CS 6466 2Chainz

Email: wdaviau@protocol.ai
CUNet ID: wtd37
Name: Wyatt Daviau
Date: March 25th 2018

2Chainz – Miner Dynamics Across Blockchains

Introduction

The goal of our analysis is to study the scenario in which miners are willing to switch between two blockchains that run the same protocol to maximize their net profits. We begin by defining a plausible parametrization of this system and then study potential chain-switching strategies that miners might employ in an attempt to increase their profits. We begin by trying to understand the profitability over time of the naive greedy approach. After this we plan to see whether more advanced mining strategies can increase profitability.

As these mining strategies will change the hash power distributions across the chains over time, a parallel goal is to quantify the effect any profitable strategies have on the difficulty and hash power distribution accorded to each chain over time.

Time permitting this will pave the way towards a preliminary analysis of the potential impact that protocol changes in one chain could have on controlling system mining power through incentives, potentially generating sustained oscillations between chain hash power that disrupt chain availabilty or perhaps even attracting a greater share of hash power to the attacking chain.

Model and Assumptions

We make many simplifying assumptions when modeling this system. The goal is to make enough assumptions to keep analysis tractable to begin while at the same time parameterizing the problem in a way that starts off at least somewhat plausible, and paves the way for more advanced modeling. The following are global system assumptions

- There are two chains in the system, chain 1 and chain 2. Both chains use the same proof of work and difficulty adjustment algorithm as the Bitcoin protocol
- Each chain rewards miners with a coinbase transaction that pays out tokens to the miner that solves the block puzzle first. We assume that the value of the coinbase reward of chain 1 is f_1 and the value of the token of chain 2 is f_2
- We model the total hashpower of the system as H (hashes / second) and assume that it does not change over the period of analysis in question.
- Fractions of the hashpower belong to three different entities, α, β_1, β_2. $\alpha + \beta_1 + \beta_2 = 1$.

- We assume a fraction α of the hash power is willing to switch between chains in order to make a profit and furthermore that α belongs to a single pool.

- β_1 and β_2 corresponding to miners who are loyal to chains 1 and 2 respectively. These hashpower fractions do not move from their respective chains regardless of the profitability of switching. For our purposes we can consider the hash power of β_i as belonging to a single pool.

- The difficulty of each chain is adjusted every b blocks. The time it takes for b blocks to be mined in epoch $i = \Delta_i$. To calculate the difficulty adjustment the protocol specifies that: $d_{i+1} = \frac{d_i \cdot b \cdot 600}{\Delta_i}$, as in bitcoin.

- We assume the system begins in a steady state where the difficulty of each chain reflects an initial allocation of the hash power. We define α_{10} and α_{20} to be the initial allocation of free moving hash power fixed on chains 1 and 2 respectively before miners become profit seeking.

Breaking down the problem

All mining strategies will make use of some foundational calculations. Our analysis must frame these calculations in terms of parameters of our model.

The fundamental calculation in this analysis is the switching decision criterion which determines some value to represent the profitability of chain 1 and chain 2 that guides decisions about allocating hash power to each chain among the switching miners. Intuitively this value should only depend on the relative value of the two coinbases (recall this is fixed in our model) and the relative difficulty of each chain. As we'll see below the relative difficulty of chains is determined by the hash power on the chains in the past.

In reality the relative difficulty of chains depends on the time it takes for a number of blocks to be mined which is governed not only by the hash power on the chain but also an element of randomness in the system. We will neglect to account for this randomness in our initial investigation of the greedy miner strategy to keep things simple. We should come back to this simplification and reintroduce randomness in our analysis of miner strategy profitability.

All switching miner strategies will make decisions to move hash power between chains. In general, strategies will consider switching hash power at the following **switching points**:

1. The starting point

2. Any point in time where at least one chain's difficulty adjusts

The key point here is that there is nothing to be gained from a miner switching hash power before the difficulty adjusts because the decision values for each chain will be the same. If we get around to including coinbase value changes into our model then this gets more complicated and strategies may need to consider switching hash power at each price change event. Additionally the miner might be able to gain information from variance in block mining times if we start considering the variance in miner strategies and this might trigger additional switching points as well.

Calculating the initial difficulty

At this point it is useful to calculate the initial difficulty in terms of model parameters. This way we can calculate the initial switching criterion, and from here the future switching times and future switching criteria. After making a few such calculations we can try to piece together the pattern of what is happening based on model parameters (either with reasoning or running simulations). The following discussion applies to both chains but let's focus on Chain 1 wlog.

Let's assume that the mining epoch before the beginning of our analysis took exactly 600 seconds on average (10 minutes) to find each block. This is a fair assumption given that our model assumes beginning in a steady state. The number of hashes tried until a block is successfully mined follows a geometric distribution and so the inverse of the probability of one success is equal to the mean number of hashes. Given our steady state condition we have the following relation

$$600H(\alpha_{01} + \beta_1) = \frac{1}{p}$$

Where p is the probability of one hash solving the block puzzle. Furthermore the difficulty of a chain is given by

$$\frac{F_{\text{max}}}{F}$$

where F is the threshold value below which a hash solves the puzzle. As the range of the hash function used in the chain puzzle is binary strings of 256 characters, $p = \frac{F}{2^{256}}$ and therefore

$$\begin{aligned} 600H(\alpha_{01} + \beta_1) &= \frac{2^{256}}{F} \\ 600H(\alpha_{01} + \beta_1)F_{\text{max}} &= \frac{2^{256}F_{\text{max}}}{F} \\ \frac{600H(\alpha_{01} + \beta_1)F_{\text{max}}}{2^{256}} &= d_{10} \end{aligned}$$

where d_{1i} is the difficulty of Chain 1 in the ith epoch. Factoring out constants we have

$$\begin{aligned} \mathcal{D} &= \frac{600HF_{\text{max}}}{2^{256}} \\ d_{10} &= \mathcal{D}(\alpha_{10} + \beta_1) \\ d_{20} &= \mathcal{D}(\alpha_{20} + \beta_1) \end{aligned}$$

The Greedy strategy

One reasonable question is what happens if the switching miner decides to maximize for profit with a simple greedy algorithm. At every switching point (as defined above) the miner looks at the immediate expected value of being on one chain or the other to chose which chain to mine on.

The miner must take time into account when making this decision. If the miner can expect a reward with value 10 from mining on Chain 1 and 5 from mining on Chain 2 it might seem like the miner should pick Chain 1. If Chain 1's expected reward is over a difficulty adjustment period that is 3 times longer than Chain 2's period however, this clearly is the wrong decision. For this reason we frame the greedy algorithm in terms of profit rates.