Mining Probability Density Function

idBrain*

January 11, 2022

Abstract

In this work we demonstrate how the binomial negative model for Bitcoin mining can be described as a Poisson process as function of time when the number of hashes tends to infinity and the mining target tends to zero. At the end we derive a simple equation connecting mining difficulty, hash power and the mean time between blocks and we also use the equation for an example with the actual numbers of the Bitcoin network at time of writing.

I. Introduction

In the Bitcoin white paper[1] Satoshi Nakamoto uses a Poisson process to model the mining mean time between blocks and based on that model to calculate the probability for a successful double spend attack. From those calculations the usual six blocks confirmation tradition was established considering less than 1% success probability with 10% enemy's power hashrate over the network. Although the blockchain progress under the attack is modeled using a binomial random walk, the mining probabilities are still calculated using Poisson distribution, which is a good approximation but does not translate the nature of the discrete process that is mining.

II. Model

The mining process has only two states which are success or failure. Those states are determined by the comparison between the value of the header's block hash and the target value set by the Bitcoin network, if the hash's value is lower than the target than the miner achieved success, otherwise it is a failure. These state probabilities can be labeled as:

$$Pr = \begin{cases} q \text{ (success)} \\ p \text{ (failure)} \end{cases} \tag{1}$$

Where the normalization condition imposes $q + p = 1$. So if we are looking for a sequence on n failures that stops at a success, we have the following probability:

$$P(n) \propto p^n = (1-q)^n \tag{2}$$

We normalize the relation to find the constant of proportionality

$$\begin{aligned} P(n) &= kp^n \\ \sum_{n=0}^{\infty} P(n) &= 1 \\ \sum_{n=0}^{\infty} kp^n &= 1 \\ k\sum_{n=0}^{\infty} p^n &= 1 \\ k\frac{1}{1-p} &= 1 \\ k &= (1-p) = q \\ P(n) &= qp^n \end{aligned} \tag{3}$$

Now we calculate the mean number of at-

*Character from "Irmãos Brain"

tempts (hashes) $\bar{n}$ till the miner hit a success:

$$\begin{aligned}\bar{n} &= \sum_{n=0}^{\infty} nP(n) = \sum_{n=0}^{\infty} nqp^n \\ &= \sum_{n=0}^{\infty} qp\frac{d}{dp}p^n \\ &= qp\frac{d}{dp}\sum_{n=0}^{\infty} p^n \\ &= qp\frac{d}{dp}\left(\frac{1}{1-p}\right) \\ &= qp\frac{1}{(1-p)^2} \\ \bar{n} &= \frac{p}{q} \end{aligned} \tag{4}$$

III. Poisson Distribution Approximation

To achieve the Poisson distribution we apply the limits $n \to \infty$ and $q \to 0$. These limits are in accordance with the actual characteristics of the Bitcoin network. We also extend the probability to a continuous process dependent on time t. Now we can relate all these numbers using a linear relation where the number of hashes n is equal to the product of mean number of hashes $\bar{n}$ and the ratio between the time past t and the mean time between blocks T_0

$$n = \bar{n}\frac{t}{T_0} \implies \bar{n} = n\frac{T_0}{t} \tag{5}$$

Substituting the relation (5) into (4)

$$\begin{aligned} n\frac{T_0}{t} &= \frac{p}{q} \\ n\frac{T_0}{t} &= \frac{1-q}{q} \\ n\frac{T_0}{t} &= \frac{1}{q} - 1 \\ \frac{1}{q} &= 1 + n\frac{T_0}{t} \\ q &= \frac{1}{1+n\frac{T_0}{t}} \end{aligned} \tag{6}$$

In this way we create a constraint where n going to infinity assures q going to zero. Now lets take the relation (6) and substitute into (3)

$$\begin{aligned} P(n) &= kp^n = k(1-q)^n \\ &= k\left(1 - \frac{1}{1+n\frac{T_0}{t}}\right)^n \\ &= k\left(\frac{1+n\frac{T_0}{t}-1}{1+n\frac{T_0}{t}}\right)^n \\ &= k\left(\frac{n\frac{T_0}{t}}{1+n\frac{T_0}{t}}\right)^n \\ &= k\left(\frac{1}{1+\frac{t}{nT_0}}\right)^n \\ P(n) &= k\frac{1}{(1+\frac{t}{nT_0})^n} \end{aligned} \tag{7}$$

Now we make the following variable substitution $\delta = \frac{nT_0}{t}$ and take the limit $n \to \infty \implies \delta \to \infty$ so the probability density becomes a function of time continuous variable t

$$\begin{aligned} \mathcal{P}(t) &= \lim_{\delta\to\infty} k\frac{1}{(1+\frac{1}{\delta})^{\delta\frac{t}{T_0}}} \\ \mathcal{P}(t) &= k\left(\frac{1}{\lim_{\delta\to\infty}(1+\frac{1}{\delta})^{\delta}}\right)^{t/T_0} \\ \mathcal{P}(t) &= k\left(\frac{1}{e}\right)^{t/T_0} \\ \mathcal{P}(t) &= ke^{-t/T_0} \end{aligned} \tag{8}$$

As expected we find the Poisson distribution. To conclude the model we normalize the distribution to find the new constant of proportionality k

$$\begin{aligned} 1 &= \int_0^{\infty} \mathcal{P}(t)dt \\ &= \int_0^{\infty} ke^{-t/T_0}dt \\ &= k\left[-T_0e^{-t/T_0}\right]_0^{\infty} \\ &= -kT_0\left[0-1\right] \\ 1 &= kT_0 \implies k = \frac{1}{T_0} \\ \mathcal{P}(t) &= \frac{e^{-t/T_0}}{T_0} \end{aligned} \tag{9}$$

Last step we calculate the mean time between blocks to check the consistency of the mathematical model

$$\begin{aligned}
\bar{t} &= \int_0^\infty t\mathcal{P}(t)dt \\
&= \int_0^\infty t\frac{e^{-t/T_0}}{T_0}dt \\
&= \frac{1}{T_0}\int_0^\infty te^{-t/T_0}dt \\
&= \frac{1}{T_0}\int_0^\infty \frac{-d}{d(1/T_0)}e^{-t/T_0}dt \\
&= \frac{-1}{T_0}\frac{d}{d(1/T_0)}\int_0^\infty e^{-t/T_0}dt \\
&= \frac{-1}{T_0}\frac{d}{d(1/T_0)}\left[-T_0e^{-t/T_0}\right]_0^\infty \\
&= \frac{-1}{T_0}\frac{d}{d(1/T_0)}(-T_0)\left[0-1\right] \\
&= \frac{-1}{T_0}\frac{dT_0}{d(1/T_0)} \\
\bar{t} &= \frac{-1}{T_0}(-T_0^2) = T_0
\end{aligned} \tag{10}$$

Exactly as expected obtain $\bar{t} = T_0$ as we defined T_0 as the mean time between blocks in the binomial model.

IV. Application

Now that we demonstrated that T_0 is the mean time between blocks we can relate it to the hashrate λ of the network

$$\bar{n} = \lambda T_0 \tag{11}$$

where the mean number of hashes equals the product of hashrate and mean time. Now we substitute that in (6)

$$q = \frac{1}{1+\lambda T_0} \tag{12}$$

To test (12) and see if it has any meaningful information we will isolate λ and try to calculate it using the current Bitcoin network target and Bitcoin mean time between blocks. The current target can be obtained in the *nBits* section in the header of a recent block. In the time of writing this number is

`nBits=170B98AB`

How this information can be translated to a numeric value can be found at [2]. The value of the target is

`0x00000000000000000000B98AB0000000000000000000000000000000000000`

which is equal to

$$11\times16^{44}+9\times16^{43}+8\times16^{42}+10\times16^{41}+11\times16^{40}$$

The probability of a success q is defined as the target value divided by the hole *SHA256* sample space that is 2^{256}, and as the Bitcoin mean time between blocks is ten minutes we can use these information to calculate the Bitcoin network hashrate

$$\begin{aligned}
\lambda &= \frac{1}{T_0}\left(\frac{1}{q}+1\right) \\
\lambda &= 1,73\times10^{20} = 173EH/s
\end{aligned}$$

Considering that in many blockchain explorer websites the hashrate is close to 175 EH/s this seems like a good result.

References

[1] Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System https://bitcoin.org/bitcoin.pdf.

[2] Difficulty calculation in Bitcoin. https://en.bitcoin.it/wiki/Difficulty.