Lecture 4 - Mining Incentives, Challenges and Future Directions

Rylan Schaeffer and Vincent Yang

April 6, 2016

Note: This lecture is based on Princeton University's BTC-Tech: Bitcoin and Cryptocurrency Technologies Spring 2015 course.

Review

- Need method to achieve distributed consensus
- One option is proof-of-work, in which we balance two opposing goals:

  challenging to validate transactions, to prevent interference

  rewarding to validate transactions, to encourage people to participate
- Accomplished through hash puzzles i.e. calculate nonce r such that $H(previoushash|tx_1|tx_2|...|tx_n|tx_s|r)$ has target number of leading zeroes
- Two incentives:

  Block reward i.e. tx_s is a new coin

  Transaction fees

To Be A Miner

1. Listen for transactions
2. Maintain block chain and listen for new blocks
3. Assemble new block
4. Mine that block i.e. find the nonce r
5. Hope your block is accepted
6. Profit

Note: Not all miners are working on the same problem! Miners (almost) always have unique set of transactions. Even if the transactions are identical, they're trying to send the transaction rewards and new block to themselves.

Protocol dictates that, on average, mining a new block should take about 10 minutes, adjusted about every two weeks.

1 Mining Hardware

- SHA-256 requires 32-bit words, 32-bit modular addition, some bitwise logic
- Central Processing Unit mining works, but is slow
- Graphics Processing Unit mining is faster, but wastes floating point unit and consumes more power.
- Field-Programmable Gate Arrays mining is comparable to GPUs, better with bitwise logic, can be easily packed together, can be controlled from one centralized unit. Difficult to optimize 32-bit step addition
- Application-Specific Integrated Circuits: best of all possible worlds

2 Mining Costs

- Technology quickly becomes antiquated
- Cheaper in cold locations with cheap electricity
- Shifted to professional companies

3 Succeeding as a Miner

Suppose you spent $6,000 on mining rig. Assuming you expect to find one block per 14 months, worth 25 bitcoins.

- Model as Poisson process
- Reduced variance when mining together

4 Mining as a Group

- Track each person's computational contribution, using "near misses" as a proxy measurement
- Model 1: Pay per share. Pool manager pays flat fee per near miss above a set difficulty. No incentive to send valid blocks. High risk for pool manager.
- Model 2: Pay proportionally. Low risk for pool manager.
- (Display mining consolidation visually)

5 Pros and Cons of Mining

- Allow individuals and groups to make money, where otherwise impossible
- Threatens stability of Bitcoin, since mining pools are a form of centralization

6 Incentives and Strategies

- Much game theory, including
  - which transactions to include
  - which block to mine on
  - choosing between blocks of the same height
  - when to announce new blocks

- Forking attack: Double spending possible if thief controls ¿51% of the network. Spends money, seller accepts because chain is long enough, then thief extends alternative chain, effectively erasing the previous transaction. May not require 51%
- Block-withholding attack: Solve a block, wait til someone else solves a block, then release yours to prevent their from being accepted. Increases your effective share of rewards.
- Blacklisting: Can refuse to process transactions from certain public keys

7 Alternatives to Mining Puzzles

- Solving puzzles takes tremendous energy which is basically wasted
- Alternative puzzle requirements:

  Easy to verify solutions

  Adjustable difficulty to control rate of mining

  "Program freeness": chances of winning should be proportional to computational power. Also known as memoryless process.
- ASIC Resistant Puzzles

  Goal: Decentivize consolidation by making individual computers competitive with ASICs.

  Method: memory-hard puzzles i.e. puzzles that require large amount of memory to solve or memory-bound puzzles i.e. puzzles that are slowed down by memory access time.

  Example: Scrypt. Used in Lytecoin. Two steps.

  1) fill large buffer of RAM with pseudorandom data i.e. for i = 1 to N: V[i] = SHA-256(V[i-1])

  2) read and update the memory in pseudorandom order i.e. X = SHA-256(V[N-1]). for i = 1 to N, j = X % N; X = SHA-256(X ^ V[j])

  Memory-hard because if V isn't stored in memory, recomputing X = SHA-256(X ^ V[j]) takes $O(n^2)$ instead of $O(n)$ as j is generated pseudorandomly.

  Unfortunately, requires as much money to verify as does to compute

  Eventually Litecoin revealed that Scrypt ASICs were better than ordinary hashing ASICs

  Example: Cuckoo Cycle. Based on difficulty of finding cycles in a graph generated from a cuckoo hash table.
- Moving Target Puzzles: change puzzles occasionally so that ASICs are worthless

8 Proof-of-Useful-Work

Idea: Use computing energy for some benefit to society instead of wasting it. Examples include:

- Distributed computing projects e.g. SETI at Home, Folding at Home (protein structure folding)

  Problem: solution space was not equiprobable
- Great Internet Mersenne Prime Search

  Problem: solutions too rare

Primecoin uses proof-of-useful work, by requiring miners to find Cunningham chains of prime numbers. Questionable how useful this is.

9 Proof-of-Storage

Idea: use miners' storage power to help with some archival project/distributed computing problem
Used in Permacoin as follows:

- Let F be a large file that we want stored e.g. Large Hadron Collider backup data of several hundred petabytes
-

chapter 8. to be finished...