CS 765: Assignment 1
Simulation of a P2P Cryptocurrency Network

Manish Kumar 22M2110
Anubhav Jana 22M2109
Dhruv Jain 22M0828

Contents

1.Why does transaction interarrival time follow exponential distribution ?

Mathematically, it can be demonstrated that the time between transactions follows an exponential distribution. If we consider a time interval of a certain size, denoted as ∆, and assume transactions are equally likely to happen at any time within this interval, then the likelihood of a transaction occurring within this interval is proportional to ∆. Let's denote this probability as β . Now, if we assume the starting point at t = 0 to be a transaction, then the probability that the subsequent transaction occurs after n such time intervals can be expressed as :

$$\Pr[t > n\Delta] = (1 - \beta\Delta)^n$$

since it is equal to the probability that there is no transaction in each of n intervals of size ∆. Rewriting this with T = n∆, we have

$$\Pr[t > T] = \left(1 - \frac{\beta T}{n}\right)^n$$

To generalize the above probability to arbitrary and continuous values of T , suppose the intervals of size Δ are infinitesimally small, i.e., Δ → 0, and consequently n = (T/Δ)→ ∞, We get

$$\lim_{n\to\infty}\left(1-\frac{\beta T}{n}\right)^n = e^{-\beta T}$$

and thus Pr[t > T] = e−βT , which implies Pr[t ≤ T] = 1 − e−βT .
The latter is exactly the cumulative distribution function (CDF) of an exponential distribution having mean 1/β. Thus, we sample transaction interarrival time from an exponential distribution having mean Ttx = 1/β, for which the probability distribution function (PDF) is

$$p(t=T) = \frac{1}{T_{tx}} e^{-T/T_{tx}}.$$

1.2 Why is the mean dij inversely proportional to Cij ?

Note that "dij" represents the delay experienced at node i when transmitting a message to node j, while "cij" denotes the speed of the link between nodes i and j, measured in bits per second. The delay in the queue tends to increase when the link speed is low and decrease when it is high. To illustrate this, consider a packet, P, in a queue with n packets queued before it. A higher link speed towards the destination implies that more packets will be taken out of the queue per unit of time, resulting in a faster dequeuing process for packet P itself. Specifically, if each packet has a size of k bits and the link speed is c bits per second, the queueing delay for packet P can be calculated as:

$$d = \frac{\text{number of bits queued before } P}{\text{link speed}} = \frac{nk}{c}.$$

Thus, we have an inverse proportionality between the mean of queueing delay dij and link speed cij .

1.3 Explanation for the choice of Tk

The time it takes for a block to be mined, Tk, follows an exponential distribution with a mean of I/hk, where hk represents the proportion of hashing power owned by node k. The parameter we can adjust here is I. The goal is to minimize the occurrence of forks in the blockchain. In Satoshi's Bitcoin, network delays are typically around 10 seconds, and the average block mining time is maintained at around 10 minutes. This choice aims to ensure that a single successfully mined block can propagate to all nodes before multiple nodes succeed in mining, thus reducing the likelihood of forks. In our simulation, network latencies are around 500-1000 milliseconds. Therefore, for an optimal block mining time, we set I to 100,000 milliseconds.

2. Experiments and Insights

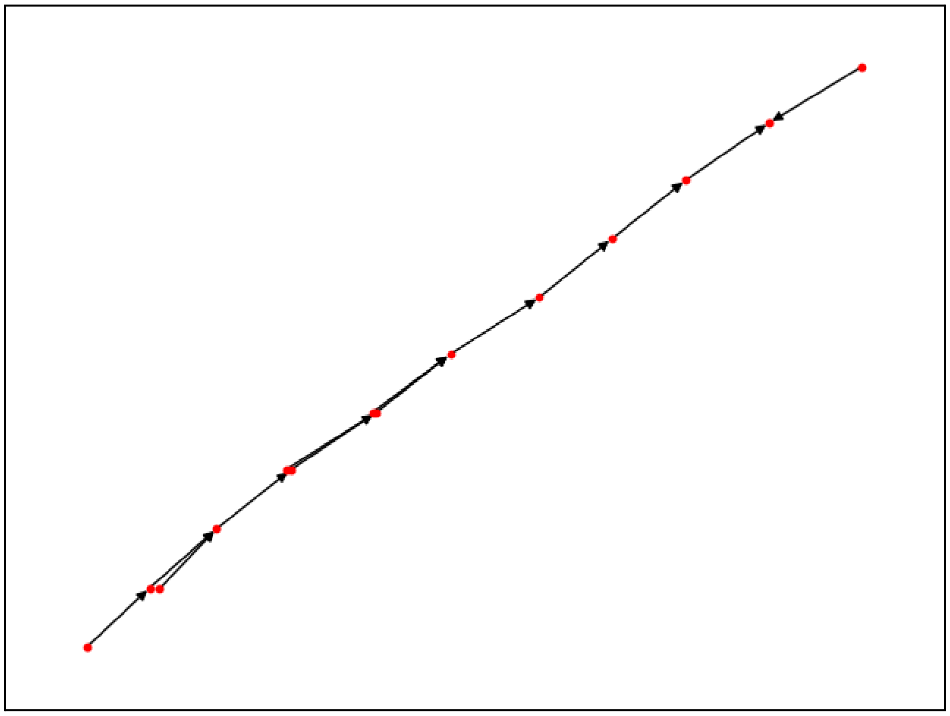
2.1 Experiment 1

For this experiment, we use the following parameters:

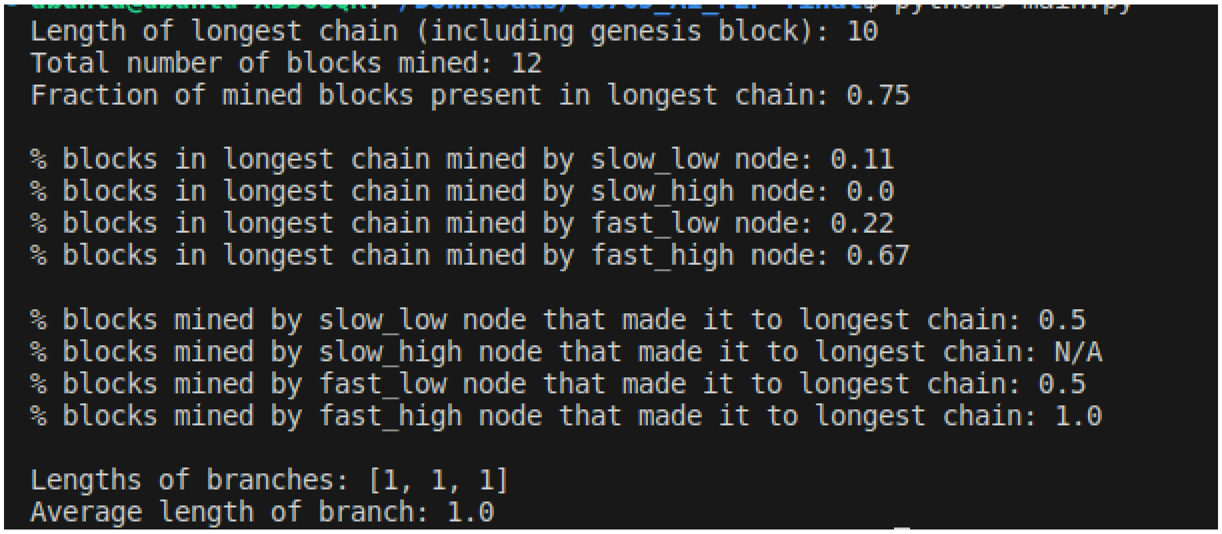
1. Number of nodes, n = 10
2. Percentage of slow nodes: z0 = 0.5
3. Percentage of nodes with low hashing power: z1 = 0.5
4. Mean transaction inter-arrival time: Ttx = 1000 ms
5. Average block mining time: I = 10000 ms
6. Total simulation time: T = 10000 ms

Command: python main.py -I 100000 -T 10000 -ttx 1000 -s

The following blockchain is generated.



The experiment yield the following results:



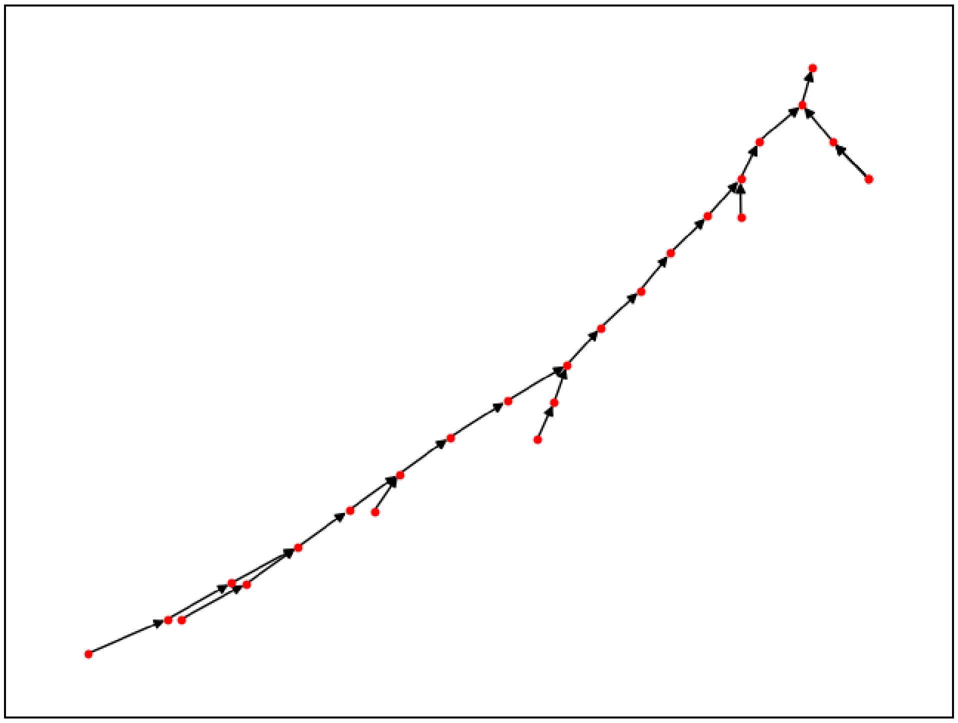
```
Length of longest chain (including genesis block): 10
Total number of blocks mined: 12
Fraction of mined blocks present in longest chain: 0.75

% blocks in longest chain mined by slow_low node: 0.11
% blocks in longest chain mined by slow_high node: 0.0
% blocks in longest chain mined by fast_low node: 0.22
% blocks in longest chain mined by fast_high node: 0.67

% blocks mined by slow_low node that made it to longest chain: 0.5
% blocks mined by slow_high node that made it to longest chain: N/A
% blocks mined by fast_low node that made it to longest chain: 0.5
% blocks mined by fast_high node that made it to longest chain: 1.0

Lengths of branches: [1, 1, 1]
Average length of branch: 1.0
```

As expected, these parameters form a single chain without any branches. The block mining time is clearly large enough to exceed the network latencies, which does not allow forking in the blockchain. Nodes are informed about a miner’s success before they are done mining, so it is highly probable that a single successfully mined block is further mined upon by all nodes in the network.

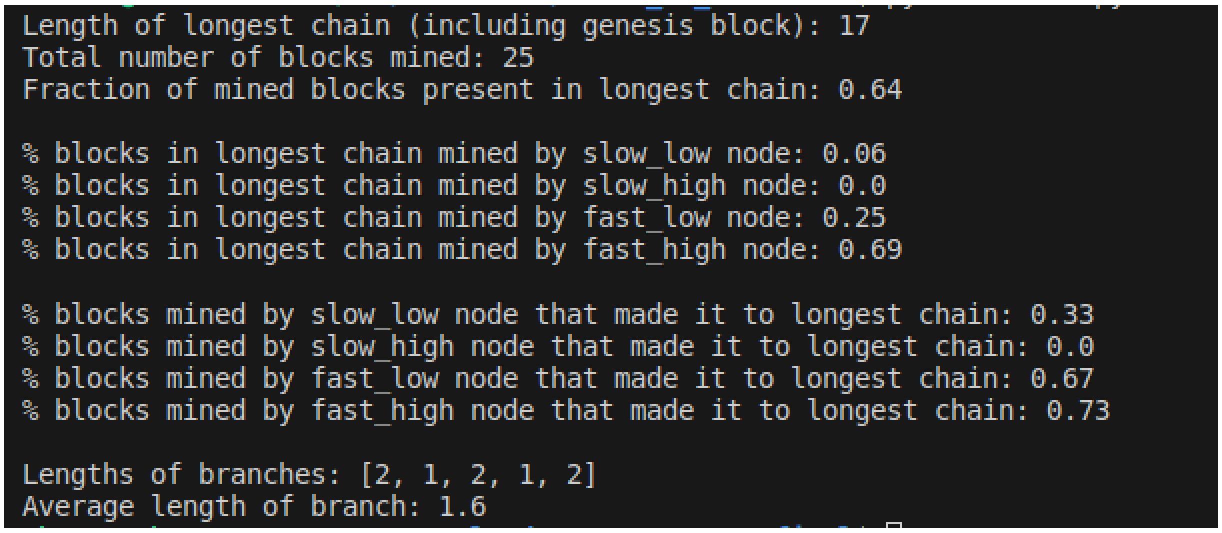
2.2 Experiment 2

We keep the same parameters as Experiment 1, except we reduce I to 1000 ms and T to 1000 ms. The block mining time is now closer to the network latency values.

The blockchain obtained is as follows



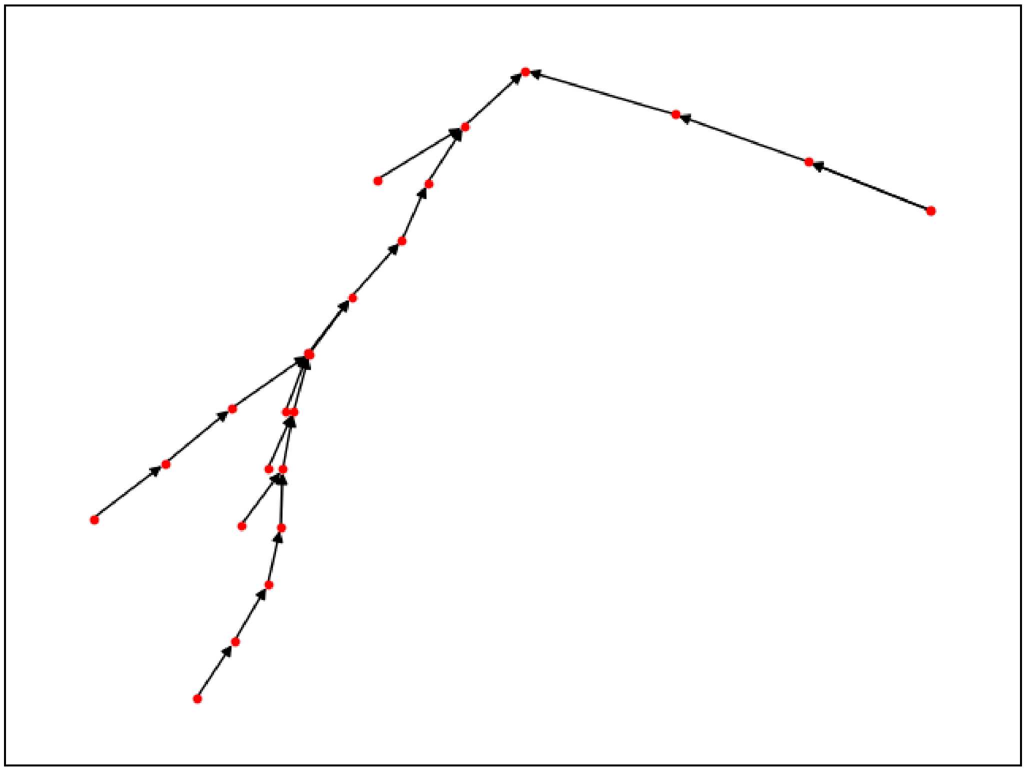
The experiment yielded the following results:



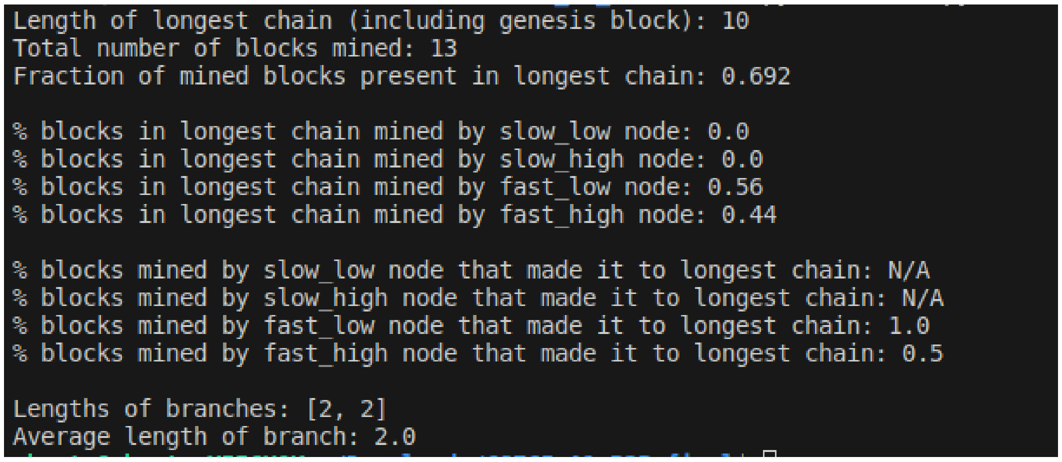
```
Length of longest chain (including genesis block): 17
Total number of blocks mined: 25
Fraction of mined blocks present in longest chain: 0.64

% blocks in longest chain mined by slow_low node: 0.06
% blocks in longest chain mined by slow_high node: 0.0
% blocks in longest chain mined by fast_low node: 0.25
% blocks in longest chain mined by fast_high node: 0.69

% blocks mined by slow_low node that made it to longest chain: 0.33
% blocks mined by slow_high node that made it to longest chain: 0.0
% blocks mined by fast_low node that made it to longest chain: 0.67
% blocks mined by fast_high node that made it to longest chain: 0.73

Lengths of branches: [2, 1, 2, 1, 2]
Average length of branch: 1.6
```

2.3 Experiment 3

We keep the same parameters as Experiment 1, except we reduce now I to 100 ms and T to 1000 ms, along
with Ttx to 10 ms and number of nodes = 20. The block mining time is now likely to be even smaller than the network latencies, which are of the order ~ 500 ms.



The experiment yielded the following results



```
Length of longest chain (including genesis block): 10
Total number of blocks mined: 13
Fraction of mined blocks present in longest chain: 0.692

% blocks in longest chain mined by slow_low node: 0.0
% blocks in longest chain mined by slow_high node: 0.0
% blocks in longest chain mined by fast_low node: 0.56
% blocks in longest chain mined by fast_high node: 0.44

% blocks mined by slow_low node that made it to longest chain: N/A
% blocks mined by slow_high node that made it to longest chain: N/A
% blocks mined by fast_low node that made it to longest chain: 1.0
% blocks mined by fast_high node that made it to longest chain: 0.5

Lengths of branches: [2, 2]
Average length of branch: 2.0
```

On doubling the number of nodes in our network, we observe that there is now a lot more forking in the blockchain than there was before. The number of branches, as can be seen in the output screen, is much larger now.