Homework 1

Simulation of a P2P Cryptocurrency Network

Chaitanya Garg (210050039), Omm Agrawal (210050110), Pulkit Goyal (210050126)

8th February 2025

1 Running Instructions

```
$ python3 main.py --help
usage: main.py [-h] -n NUM_PEERS -w Z0 -c Z1 -t TRANSACTION_INTERARRIVAL
-b BLOCK_INTERARRIVAL -s SIM_TIME [-f FOLDER]

Process CLI Inputs.

optional arguments:
  -h, --help            show this help message and exit
  -n NUM_PEERS, --num_peers NUM_PEERS
                        Number of Peers
  -w Z0, --z0 Z0        Fraction of slow peers (0 to 1)
  -c Z1, --z1 Z1        Fraction of low CPU peers (0 to 1)
  -t TRANSACTION_INTERARRIVAL, --transaction_interarrival TRANSACTION_INTERARRIVAL
                        Mean Interarrival Time for Transaction Generation (seconds)
  -b BLOCK_INTERARRIVAL, --block_interarrival BLOCK_INTERARRIVAL
                        Mean Interarrival Time of Blocks (seconds)
  -s SIM_TIME, --sim_time SIM_TIME
                        Simulation Time (seconds)
  -f FOLDER, --folder FOLDER
                        Folder to store results
```

All arguments mentioned in the Usage Instructions are required, except `-f, -folder` flag. The default folder name is created using all the other parameters.

2 Distribution for inter-arrival time sampling of Transactions

The theoretical reason for choosing exponential distribution for inter-arrival time sampling is the **Memoryless Property** it follows. This ensures that the probability of transaction generation doesn't change if a transaction is not generated for a while.

In this way, implements the fact that the transactions are generated independently of the previous transaction

3 Relation between Queuing delay and Link speed

Mean of Queuing delay and Link speed are inversely related. This is because when link speed increases, packets are removed faster from packet queues during transmission and receiving. Hence the probability

of crowding at that peer decreases on increase of link speed.
So, the mean queuing delay decreases but still due to random crowding at any point of time, random sampling is done

4 Choice of Mean Inter-arrival time of blocks

We take the idea from Bitcoin that the inter-arrival time of blocks should be considerably larger than network delay in order to avoid forks, while at the same time we need to simulate the creation of 100s of blocks in reasonable time. The network delay is around 1 second on an average. So, enforcing the above conditions, we decide mean inter-arrival time of blocks to be 10s

5 Variation of the Mean Block Inter-arrival time

5.1 Experiments

The following parameters were kept constant throughout :

Number of peers:	50
Fraction CPU_LOW:	40%
Fraction NETWORK_SLOW:	40%
Interarrival Time of Transactions:	0.2 seconds
Number of blocks mined:	100

The following block chain trees were observed for various values of Mean Block Inter-arrival time

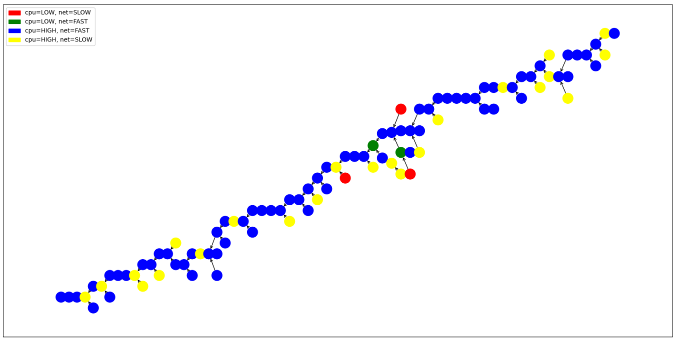

Figure 1: Block Interarrival Time: 1 second

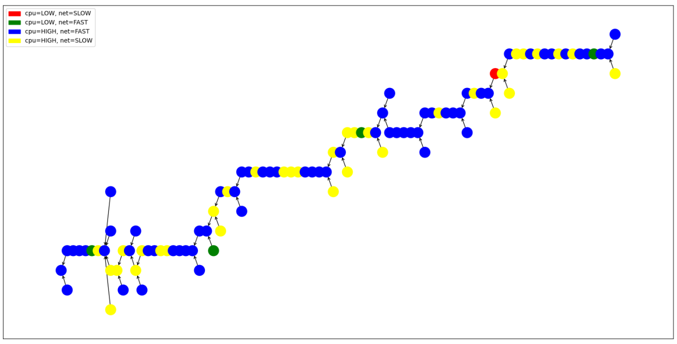

Figure 2: Block Interarrival Time: 5 second

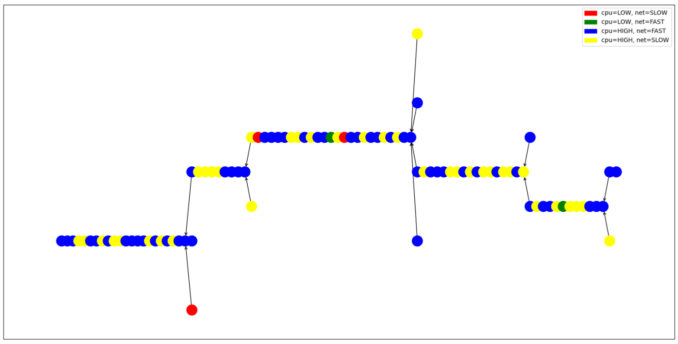

Figure 3: Block Interarrival Time: 10 second

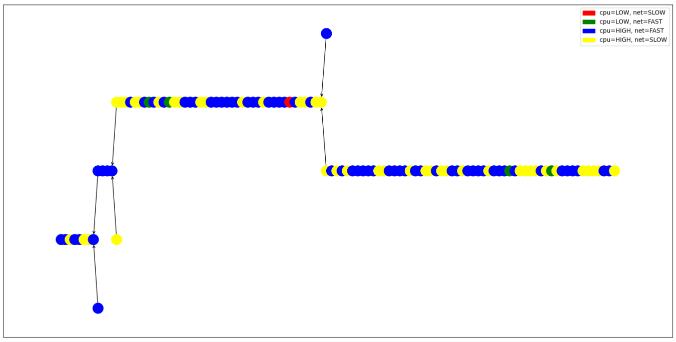

Figure 4: Block Interarrival Time: 20 second

The following branching parameters were correspondingly calculated :

Extent of branching

Mean inter-arrival time	*Number of Forks*	*Branching ratio*
1	*31*	*0.37*
5	*20*	*0.21*
10	*8*	*0.10*
20	*3*	*0.04*

$$\text{Branching ratio} = \frac{\text{Number of blocks not in longest chain}}{\text{Total number of blocks in tree}}$$

5.2 Insights

It is easily observed from the above Block tree figures as well as the objective measure of branching ratio as defined above, that the extent of branching reduces as the Mean inter-arrival time of block increases.

It can be reasoned as follows. The network delay is of the order of 1-2 seconds, and so when the inter-arrival time of blocks is comparable to it (1 seconds), the blocks are generated quite fast as compared to the time it takes to spread the block info across the P2P network. Hence at any peer, due to delay of arrival of block info, it starts to mine on an older block, causing forks

As this inter-arrival time increases, the rate of block generation is quite small as compared to the network delay, hence block generation information spreads a lot faster relatively, and so the miners do not work to mine on an older block, leading to less forks

6 Variation of Low CPU and Slow Network Peer Percentage

6.1 Experiments

The following parameters were kept constant throughout :

Number of peers:	50
Inter-arrival Time of Transactions:	0.2 seconds
Inter-arrival Time of Blocks :	5 seconds
Number of blocks mined:	100

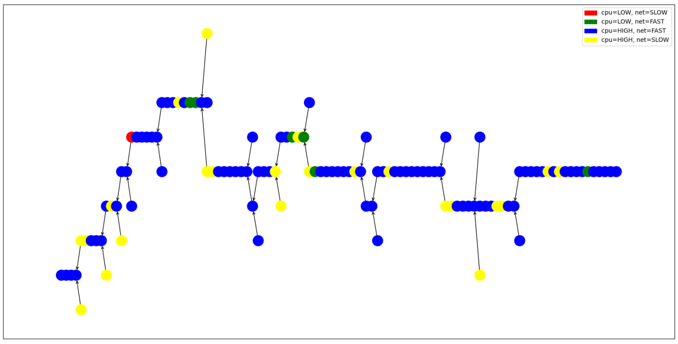

Figure 5: CPU LOW : 0.4 NET SLOW : 0.2

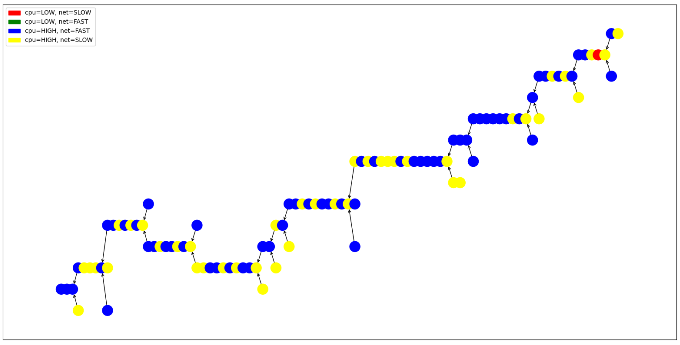

Figure 6: CPU LOW : 0.2 NET SLOW : 0.4

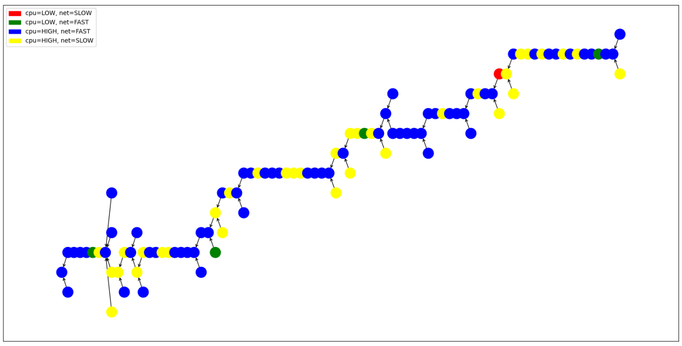

Figure 7: CPU LOW : 0.4 NET SLOW : 0.4

The following branching parameters were correspondingly calculated :

Extent of branching

CPU_LOW Ratio	NET_SLOW Ratio	Branching ratio
0.2	0.4	0.17
0.4	0.2	0.15
0.4	0.4	0.18

6.2 Insights

We can observe that the extent of forking/branching doesn't change considerably on varying the percentage of slow network nodes or low cpu nodes.
Hence the branching is strongly correlated with mostly the inter-arrival block time.

7 Contribution of Node type in the longest chain and Block tree

7.1 Experiment

The following parameters were kept constant throughout :

Number of peers:	50
Fraction of CPU_LOW:	40%
Fraction of NETWORK_SLOW:	40%
Inter-arrival Time of Transactions:	0.2 seconds
Inter-arrival Time of Blocks :	10 seconds
Number of blocks mined:	500

7.2 Observation

The following blockchain tree was observed for the above experiment

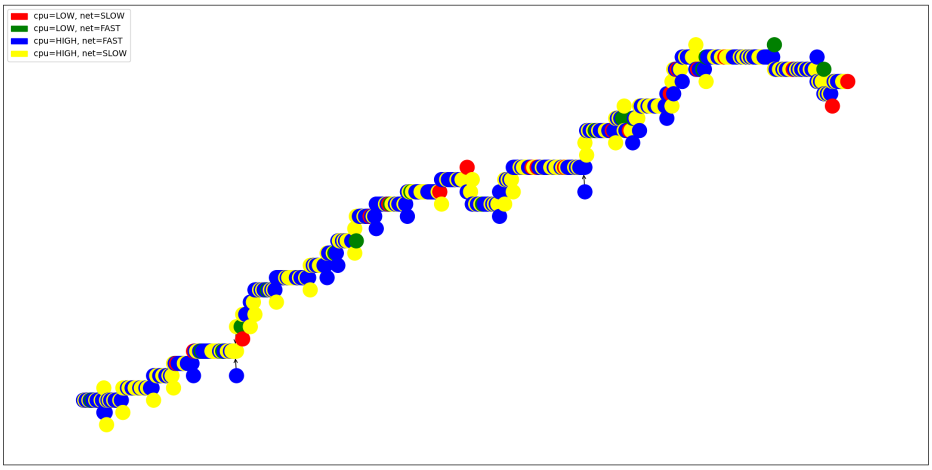

Figure 8: Corresponding Blockchain Tree

7.3 Insights

We calculate the following for each Node type (CPU type and Network type in combination gives the Node type)

$$\textbf{Contribution in longest chain per node} = \frac{\textbf{Blocks generated by nodes of this type}}{\textbf{Longest chain length x Number of nodes of this type}}$$

$$\textbf{Contribution in block tree per node} = \frac{\textbf{Blocks generated by nodes of this type}}{\textbf{Nodes in Block tree x Number of nodes of this type}}$$

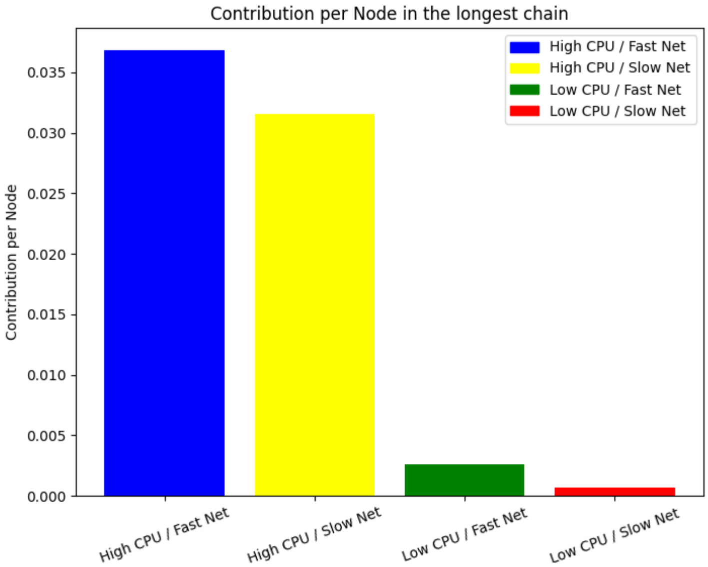

Figure 9: Per Node Contribution in the longest chain for the different types of nodes.

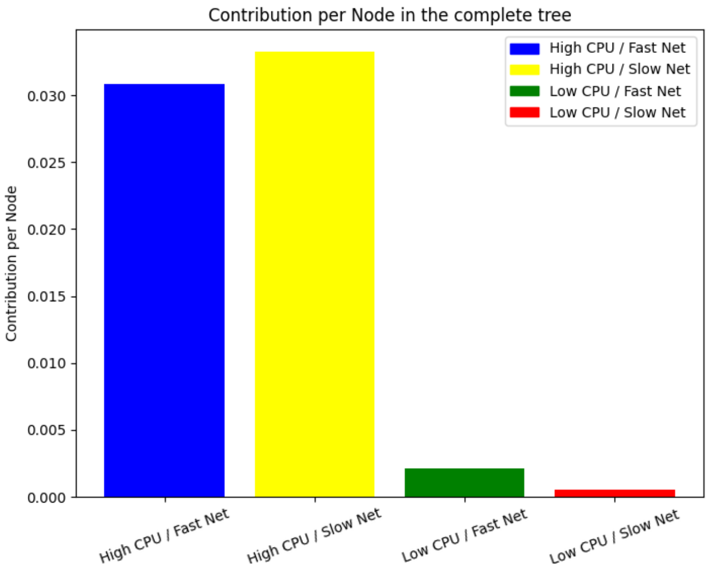

Figure 10: Per Node Contribution in Block tree for the different types of nodes.

We derive the following insights and critiques from the above data :

- High cpu nodes have significantly higher contribution as compared to low cpu nodes as expected. Due to higher hashing power, the time taken to generate a new block is less with much higher probability.

- Fast or slow network type of node has significantly less impact. This is because the overall network delay guides the contribution in the longest chain, rather than network type of a single node.
- Comparing fast and slow network types for high CPU nodes, we can see that although the slow ones have a bigger contribution in total block tree, it decreases in the longest chain contribution. This is because, the probability of network delay increases for them, and the information of where they are adding the blocks takes time to reach others. So other nodes can work on some other blocks in the mean time, changing the longest chain