Homework 8

Cyril Renevey, Lukas Pestalozzi, Joey Zenhäusern, Leonardo Wirz

November 24, 2017

Problem B

We know that the time distribution is an exponential distribution, which describes the time between events of a poisson process. The PDF is : $f(t) = \lambda e^{-\lambda t}$ where $t \in [0, \infty]$ and λ needs to be determined. The expectation of such a distribution is λ^{-1} and the variance is λ^{-2}. For the bitcoins, the mean is fixed to 10 minutes or 600 seconds, thus we can find λ for the distribution of the time to mine : $\lambda^{-1} = 600 \implies \lambda = 1/600\ \mathrm{s}^{-1}$. Thus the distribution of time it takes to successfully mine a block is the exponential distribution $exp(1/600)$, with a PDF :

$$f(t) = \frac{e^{\frac{-t}{600}}}{600}.$$

Furthermore, if a miner as a fraction $q \in [0, 1]$ of hash power, its mean time to find a block is reduced to 600/q seconds. This implies that its time distribution becomes $exp(q/600)$ which has mean $600/q$ seconds and standard deviation $\sqrt{variance} = \sqrt{600^2/q^2} = 600/q$ seconds.

Problem C

The hash of a new block is calculated from several fields, including the hash of the last block, a "Nonce" and the hash of the Merkel root which contains all transactions and, on his left-most leaf, an "extraNonce". The goal is to find a hash that starts with enough 0's to fulfill the current target difficulty. The Nonce starts at 0 and is incremented for each hash. Whenever Nonce overflows, which it does frequently because of the current difficulty, the extraNonce portion of the generation transaction is incremented, which changes the Merkle root. In summary, the nonce being only a 32-bit number, it is not big enough to generate enough different hashes. Therefore there is another extraNonce in the Merke tree that is used to generate more different hashes.

Problem D

In order to estimate the amount of bitcoins spent per block there are a few things to consider. A naive approach would be, to simply sum up all transaction inputs, this would included the fees and the bitcoins sent to the output addresses. But this does not account for the change, the part of the transaction that is sent back to the sender. So the actual spent amount would be the sum of the inputs of all transactions minus the change outputs. Because many wallets send the change back to a newly created address, it is hard to tell which output is the change output, and which is the output that is the actual payment to another entity.

An algorithm that estimates the amount of bitcoin actually spent per block would need to estimate the amount of change per block and subtract it from the total of the inputs of all transactions in that block. We could this this by assigning a probability to every output of it being a change output, and then summing those which have the highest probability.

Another fact that can help us identify change outputs is the number of inputs. If there is only one input, the above applies. But if there are multiple inputs, we can make assumptions about which outputs can not be change based on the fact that the amount of the payment output needs to be higher than any individual input amount, otherwise less inputs could have been used to make the transaction.