Decentralised Mining Pool for Bitcoin (Draft 0.1)

Kulpreet Singh

February 2021

Abstract

Bitcoin p2pool's usage has steadily declined over the years, negatively impacting bitcoin's ability to remain decentralised. The primary problems with p2pool were twofold. First, the variance in earnings for miners didn't reduce with the increase in hashrate particapting in p2pool. Secondly, making payouts to miners required a linearly increasing blockspace with the increase in the numer of miners on p2pool. Building a DAG of miner's shares and the use of payment channels are two proposals trying to alleviate these problems faced by p2pool. In this document, we present a unified solution that uses a directed acyclic graph to track miners shares and uses payments channels to reward miners. The shares calculation can be carried out by any node on the p2p, and the rewards are paid out by a pool operator. Using the payment channels construction neither the pool operator nor the miners can cheat each other. We show that our approach is incentives compatible and reduces variance in earnings for miners. We also propose a solution for trading the miner's proof of work for BTC in an open market.

1 Motivation

P2Pool [1] helped decentralise bitcoin by enabling miners to select which transactions they mined and thus avoid any potential censorship by pool operators. However, the construction used by P2Pool faced a number of problems that lead to miners abandonding the pool.

1. Large variance in earnings for miners.
2. Large number of dead on arrival blocks.
3. Large block space requirement.

The first two problems are a direct consequence of the shares block rate limited to 30 seconds. With only one block possible every thirty seconds, any increase of hashrate on P2Pool resulted in shares competing to be the next block in the p2pool chain.

There is a clear tension in play here, increasing the block rate frequency doesn't scale the throughput of the pool in terms of number of shares found, as most of them are orphaned and the miners not being rewarded for those orphans. Ethereum's inclusive protocols [4] help alleviate the problem for the Ethereum blockchain, where small pools can work with a reduced variance in their rewards as shown in the analysis by McElrath [5].

Knowing the challenges faced by P2Pool, we list the goals of a new decentralised mining pool as:

1. Lower variance for miners to enable the long tail of independent miners.
2. Indepedent miners that build their own blocks.
3. Payouts for miners with constant size block space requirement.
4. Provide building blocks for a hash rate futures market.

2 Current Proposals

TerraHash Coin [5], Jute [9] and [8] are some of the attempts to use a DAG for faster block times. However, these works focus on changing the consensus layer of bitcoin itself. The ideas in these proposals allowed miners to produce shares that have conflicting transactions and then apply rules to find a set of transactions acceptable at various cuts of the DAG.

Instead we propose to build a DAG of miner shares to enable faster block times and using this DAG for calculating distribution of payouts between miners. We then propose using payment channels as defined by Belcher [2] to avoid using block space for making payouts to miners.

Apart from the work for increasing block rates using DAGs, Belcher [2] proposes a different idea to help decentralise mining deals with the problem of a paying miners from a decentralised mining pool. P2Pool uses the coinbase tranasaction of a block to pay out miners. Belcher shows how a scheme can be constructed using payment channels between federated hubs to pay miners after a block has been successfully mined. The payouts can be paid after a long enough period, similar to the 100 blocks requirements for spending from coinbase transactions. Miners can register with hubs where bitcoin has been locked in to open payment channels to miners. The construction presented by Belcher shows how both miners and hubs can't cheat and how the funders of the hub can earn a reward for funding the payment channels.

The ideas of the Hash Rate Futures and Payment Channels for Rewards Payouts together present a potential path for rebooting P2Pool. In the rest of the document we present a slightly modified version of TerraHash Coin and show how the various components can work toegether.

3 Decentralised Bitcoin Mining

In this section we present a modified version of TerraHash Coin and show how it can use payments channels with hubs to deliver a decentralised mining pool for bitcoin.

3.1 A DAG of Shares

The braiding the blockchain proposal [5] shows how smaller more frequent blocks can form a directed acyclic graph (DAG) of blocks, with each block pointing to one or more one previous blocks. Blocks in TerraHash Coin can have transactions repeated in different blocks. The proposal describes how repeated and potentially some double spend transactions can be resolved to decide on the state of the ledger at any cut of the DAG.

The rewards that miners earn in the proposal is a coin native to the braid blockchain and is called TerraHash Coin. This coin can then be swapped for Bitcoin. The proposal doesn't yet define how this native coin will be swapped by bitcoin. Some of the suggestions under discussion include using atomic swaps, burning the TerrahHash Coin, or using financial instruments like futures of the bitcoin's hash rate to swap TerraHash Coins for BTC.

We propose taking a slightly different approach, where the blocks of a DAG represent shares of the mining pool, use payment channels between miners and a hub for distributing payouts in proportion to the work done by the miners.

Each miner builds their own block, selecting transactions as they want, we call this block the WORK. The description of WORK is then disseminated to the p2p network of miners using the compact block specifications [3].

The miner then starts mining on WORK and generates SHARE. Each SHARE is mined at a difficulty level chosen by the miner. This difficutly can be dynamically chosen by the miner after each SHARE, depending on what the miner observes on the p2p network. This dynamic adjustment allows miners to adjust the rate at which they produce SHAREs. [TODO: Build a model to recommend an emission rate of these shares.]

Figure 1 shows the relationship between WORK and its SHAREs. Each WORK created by a miner has multiple SHAREs and they are both broadcast on the p2p network.

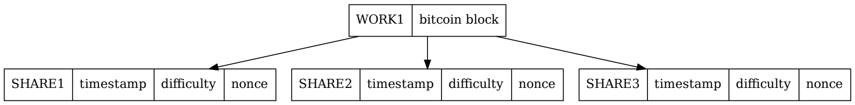

Figure 1: Each WORK generated and shared by a miner is then followed by the SHAREs the miner finds.

The nodes in the DAG are SHAREs mined at varied difficulty levels. Each SHARE that matches or exceeds the current bitcoin difficulty starts a new *epoch* for the p2p mining pool. Figure 2 shows l and r as the two valid bitcoin blocks that have been mined such that they meet bitcoin's difficulty at the time the block was mined, and all the blocks between l and r are in the same *epoch*.

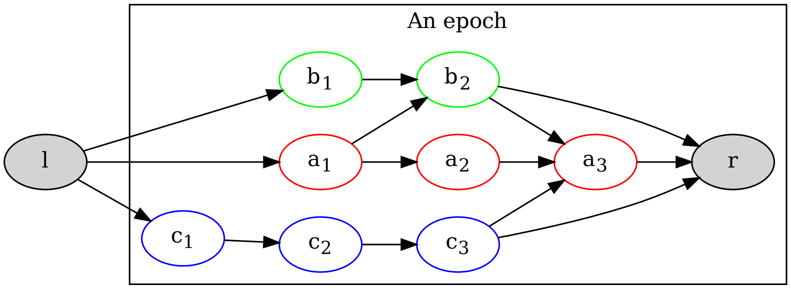

Figure 2: A epoch is defined as all the SHAREs mined between two bitcoin blocks. Here all the SHAREs between l and r are in the same *epoch*.

When a miner starts working on a SHARE it includes a reference to the highest known SHARE from all other miners that the miner has receieved valid shares from. Note, the miner also has access to WORK blocks from all participating miners. If a miner doesn't have the WORK block from another miner, then it rejects any SHAREs received from the other miner. This requires that miners is to the deteriment of the other miners as we show in Section 3.2.

In the next section we then describe how all peers compute their fair share of profits using the DAG of shares. We show how our reward computation algorithm is incentives compatible [7].

3.2 Incentives Compatible Rewards

Each participating node, which includes the miners and the hub, is able to the see the DAG of SHAREs as broadcast on the network. Each SHARE includes a reference to the blocks the miner was aware of when the SHARE was found. The reason for doing so is simple. If a miner a doesn't include the SHAREs of miner b, then b has a clear signal to stop including the SHAREs of a, and as we will see a miner wants that their SHAREs are referenced by other miners as only then they will be rewarded for their work.

The incentive in lay terms is that all miners should honestly include the SHAREs discovered by other miners, as otherwise they will most likely be excluded by other miners and they will lose the opportunity to be rewarded for their work. We call this the degenerative case of "isolated miners" and argue that miners have no incentives to act in this manner. Figure 3 shows a DAG where all three miners a, b and c are working independently. In such a situation when the miner a discovers a share and the reward is not shared with any other miner.

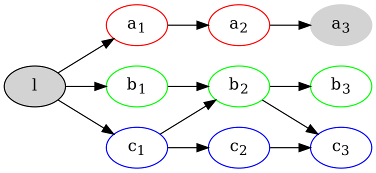

Figure 3: a discovers a share and the reward is not shared with any other miner.

With the above understanding of why miners will co-operate, we now state the rules to calculate how the block reward should be divided between miners.

1. Traverse the DAG in reverse order from the SHARE that found the latest bitcoin block to the previous bitcoin block found and collect a set of shares.
2. From the above set of shares remove all shares that don't have a reverse path to the previous bitcoin block.
3. Distribute the reward between miners weighted by the sum of the diffcultly of all SHAREs found by miners.

As an example consider the p2p network of miners a, b and c with the DAG of shares as shown in Figure 4. In the DAG the set of shares that receive reward proporitional to their difficulty are $\{a_i..a_5, b_1..b_3\}$. The shares $\{c_1..c_3\}$ do not receive any reward as they are not reachable from the bitcoin block, a_5, even if they are reachable from l.

For the second bitcoin block b_5 only the miners a and b receive rewards in proportion to the difficulties of their shares $\{b_4, b_5, a_6\}$. c doesn't receieve any reward for $c4$ as it is doesn't include a reference to the last found bitcoin block a_5.

With the above rules, it should be easy to prove that the rule reward is an incentives compatible reward function as defined by [ref:incentives compatibility], and we present an outline of proofs that will be be formalised in future work.

Incentive Compatibility Given the rules above, if a miner finds a bitcoin block the miner wants to get maximum reward possible based on all the shares it has found and therefore is incentivized to announce their SHARE as soon as they find it.

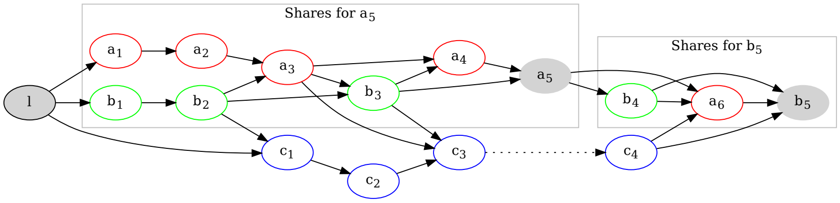

Figure 4: Two epochs in a DAG of shares mined by three mines — *a*, *b* and *c*. The shares in grey meet the bitcoin difficulty at the time they were mined.

Proportional Payments Since rewards are calculated at the end of an epoch, that is when all the valid shares found by all miners have been discovered, all miners are guaranteed payments for the shares that have reached the miner who found the block.

Budget Balanced Again, since rewards are paid at the end of the epoch, a hub pays out rewards without losing or retaining any amount.

3.3 Payment Channels

The Hubs and payments channels proposal is based on the proposal described by Belcher [2] with a change that there is a single hub and it is protected from DDoS by using techniques to enable responder anonymity in p2p networks [6]. With this change the hub acts just like any other miner on the network to be completely undifferentiated from other miners, thus protecting itself from DDoS, by requiring an attacker to attack the whole network.

The only difference is that Hubs calculate and distribute the rewards according to the incentives compatible rewards scheme described above. We also adopt the proposal that rewards are paid out after a 100 blocks of a block being mined.

We would like to extend the proposal to include funding of hubs by more than one party in a trustless manner, so that parties with small BTC parties can engage in the peer to peer mining economy. This will encourage faster adoption of our p2p mining network and make the network stronger against DDoS attacks with multiple hub operators available online.

4 Future Work

4.1 Proofs

We want to use the model presented by Boneh et.al. to provide proofs for how the rewards distribution is incentives compatible.

4.2 Simulations

Before we work on implementing they system, our next step is to simulate p2p mining network using ns-3 [ref] and make informed decisions about how large a network each hub will want to support. The observations we want to make are how large a p2p network can be sustained without an increase in work lost by miners. Each hub and p2p network can grow as long as miners are communicate WORK and SHARES with each other with bounded latency and can limit their lost work. With a simulation we want to find out the bounds of these.

4.3 Specifications and Implementation

We want to specify the p2p protocol messages and the rules more precisely. We also plan to implement the specifications and we expect the two tasks to proceed hand in hand.

References

[1] P2pool. `https://en.bitcoin.it/wiki/P2Pool`.

[2] belcher. Payment channel payouts: An idea for improving p2pool scalability. `https://bitcointalk.org/index.php?topic=2135429.0`.

[3] Corallo, M. Compact block relay. `https://github.com/bitcoin/bips/blob/master/bip-0152.mediawiki`.

[4] Lewenberg, Y., Sompolinsky, Y., and Zohar, A. Inclusive block chain protocols. In *Financial Cryptography and Data Security* (Berlin, Heidelberg, 2015), R. Böhme and T. Okamoto, Eds., Springer Berlin Heidelberg, pp. 528–547.

[5] McElrath, B. Decentralized mining pools for bitcoin. `https://www.youtube.com/watch?v=91WKy7RYHD4`.

[6] Scarlata, V., Levine, B., and Shields, C. Responder anonymity and anonymous peer-to-peer file sharing.

[7] Schrijvers, O., Bonneau, J., Boneh, D., and Roughgarden, T. Incentive compatibility of bitcoin mining pool reward functions. In *Financial Cryptography and Data Security* (Berlin, Heidelberg, 2017), J. Grossklags and B. Preneel, Eds., Springer Berlin Heidelberg, pp. 477–498.

[8] Sompolinsky, Y., Lewenberg, Y., and Zohar, A. Spectre: A fast and scalable cryptocurrency protocol. Cryptology ePrint Archive, Report 2016/1159, 2016. `https://eprint.iacr.org/2016/1159`.

[9] Vorick, D. Jute: More scalable, more decentralized proof-of-work consensus. `https://github.com/Taek42/jute`.