hidden blocks.
  - Mine in your longest chain until the lead reduces to 1 block.
  - Once the lead is 1 publish all hidden blocks.
- **Payoff**: better than honest mining if $x(k) > 1/3$ (everyone else is honest).



[FW, 21] Random walk analysis: On expectation you will create $x(k)/(1-2x(k))$.

Outline

- Mining games.
- **Transaction fees.**
- Mining pools.
- Difficult Adjustment.

The resource allocation problem

- What if there are more transactions than block space?
- [Nakamoto, 08] Transactions compete in a first-price auction.
  - Miner that creates the block receives all the auction revenue.

Ethereum Miners Earned Record $830M in January

CONTRIBUTOR

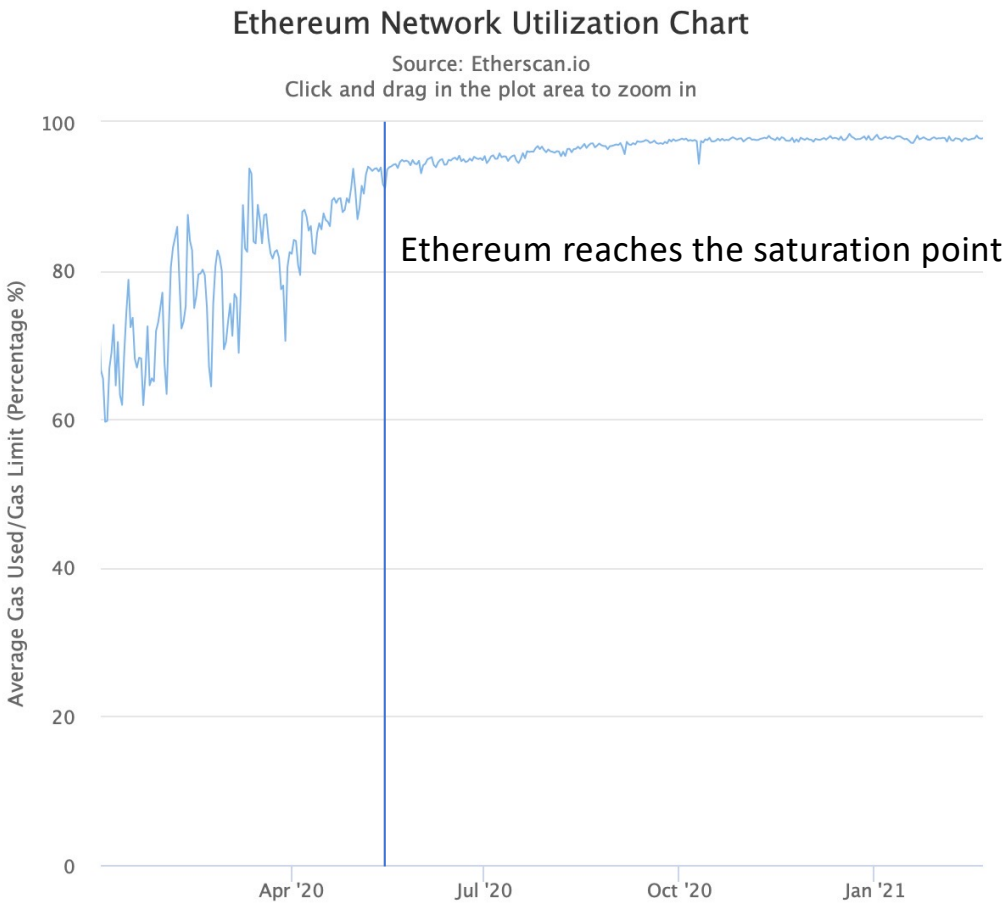
William Foxley — CoinDesk

PUBLISHED

FEB 2, 2021 11:15AM EST

- Ethereum miners have been a primary beneficiary of the fee spike. The industry earned some $830 million in ether last month with 40% attributed from fees alone.

Block space is a limited resource



Alternative transaction auctions

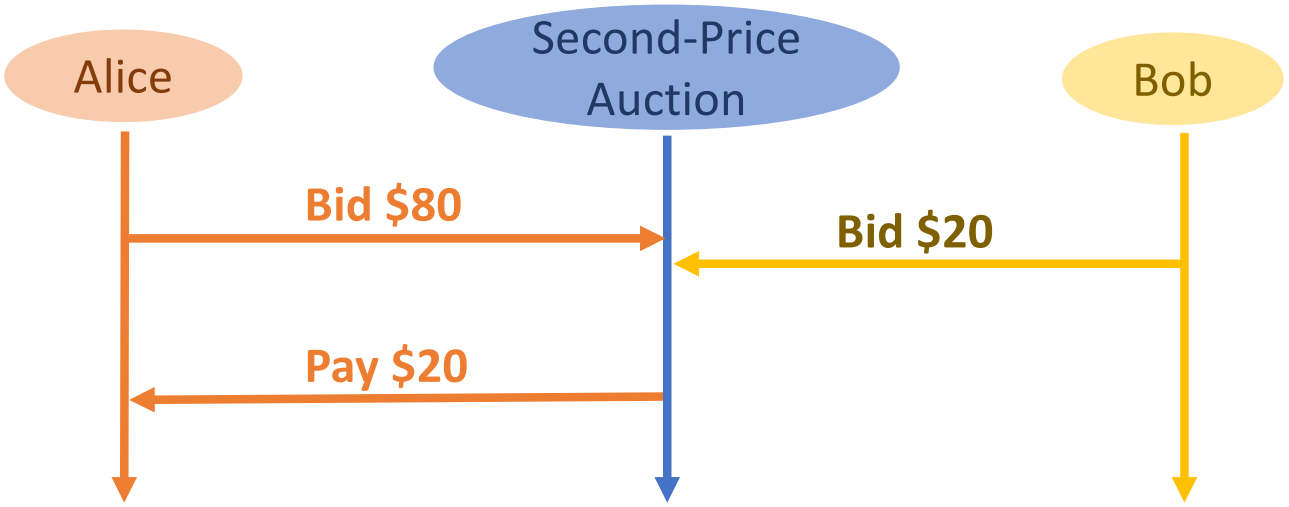
- First-price auctions are **unstable** when there is excess demand.
  - If bid too low, transactions can take hours for confirmation.
  - If bid too high, you risk overbidding.
- Monopolistic price auction [LSZ, 2019].
  - Sort bids $b_1 > b_2 > \ \ldots > b_n$ and allocate slots to the k highest bids

$$k = \arg\max_{i \in [n]} i \cdot b_i$$

  - Bidders $1, 2, \ldots, k$ receive block space pays b_k to the miner.
  - Concern: network **latency** degrades as block size increases.

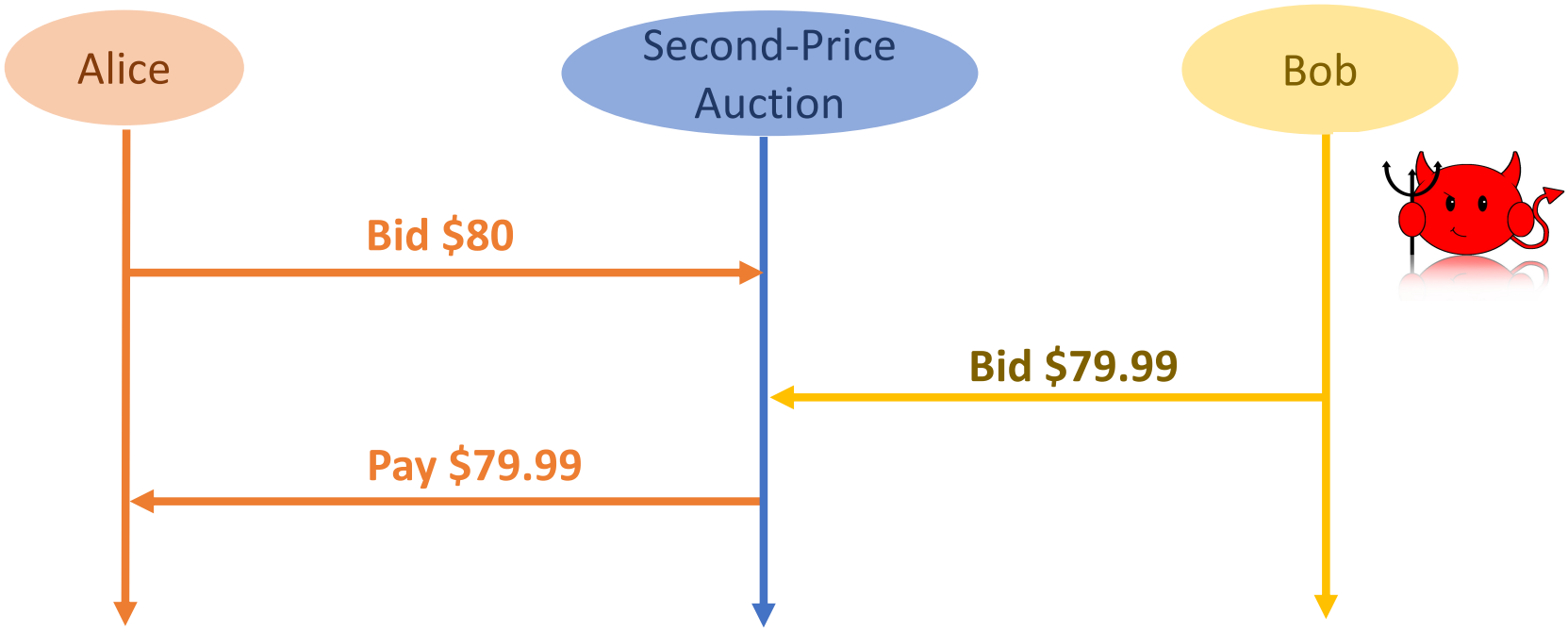
Why not use a second-price auction?

- [Vickrey 61] Auctioneer collects bids privately.
- Winner pays the second highest bid.



Vickrey. Awarded the 1996 Nobel prize in economics.

What can go wrong in a public blockchain



Algorithm designer as an adversary

The New York Times

F.B.I. Opens Investigation of EBay Bids

Suspicion of Shills Rises as Web Auctions Grow

By JUDITH H. DOBRZYNSKI

The Register®

Business ▸ Policy

eBay jewellery store fined $400,000 for shill bidding

eBay reports offender to authorities

By Lester Haines 11 Jun 2007 at 11:17 18 SHARE ▼

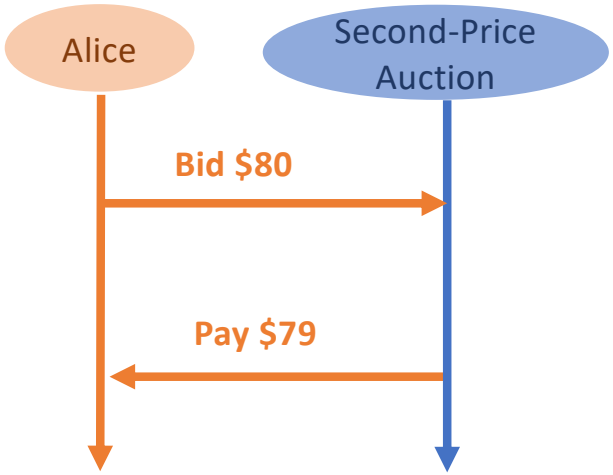
ARTnews Est. 1902

Legislators Seek to Stop 'Chandelier Bidding' at Auction

BY *Daniel Grant* ORIGINALLY PUBLISHED 09/04/07

Credible Auctions [Akbarpour and Li, 20]

- The **ascending price auction with reserves** is the **ONLY** auction**:**
  - Incentive compatible for bidders.
  - Incentive compatible for the auctioneer (i.e., credible).
  - Revenue optimal.
  - **BUT** the communication complexity is **unbounded**!
- Cryptographic auctions [FW, 20]
  - Avoids their impossibility result.
  - **BUT** communication complexity $\Omega(n)$.
  - We really want communication complexity $\mathrm{O}(m)$.



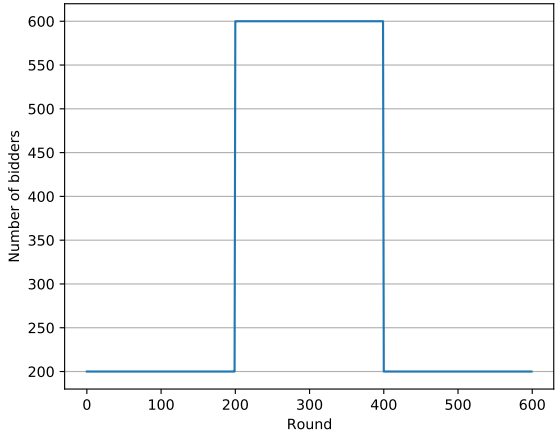
Model: Transaction fee auctions [FMPS, 21]

- Each time step t, an unknown # bidders n arrive with values drawn i.i.d. from some unknown distribution D.
  - Each bidder submits a bid b_i.
  - Information asymmetry: Miners know n and D, but the blockchain doesn't.
- **Sample miner k from distribution $x(1), x(2), \dots, x(M)$.**
- Miner k chooses a set of bids $B_t \in 2^{[n]}$ to receive block space.
- The mechanism observes B_t:
  - Charge payments $p_i(B_1, B_2, \dots, B_t)$ from each bidder i in B_t.
  - Miners receive all payments.

Dynamic Posted-Pricing

1. Each time step t, the mechanism **sets** posted-price q_t.
2. The active miner **allocates** space to at most m bidders $i \in B$ with $b_i \geq q_t$. Break ties **randomly**.
3. Each bidders $i \in B$ pays q_t to the active miner.
4. The mechanism computes $q_{t+1} = T(q_t, B)$ (**update rule**).

Supply of $m = 100$ blockchain slots. Distribution values a drawn from: $\Pr[v > x] = e^{-x}$.



Allocation-based update rule (EIP-1559)

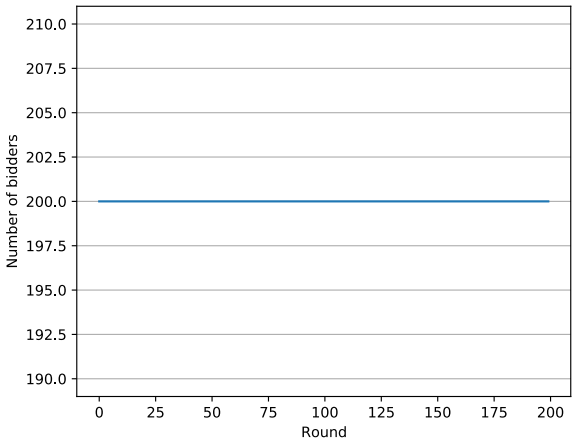
- When $|B| = 0$, $\mathrm{T_A}(\mathrm{q}, B) = (1 - \alpha)q$.
- When $|B| = \frac{m}{2}$, $\mathrm{T_A}(\mathrm{q}, B) = q$.
- When $|B| = m$, $\mathrm{T_A}(\mathrm{q}, B) = (1 + \alpha)q$.

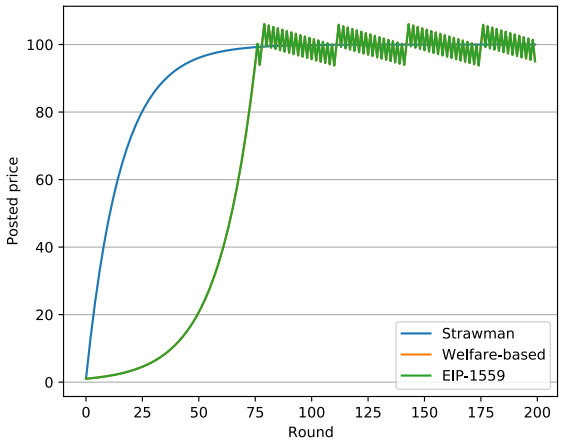
$$T_A(q, B) = \alpha \frac{2|B|}{m} q + (1 - \alpha)q$$

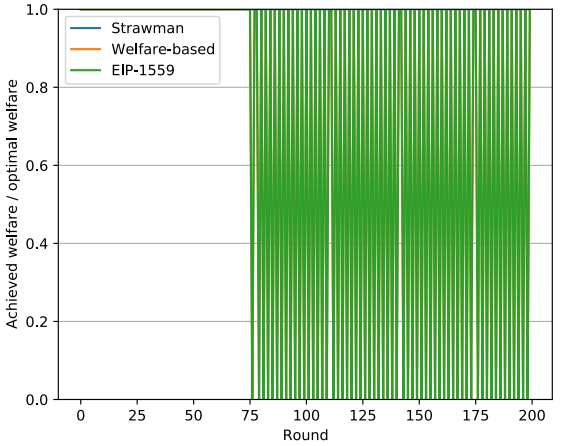
- Hence, the goal is to keep utilization at $\frac{m}{2}$.

Not always stable

Supply of $m = 100$. Distribution values a drawn from: $\Pr[v = 100] = 1$.



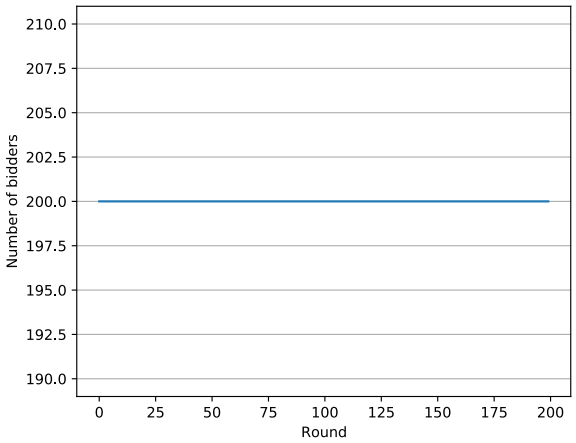


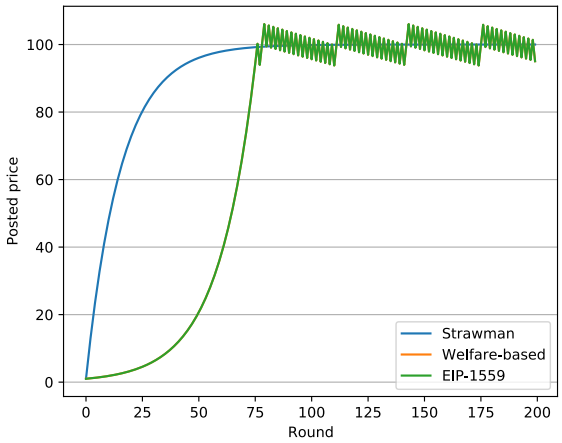


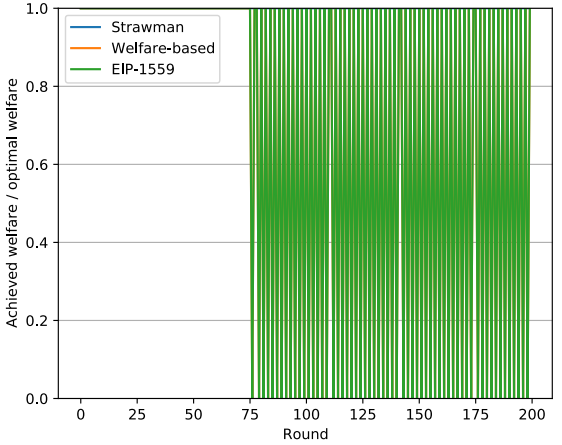
- **When $|\boldsymbol{B}| = \mathbf{0}, \boldsymbol{T_A}(\mathbf{q}, \boldsymbol{B}) = (\mathbf{1} - \boldsymbol{\alpha})\boldsymbol{q}$.**
- When $|B| = \frac{m}{2}$, $\mathrm{T_A}(\mathrm{q}, B) = q$.
- **When $|\boldsymbol{B}| = \boldsymbol{m}, \boldsymbol{T_A}(\boldsymbol{q}, \boldsymbol{B}) = (\mathbf{1} + \boldsymbol{\alpha})\boldsymbol{q}$.**

Strawman update rule

Supply of $m = 100$. Distribution values a drawn from: $\Pr[v = 100] = 1$.



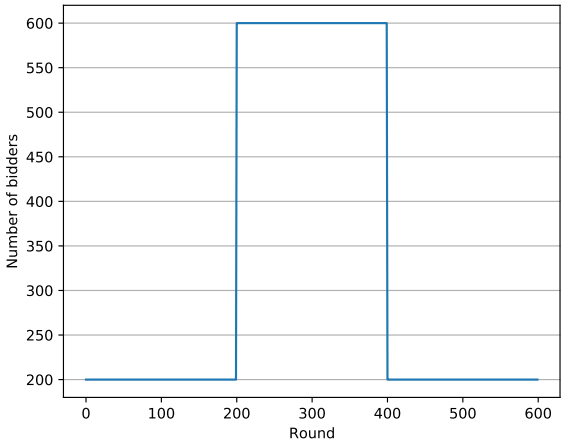


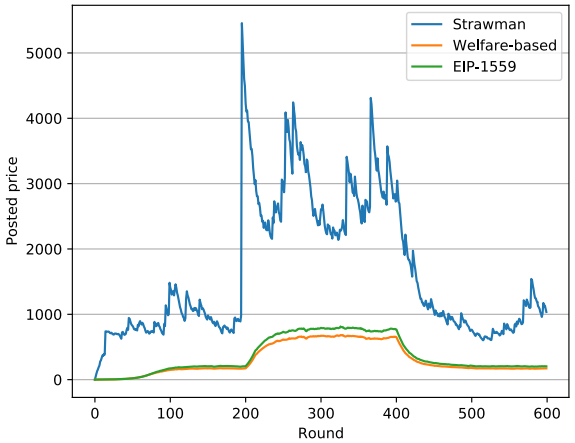


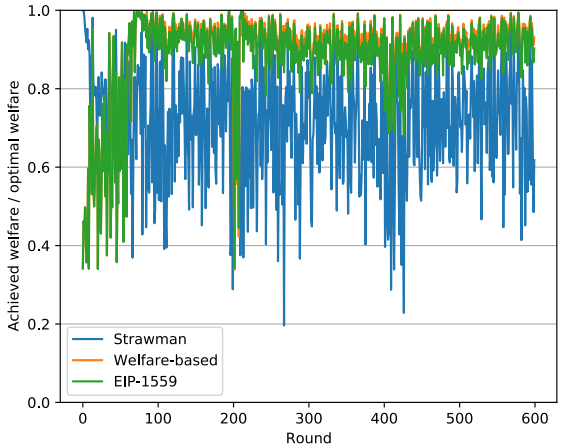
$$T_S(q, B) = \underbrace{\alpha \frac{1}{m} \sum_{i \in B} b_i}_{\text{Welfare}} + (1 - \alpha) q = \underbrace{\alpha \frac{1}{m} \sum_{i \in B} (b_i - q)}_{\text{Surplus}} + \left(1 - \frac{\alpha(1 - |B|)}{m}\right) q$$

High variance under heavy-tailed distributions

Supply of $m = 100$. Distribution values a drawn from: $\Pr[v > x] = \frac{1}{x}$.







$$T_S(q, B) = \underbrace{\alpha \frac{1}{m} \sum_{i \in B} b_i}_{\text{Welfare}} + (1 - \alpha) q = \underbrace{\alpha \frac{1}{m} \sum_{i \in B} (b_i - q)}_{\text{Surplus}} + \left(1 - \frac{\alpha(1 - |B|)}{m}\right) q$$

Stability

- Define the expected value (with respect to having n i.i.d. bids drawn from distribution D and miner's randomness):

$$E_T(q) = E[T(q, B(q))].$$

- An update rule T is **asymptotically stable** with respect to distribution D^n if the following fixed-point iteration converges for any positive q:

$$q, E_T(q), E_T\big(E_T(q)\big), \ldots \to q^*$$

Equilibrium price

Stability of allocation-based update rule

- [**Theorem**] Assume distribution D^n. If the revenue curve is **L-Lipschitz** and **strictly concave** on $[0, V]$, then the **allocation-based update rule** is asymptotically stable with respect to D^n whenever $\alpha < 1/(2L/m + 1)$. If the **equilibrium price** is q^*, $Welfare(q^*) \geq OPT/4$.

$$E_{T_A}(q) = \frac{2\alpha}{m} E[q|B|] + (1-\alpha)q$$

Expected revenue with posted-price q

$$Welfare(q) = E[\sum_{i \in B(q)} v_i].$$

Summary: Transaction fees

- Open direction: other notions of stability?
  - Study equilibrium when $m \to \infty$ and $n = O(m)$.
- Open direction: design incentive compatible transaction auctions.
  - Challenge: adoption and side payments.
  - E.g., we could divide the revenue of block B among B and the next 100 blocks.
  - **BUT** miner of block B could request side payments.
  - E.g., not pay transaction fees to miners (except for a reserve price).
  - **BUT** how do we incentivize miners to participate?
- [Babaioff, Dobzinski, Orzen, Zohar 12] How to propagate transactions in P2P?
  - Challenge: If I don't broadcast transactions, I face less competition.

Outline

- Mining games.
- Transaction fees.
- **Mining pools.**
- Difficult Adjustment.

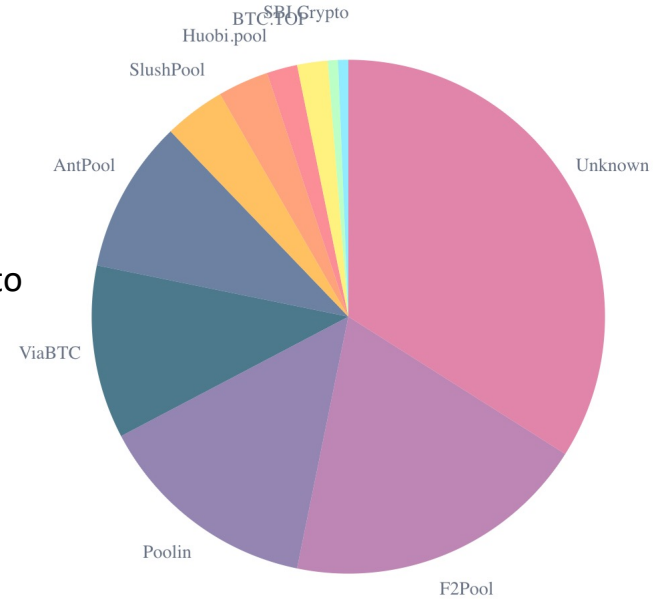
Mining Pools

Based on ACM EC tutorial from 2018 by Matt Weinberg

- A single blocks is created every 10 minutes by the entire network.
- **High** variance in block reward.

- Alternative is to join mining pools.
  - A manager coordinates the mining and splits reward proportionally to contribution.
  - Expected reward is the same but lower variance.

- How pools work:
  - Manager gives their ID, M_{Pkey} to participants.
  - Participant search for a PoW $H(block) < 2^{64}$ with M_{Pkey}.
  - Send partial PoW $H(block) < 2^{50}$.
  - Manager keeps record of all partial PoW.
  - Whenever the pool creates a PoW, all participants share the reward:
    - Miner k receives $F_k(\#\ partial\ PoW)$.
    - How to pick a good F_k [Schrijvers, Booneau, Boneh, Roughgarden 16].



Source: blockchain.com

Folklore knowledge

- Make sense to attack a pool to recruit its participants.
- Attack: send all partial proofs, but if you find a valid block, throw it away.
  - Sanity check: can't steal the block for yourself because it was created using M_{Pkey}.
    - Can't use your on public key because it will not be a valid partial PoW.
- Downside: wasting your own mining power.
- [Eyal 14] If attack with right proportion, can profit and hurt other pool.

Mining pool infiltration attack

Two pools, **A** and **B**, each with 50% of mining power.

- **Pool A** attacks **Pool B**, with 25% of total mining power.
- **Pool A** gets 33% of **Pool B**'s reward, and 100% of **Pool A**'s rewards.
- **Pool A** finds 33% (= 25/75) of valid pools.
- **Pool B** finds 66% (= 50/75) of valid blocks.
  - **Pool A** gets 1/3 + 1/3 2/3 = 5/9 of the total reward.

Summary: Mining pools

- Mining pools increase centralization.
- [Cong He Li, 18] But lower risk attracts more miners to the game.
- [Miller, Kosba, Katz, Shi 14] There is clever solutions that would kill mining pools.
  - These solutions were not adopted by major cryptocurrencies.

- Open directions: understand the economics of mining pools.
  - E.g. economic valued added (reduced risk) versus subtracted (centralization).

Outline

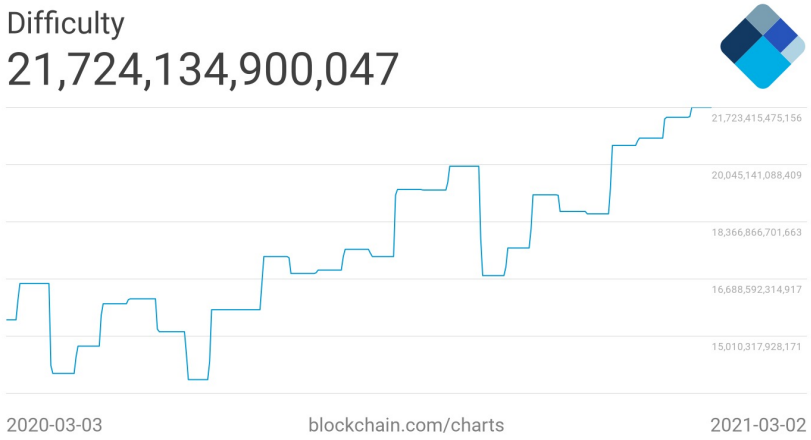
- Mining games.
- Transaction fees.
- Mining pools.
- **Difficult Adjustment.**

Bitcoin - Difficult Adjustment Algorithm (DAA)

- Let W be current difficulty.
- Let $T = (T_1, T_2, \dots)$ denote the **physical time** it takes to create the i-th block.
- Difficult from blocks $(i+1)n$ to $(i+2)n-1$ is:

$$W_{i+1} = W_i \frac{\sum_{j=n\cdot i}^{(i+1)n-1} T_j}{T^* n}$$

- $T^* = 10\ min, n = 2016.$



Bitcoin cash DAA

- Bitcoin cash was a hard fork on Bitcoin intended to increase block sizes.
- Uses a slide window of size 128 as its DAA.
- [NOH 20] Compares the stability of Bitcoin DAA and Bitcoin cash DAA.
- In practice, Bitcoin cash DAA appears to be more vulnerable to manipulation.
- [FKKP] If miners have flexibility to change the difficult and there is too many miners, the only pure equilibriums are chaotic.

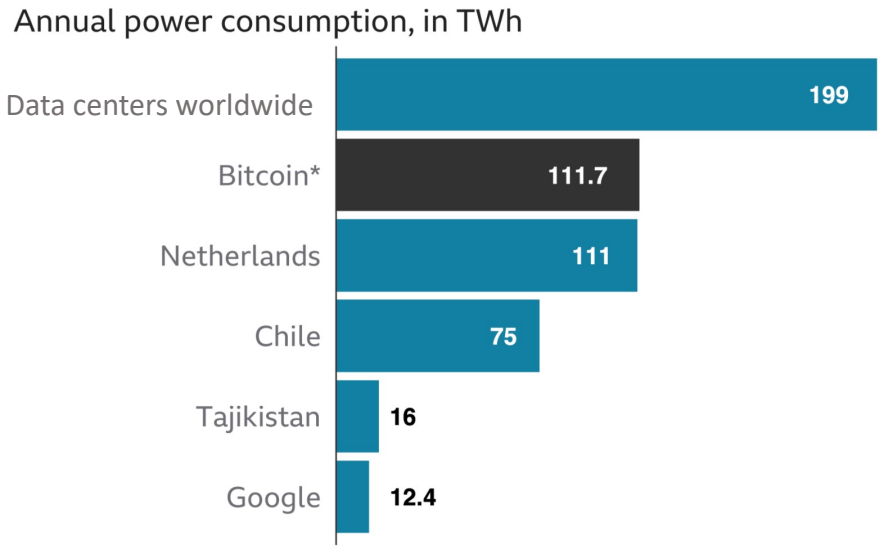
Final remarks

Summary

- Classic mining games
  - Proof-of-work [ES 14, SSZ 16, KKKT 16]
  - Proof-of-stake [**F**W, 21].
  - Open questions: How transaction auctions and applications impact miner's incentives?
- Transaction fee auctions
  - [LSZ 19, **F**MPS 21, CKWN 16]
  - Open questions: how collusion impacts existing proposals?
  - Study other notions of stability for dynamic posted-price mechanisms.
  - Study stability in the regime $\mathrm{m} \rightarrow \infty$.
- Mining pools
  - Attacks on mining pools [E 14].
- Difficult adjustment algorithms (DAA)
  - [FKKP 19, NOH 20]
  - Questions: are DAAs stable under strategic mining?

Environmental cost of Bitcoin

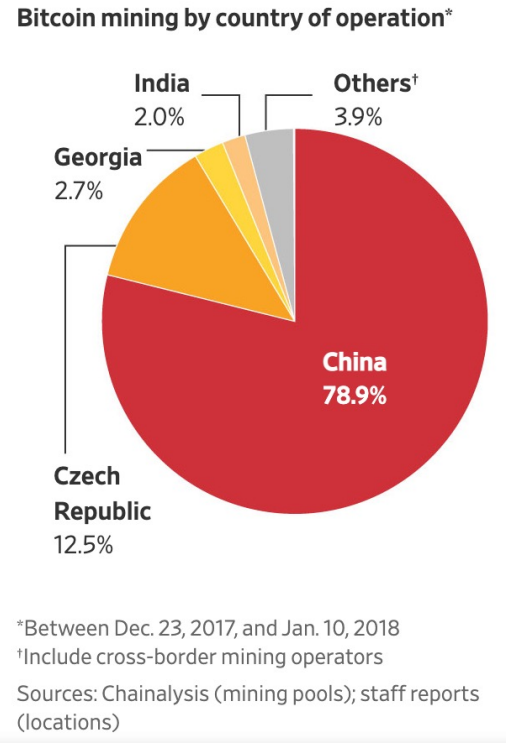
- Bitcoin's consensus algorithm is based on proof-of-work.
  - Energy intensive yet can only process 7 transactions per second.



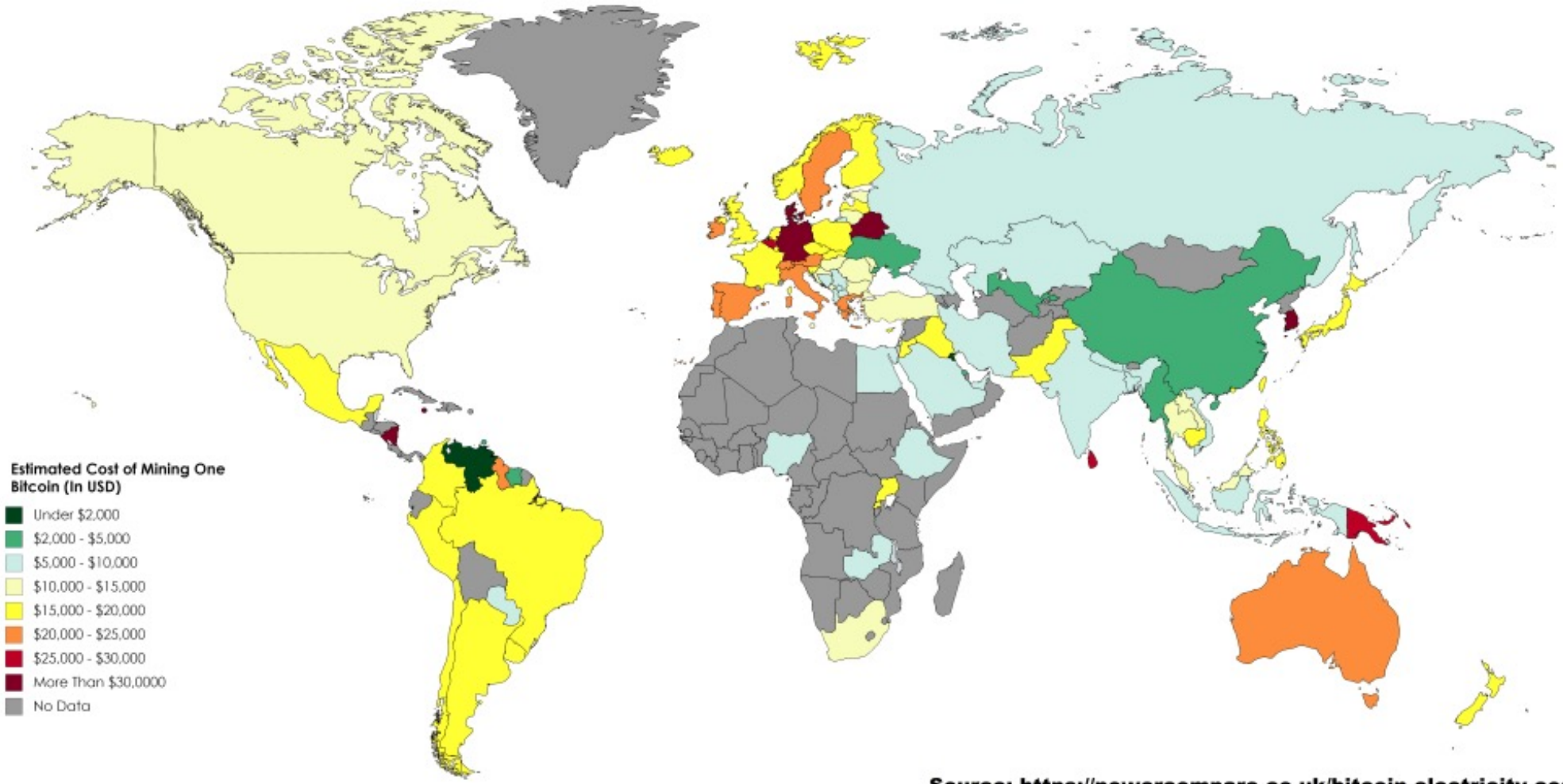
Source: BBC News. January 2021.

Mining centralization in PoW

- Profitability of PoW mining varies by geographic region.
  - Energy and hardware cost.
- [Arnosti, Weinberg, 18] In equilibrium:
  - If a miner has 20% lower cost than other miners, then she will control at least 20% of the mining power.
  - Economics of scale also generates high centralization.



Estimated Electricity Cost Of Mining One Bitcoin By Country

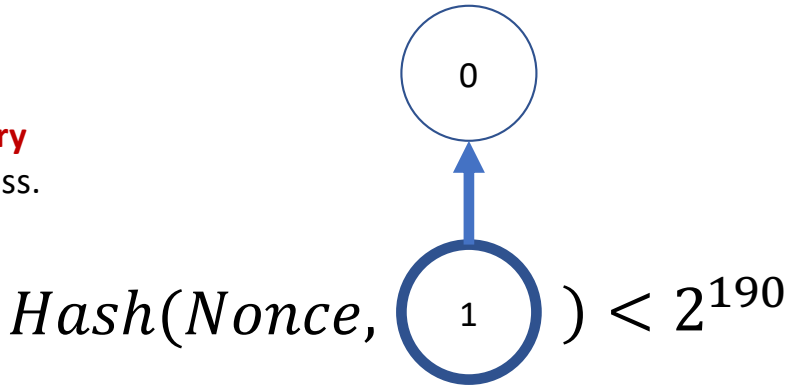


Source: https://powercompare.co.uk/bitcoin-electricity-cost/

Beyond Proof-of-Work

- Proof-of-work (PoW) is energy intensive
  - Mining election: cryptopuzzles (One CPU cycle gets One vote).

Takeaway: Cryptopuzzles are a **very good** source of pseudo-randomness.



- Proof-of-stake (PoS) is an energy friendly alternative.
  - Mining election: sample a uniformly random coin (One coin gest One vote).
  - Proof-of-stake Mining games [**F**, Weinberg, 21]

References: Mining games

- Majority is not enough: Bitcoin mining is vulnerable [ES 13].
- Proof-of-stake mining games with perfect randomness [FW 21].
- Optimal selfish mining strategies in Bitcoin [SSZ 16].

References: Mining pools

- The miner's dilemma [E14].
- Incentive compatibility of bitcoin mining pool reward functions [SBBR 17]

References: Transaction fee auctions

- Credible auctions: A trilemma [AL 20].
- On the instability of Bitcoin without the block reward [CKWN 16].
- Credible, truthful, and two-round optimal auctions via cryptographic commitments [FW 20].
- Dynamic posted-pricing mechanisms as a blockchain transaction fee mechanism [FMPS 21].
- Redesigning Bitcoin's fee market [LSZ 19].
- An incentive analysis of some bitcoin fee mechanisms [Y 20].
- On Bitcoin and red balloons [BDOZ 11].

References: Others

- Bitcoin: a natural oligopoly [AW 18].
- Energy equilibria in proof-of-work [FKKP 19].
- The byzantine generals problem [LSP, 82].
- An economic analysis of difficulty adjustment algorithms in proof-of-work blockchain systems [NOH 20].