Simulation of a P2P Cryptocurrency Network

CS 765 Assignment 1

Neel Aryan Gupta	Saurav Garg	Tathagat Verma
180050067	180050093	180050111

1. **Theoretical reasons for choosing the exponential distribution for transaction interarrival time [2]**

   We present the following mathematical argument:
   Let δ be a very small time interval such that $\delta \rightarrow 0$
   $P(txn\ is\ generated\ in\ the\ next\ \delta\ interval)\ is\ proportional\ to\ \delta$
   $P(txn\ is\ generated\ in\ the\ next\ \delta\ interval)\ = T_{tx}\ \delta$
   $P(txn\ is\ generated\ after\ n\delta\ time)\ = P(txn\ not\ generated\ in\ first\ n\delta\ time)\ = (1\ -\ T_{tx}\delta)^{n}$
   $Substituting\ t\ = n\delta\ and\ denoting\ transaction\ generation\ time\ as\ Random\ Variable\ t_{tx}$
   $,P(t_{tx} > t)\ =\ (1 - \frac{T_{tx}t}{n})^{n}$
   $Since\ \delta \rightarrow 0,\ for\ \ t\ \ to\ be\ finite,\ we\ have\ n \rightarrow \infty,\ therefore,$
   $P(t_{tx} > t) \sim\ e^{-T_{tx}t}\ \ \Rightarrow\ P(t_{tx}\ <\ t)\ =\ 1 - e^{-T_{tx}t}$
   Note that this is the CDF of Exponential distribution, therefore
   $P(t_{tx} = t) = T_{tx}e^{-T_{tx}t}$
   Therefore, we choose transaction interarrival time to be sampled from an exponential distribution.

2. **Random sampling for the peer network [4]**

   Research Paper
   We implemented a scale free network as suggested in the given research paper. We used the Barabási–Albert model to generate a random scale-free network using the preferential attachment mechanism. For every new node, we choose the neighbour *i* in the following way: the probability p_i that the new node is connected to node *i* is
   $p_i\ =\ \frac{k_i}{\sum_j k_j}$

where k_i is the degree of node i and the sum is made over all nodes j . Nodes have a preference to attach themselves to already heavily linked nodes. The degree distribution is roughly given by

$P(k) \sim k^{-\gamma} \quad where \quad \gamma \in (2, 3)$

The above follows from the fact that the log-log scale relationship between the fraction of k-degree nodes vs the degree k itself is linear with a negative slope $-\ \gamma$.

3. **Why is the mean of d_{ij} inversely related to c_{ij} ? Give justification for this choice. [2]**

   The queuing delay depends on the rate at which traffic is pushed out of the queue. For example, if there are 10 packets in the queue, then the 10th packet will have lesser queuing delay if the link rate c_{ij} is high as the first 9 packets can be pushed out at a fast rate. So, if the link rate is high, then more traffic can be pushed out per unit time, resulting in less delay for the other packets waiting at the queue, which in turn reduces the average queuing delay d_{ij} . Hence they are inversely proportional.

4. **Justify the chosen mean value for Tk.**
   We have chosen the default value of Tk as 1000 secs, which is approx 15 mins. Justification: Once a new block A is mined, the latency for transmitting it is about half a second for each link between any two peers. So, it'll take a couple of seconds to send this block to all peers in the network. We ideally want to reduce the number of forks in the blockchain. Fork would occur if some other node mines a new block before block A reaches that node. Hence, we choose the default value of Tk such that it is significantly higher than the transmission time. This reduces the probability of forking and we get almost a single chain without any forks. The value of Tk also determines the number of blocks generated over a time period. So, if Tk is very small, the number of transactions in each block would be less.

5. **Visualization and analysis**

   We provide the command to run each of them. We follow the key given below in experiments:

```
-n --peers                    Number of peers in the network
-e --edges                    Number of edges in the peer network
```

```
-z --slowpeers              Fraction of slow peers in the network
-t --time_limit             Run the simulation upto a time limit
-Ttx --txn_interarrival       Mean of exp distribution of interarrival time b/w
txns
-Tk --mining_time             Mean of exponential distribution of time to mine a
block
-s --seed                   Seed for random number generator
-txn --max_txns             Run simulation till max transactions are generated
-blk --max_blocks           Run simulation till max blocks are generated
-it --invalid_txn_prob      Probability of generating an invalid transaction
-ib --invalid_block_prob    Probability of generating an invalid block
```

There are two kinds of hash power, either low or high. The high hash power is set to be 5 times the low hash power. So, there are 4 categories of peers. These are based on the hash power (either Low or High) and the network speed (either Fast or Slow) of a peer. We analyse the fraction of blocks in the final blockchain of the peers corresponding to these four categories.

We report the *mean* of the *fraction of blocks in the blockchain to blocks generated by a peer* over the peers of each category in every simulation.

Simulation 1 :

```
./blockchain_simulator -n 40 -e 150 -z 0.4 -Ttx 10 -Tk 1000 -s 26 -blk 100
```

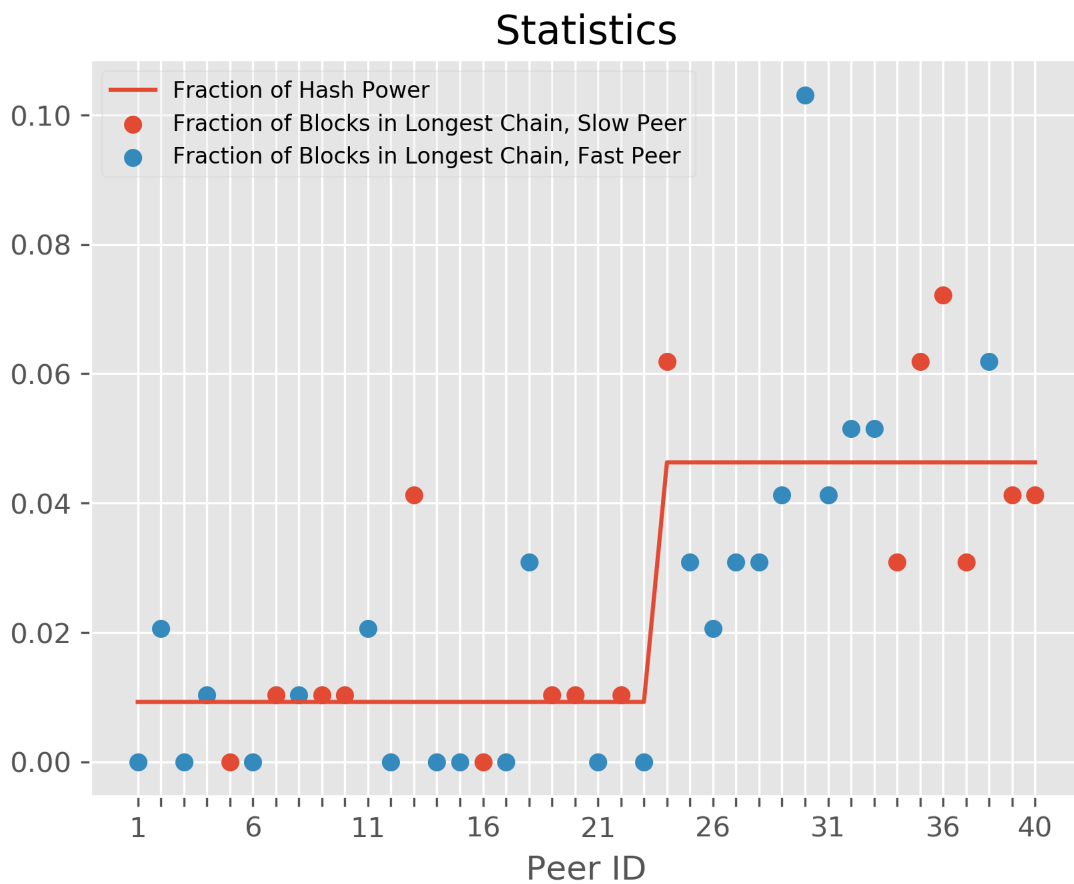
This simulation takes into account the typical Ttx and Tk values. Thus Tk is significantly higher than Ttx. We have kept Ttx as 10 sec, which is the general average transaction interarrival time across a network as discussed in lectures. And Tk is taken as 800 (approx. 10 min), again a general scenario.

Results:

```
Total blocks in Blockchain: 98
Longest chain length: 98
Longest chain length / total number of blocks: 1.000
No branches created

Hash power: LOW, Fast Node? False, Mean fraction of blocks: 0.778
Hash power: LOW, Fast Node? True, Mean fraction of blocks: 0.357
Hash power: HIGH, Fast Node? False, Mean fraction of blocks: 1.000
Hash power: HIGH, Fast Node? True, Mean fraction of blocks: 1.000
```

The ratio of longest chain length to total number of blocks is also high. The reason is that the Tk value is high. Since Tk value is approx 15 min which is much higher than the time taken by a mined block to propagate in the complete network, we see that there are no forks. This is illustrated in the blockchain tree plot below as well.
Blockchain Tree:

Statistics over peers:



We further plot for each peer, the ratio of blocks generated by it that are in the longest chain to the total number of blocks in the longest chain. Ideally, this fraction should be determined by the mining power (network propagation has lower effect in the usual scenario). This is validated by the plot in which we observe that network delay (specified by Slow and Fast peer) has a lower contribution to the ratio of blocks in the longest chain than the hash power.

Simulation 2:

```
./blockchain_simulator.exe -n 40 -e 150 -z 0.4 -Ttx 5 -Tk 5 -s 32 -blk 100
```

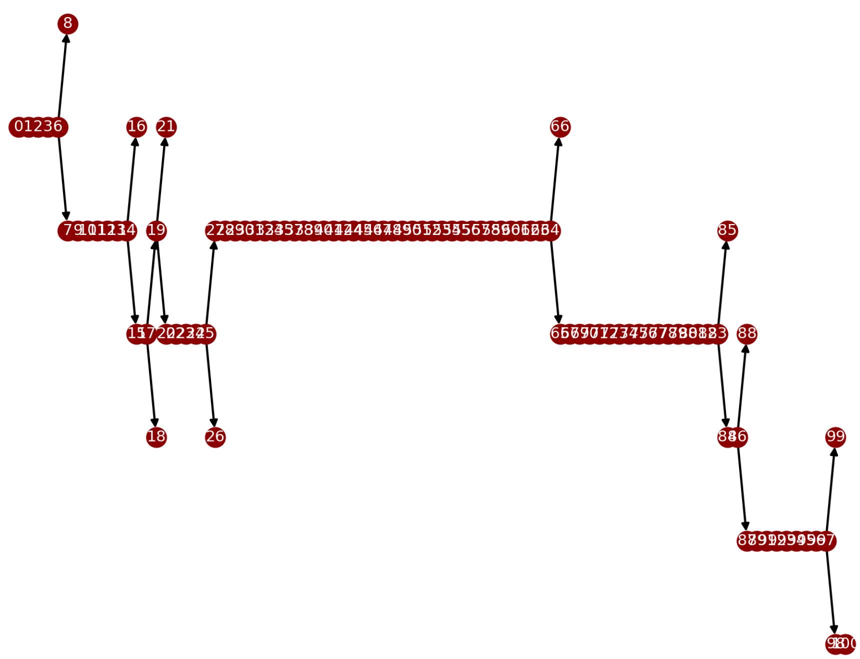
Here, we bring Tk down equal to Ttx and the total number of blocks generated to 100 (only changes from simulation 1). Since the average mining time (Tk) has reduced, we expect a high number of forks as more new blocks are mined before a previously mined block is seen by all nodes.

Results:

```
Total blocks in Blockchain: 94
Longest chain length: 85
Longest chain length / total number of blocks: 0.894
Branch Lengths: Total=9, Max=1, Mean=1.000, Min=1

Hash power: LOW, Fast Node? False, Mean fraction of blocks: 0.286
Hash power: LOW, Fast Node? True, Mean fraction of blocks: 0.250
Hash power: HIGH, Fast Node? False, Mean fraction of blocks: 0.963
Hash power: HIGH, Fast Node? True, Mean fraction of blocks: 0.817
```

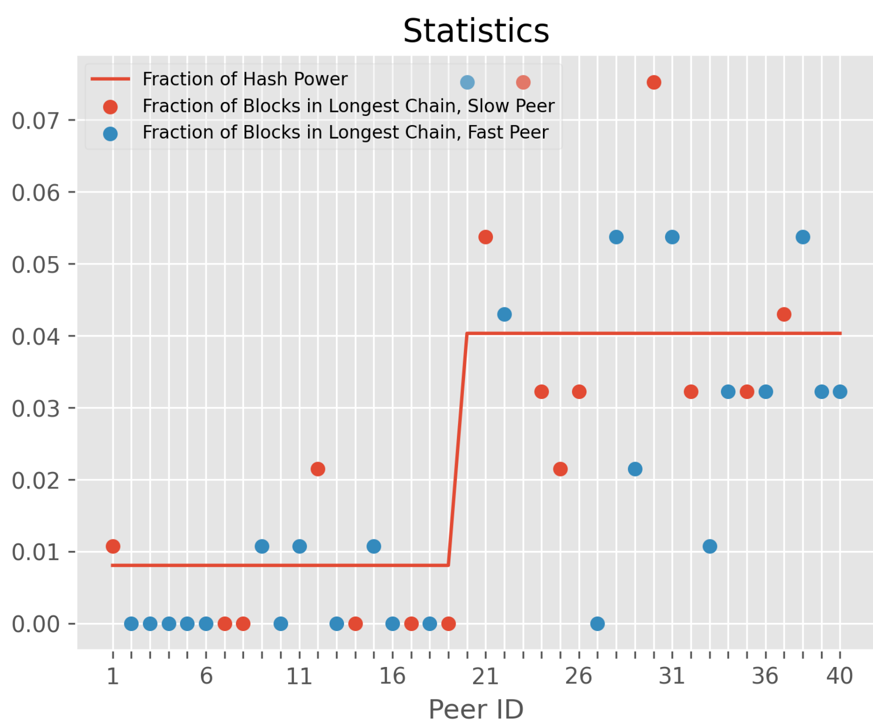
The results validate our expectation of a high number of forks and thus the rejection of a higher number of blocks. This causes the ratio of length of the longest chain to total number of blocks in the blockchain to fall down to 0.894 from 0.98 (simulation 1).
We also observe reduced fractions of blocks taken into the longest chain to the blocks generated for each of the 4 categories. The reason for this is that now more number of forks are occurring and hence more blocks are not getting included in the longest chain. Thus the lower fractions.

Blockchain Tree (notice the higher amount forking):



In peer level statistics, we notice that the fraction of blocks in the longest chain has a higher variance and is less dependent on the hash power. The reason is that now Tk is small thus the effect of difference in hash power is lowered.

Peer statistics:



Simulation 3:

```
./blockchain_simulator.exe -n 40 -e 150 -z 0.4 -Ttx 0.005 -Tk 0.01 -s 32 -blk 100
```

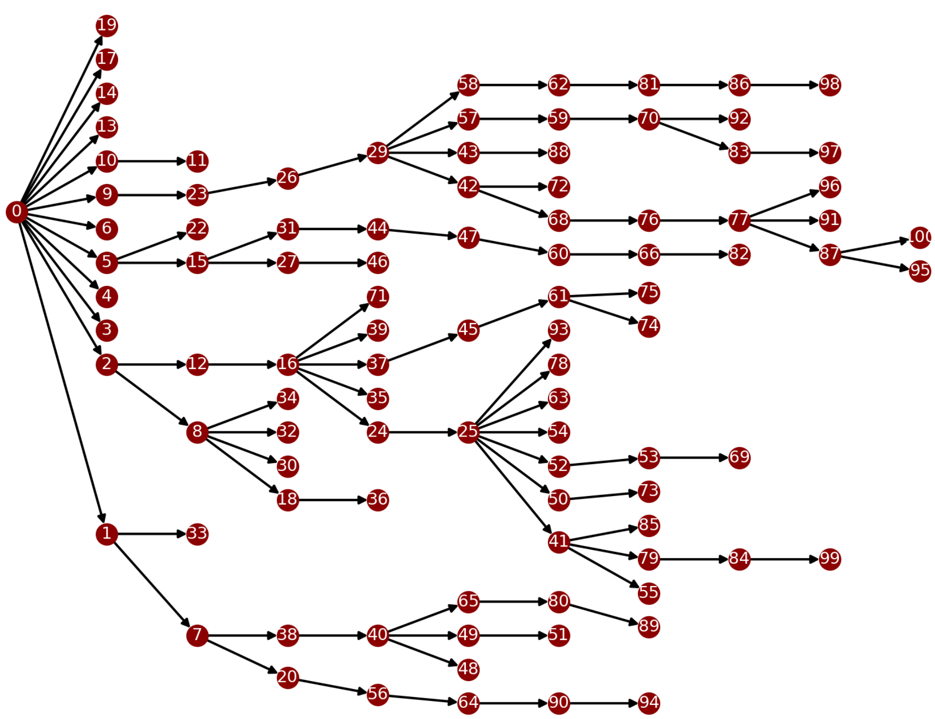
In this simulation, we reduce Ttx and Tk further. The significance of this experiment is that now Ttx and Tk are lower than the Link speeds of the network.

Results:

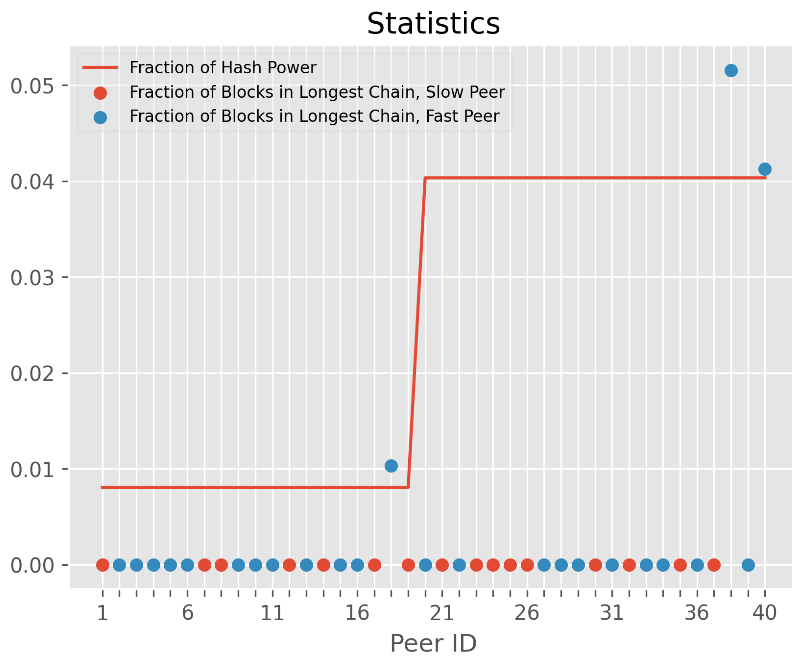
```
Total blocks in Blockchain: 98
Longest chain length: 11
Longest chain length / total number of blocks: 0.102
Branch Lengths: Total=18, Max=9, Mean=2.722, Min=1

Hash power: LOW, Fast Node? False, Mean fraction of blocks: 0.000
Hash power: LOW, Fast Node? True, Mean fraction of blocks: 0.083
Hash power: HIGH, Fast Node? False, Mean fraction of blocks: 0.000
Hash power: HIGH, Fast Node? True, Mean fraction of blocks: 0.125
```

Since the mining and transaction creation rate is higher than the network link speeds, we get an increased amount of forking as the network is now flooded with blocks. Thus the longest chain to total blocks ratio further reduces.

Blockchain Tree (much higher forking):



Peer statistics:



At the peer level, very few blocks get into the longest chain and many of the blocks get rejected. Thus only a few users have their blocks in the network and most other nodes have 0 blocks in the network.

Simulation 4:

```
time ./blockchain_simulator.exe -n 20 -e 80 -z 0.4 -Ttx 10 -Tk 1000 -s 26
```

In this simulation, we reduce the number of peers (and edges) and keep the other parameters the same as that of simulation 1.

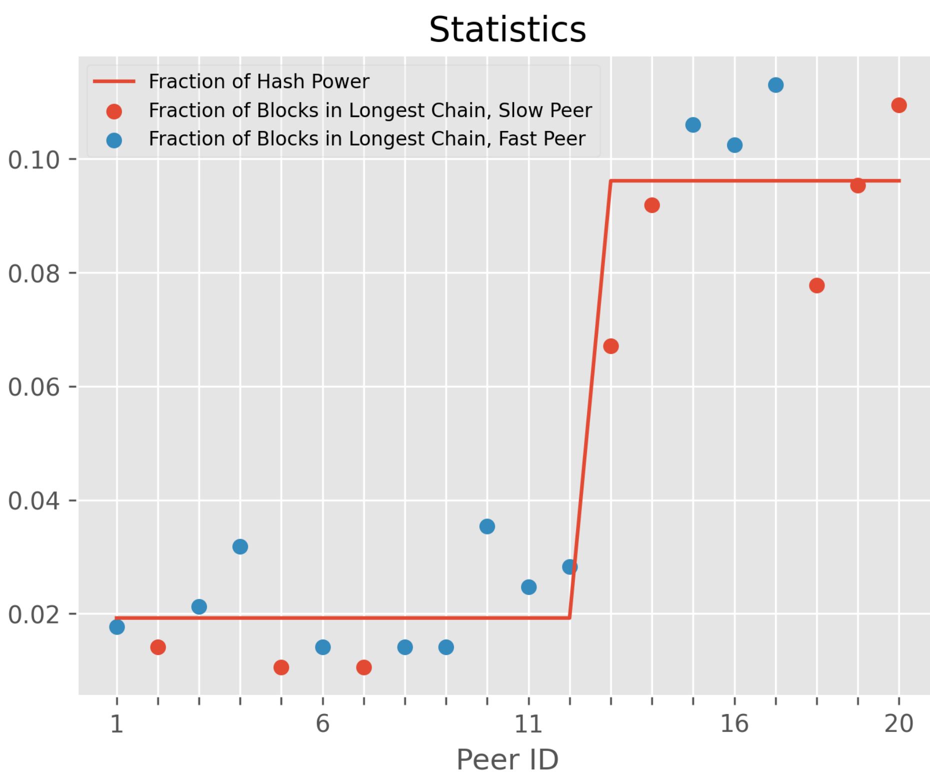
Results:

```
Total blocks in Blockchain: 284
Longest chain length: 284
Longest chain length / total number of blocks: 1.000
No branches created

Hash power: LOW, Fast Node? False, Mean fraction of blocks: 1.000
Hash power: LOW, Fast Node? True, Mean fraction of blocks: 1.000
Hash power: HIGH, Fast Node? False, Mean fraction of blocks: 1.000
Hash power: HIGH, Fast Node? True, Mean fraction of blocks: 1.000
```

We observe that there has been no forking and all blocks generated by a node are accepted in the longest chain. This behaviour is justified as the number of peers in the network is less thus reducing the competition for block mining. Due to this reduction in the number of blocks being generated overall, there is reduced forking.

Blockchain Tree:

Peer statistics:



Peer statistics trend stays similar to simulation 1, not being affected significantly by the number of peers. The fraction of longest chain blocks created by a peer are determined by the relative hash power and the relative network speed.