CS 765: Simulation of a P2P Cryptocurrency Network

Advait, Ashwin, Harshvardhan
200100014, 200050016, 200050050

February 2023

1 What are the theoretical reasons for choosing the exponential distribution?

The generation of blocks is probabilistic since nobody can predict which nonce value is going to result in solving the hash puzzle. The only way to do it is by trying different nonces one after the other and hoping that one of them succeeds. Mathematically, this process can be modeled using Bernoulli trials. Assuming that the hash function behaves like a random function, the probability of the hash falling in the target is fixed. Let's assume that this probability of mining a (new) block in a finite time interval Δ is given by $\beta\Delta$, where β is some constant proportional to the miner's computation power.

Let I denote the inter-arrival time between 2 blocks and let us define $x = n\Delta$.

$$\Pr[\mathtt{I} = n\Delta] = (1 - \beta\Delta)^{(n-1)}\beta\Delta$$

This is because the second block will be generated at the n^{th} Bernoulli trial if it fails in the first (n - 1) trials and succeeds in the last trial. Now, the second block will be generated after n Bernoulli trials if it fails in all the first n trials. The probability for this event is given by :-

$$\Pr[\mathtt{I} > n\Delta] = (1 - \beta\Delta)^n$$

$$\Pr[\mathtt{I} > x] = \left(1 - \frac{\beta x}{n}\right)^n$$

$$\lim_{n\to\infty} \Pr[\mathtt{I} > x] = \lim_{n\to\infty} \left(1 - \frac{\beta x}{n}\right)^n = e^{-\beta x}$$

This is because as n tends to infinity, Δ tends to 0 thus allowing us to simulate continuous time intervals)

Moreover, the probability that a transaction will be made in the future will not depend upon the transactions that have been made till now. Therefore, the distribution over inter-arrival time for transactions generated by a peer must be memoryless. Since exponential distribution possesses the memoryless property, we choose this distribution to model the interarrival time of transactions.

These are some of the theoretical reasons for choosing exponential distribution.

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

d_{ij} is the queuing delay at node `i` to forward the message to node `j` whereas c_{ij} is the link speed between node `i` and `j`. Now, if the link speed is high, then packets will queue at the end resulting in a higher queuing delay. Therefore, mean of d_{ij} inversely related to c_{ij}.

3 Analysis of Blockchain Tree with different parameters

3.1 Varying Number of Peers (n)

For this analysis, we will take the z_0 and z_1 to be 20%, the interarrival time between transactions (T_{tx}) to be 1000 seconds and the time for Block Generation to be 600 seconds just as it is in Bitcoin.

- For n=10, the length of the longest chain is 112.
- For n=20, the length of the longest chain is 62.
- For n=50, the length of the longest chain is 27.

As can be seen, the length of the longest chain decreases as the number of peers in the network increases. Moreover, there is almost no branching for less number of peers while there is significant branching in the blockchain tree for higher number of peers.

3.2 Varying percentage of slow users in the network (z_0)

For this analysis, we will take the number of peers in the network to be 20, z_1 to be 20%, the interarrival time between transactions (T_{tx}) to be 1000 seconds and the time for Block Generation to be 600 seconds.

- For z_0=0%, the length of the longest chain is 63 and there are no branches in the blockchain tree
- For z_0=100%, the length of the longest chain is 51 and the number of branches in the blockchain tree is 3

As can be seen, there is less branching for less number of slow nodes while there is significant branching in the blockchain tree for higher number of slow nodes. A possible reason for this observation could be that the blocks broadcasted by the slow nodes take more time to reach all nodes in the network and in the meantime some other node may have broadcasted its block leading to more branching in the blockchain.

3.3 Varying percentage of nodes with lower CPU power (z_1)

For this analysis, we will take the number of peers in the network to be 20, z_0 to be 20%, the interarrival time between transactions (T_{tx}) to be 1000 seconds and the time for Block Generation to be 600 seconds.

- For z_0=0%, the number of transactions included in each block generated by a node is approximately 8
- For z_0=100%, the number of transactions included in each block generated by a node is approximately 18

As can be seen, there are more transactions present in a block generated by a node that has lower hashing power compared to a node having higher hashing power. A possible reason for this observation could be that the nodes having lower hashing power will take a longer time to generate a block and in the meanwhile, the node will receive more and more transactions.

3.4 Varying the mean interarrival time between transactions (T_{tx})

For this analysis, we will take the number of peers in the network to be 20, z_0 and z_1 to be 20% and the time for Block Generation to be 600 seconds.

- For T_{tx}=300 seconds, the number of transactions in each block is approximately 57
- For z_0=1500 seconds, the number of transactions in each block is approximately 8

As T_{tx} is decreased, more transactions are generated by each node and hence more transactions are included in a block.

3.5 Varying the mean interarrival time between blocks `I`

For this analysis, we will take the number of peers in the network to be 20, z_0 and z_1 to be 20% the interarrival time between transactions (T_{tx}) to be 5000 seconds.

- For I=5 seconds, the number of branches in the blockchain tree is 7.

- For I=100 seconds, there are almost no number of branches in the blockchain tree

As can be seen, if the mean inter-arrival time between blocks is low, then there is more branching in the blockchain tree. A possible reason behind this observation could be that a node `A` may generate a block and transmit it while at the same time, some other node `B` may have already transmitted its block as it may not have received the transmitted block generated by `A` due to network delays.

3.6 Ratio of number of blocks generated in blockchain to total number of blocks created by a node

We used n=40, with 10 peers belonging to each of the 4 categories formed based on transmission speed and hashing power (i.e. $z_0 = 50$ and $z_1 = 50$). For the sake of testing, we reduced inter-arrival times between blocks to 1 sec increasing conflict branches and signifying the difference between high and low-speed peers.

We observe the ratio of blocks in the longest chain to total number of blocks generated by peers grouped based on the categories as shown in Figure 1.

As expected, the ratio of number of blocks generated in the blockchain to the total number of blocks created by nodes that are slow and have less hashing power is the lowest while the ratio for nodes that are fast and have high hashing power is the highest amongst all the types of nodes. In between, we can observe that the nodes that are slow and have higher hashing power have a better ratio than those nodes that are slow but have less hashing power. A possible reason behind this could be that the nodes that have higher hashing power are able to generate more blocks than the nodes with less hashing power and even if some of those blocks may not get transmitted to their neighbors on time, most of them will be present in the longest chain. On the other hand, even if the nodes that have high transmission speed are able to send their blocks to their peers on time, they aren't able to generate enough blocks due to their low hashing power.

3.7 Length of branches

To calculate length of branches, we first find leaf nodes (the nodes which don't have any child). We traverse up along the tree on these leaves to find the distance upto genesis block. We found branch length as [7, 8, 10, 10, 12] for figure 2.

4 Network Topology

We have the contraint that every peer must be connected to between 4 and 8 other peers. At each iteration, we pick a node, check the number of nodes it is

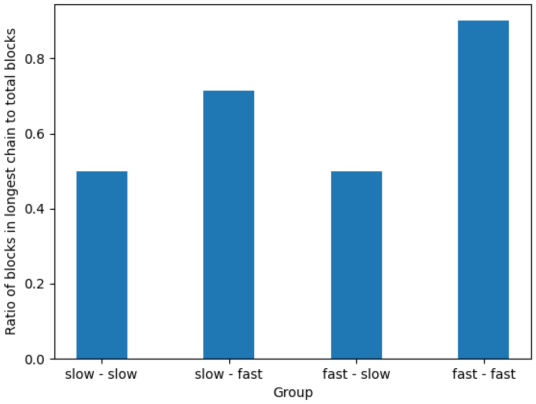

Figure 1: Bar plot of ratio based on different groups (transmission speed, cpu power)

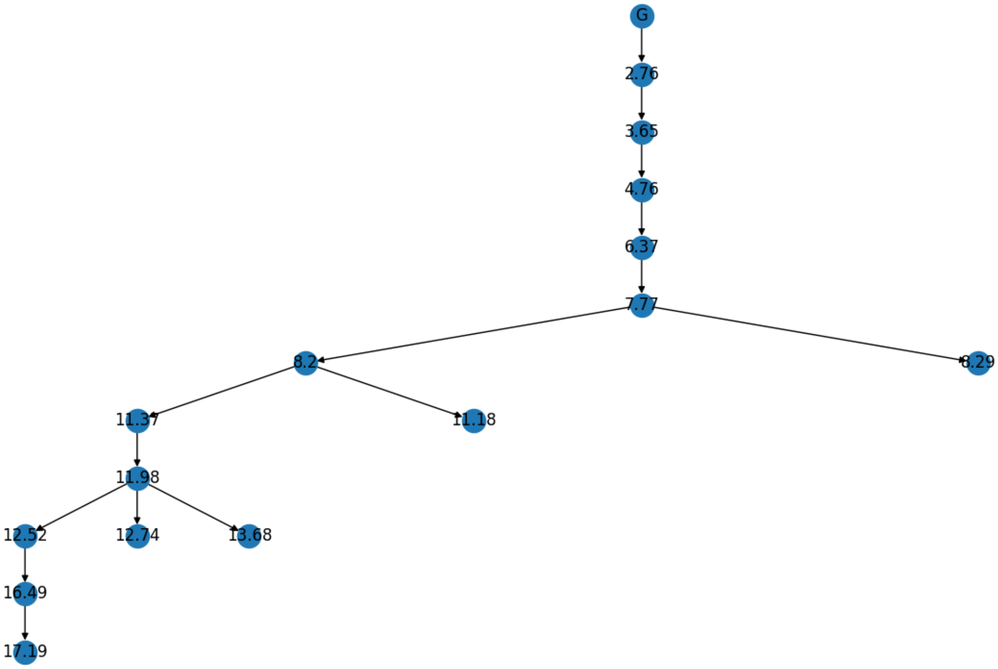

Figure 2: Length of branches

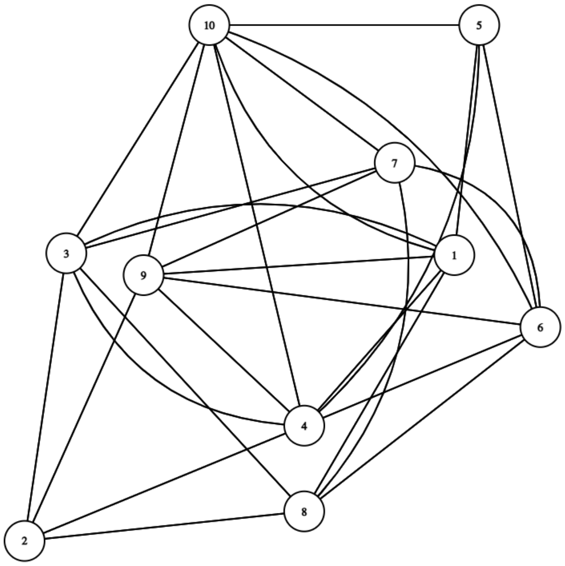

Figure 3: Network generated for 10 peers

connected to, select a number uniformly randomly so that finally the number of edges will lie between 4 and 8. Refer to Figure 3.

5 Blockchain Trees

We used networkx to generate visualization of blockchain tree formed. Below are some typical examples for different interarrival time between blocks. We also print time of block generation/receive as observed by a peer.

We observe greater forks when the interarrival time of blocks is less as seen in Figure 5. We observe lesser forks when interarrival time of blocks is higher as seen in Figure 4. We reason this observation with the increase in generation of blocks in a short span of time when decreasing the interarrival time. When interarrival time is high, the probability of generation of two or more blocks around a (network propagation delay) of each other is less. When it's less, then the probability is higher and hence multiple blocks are mined and broadcast before another block can reach the peer and invalidate the mined block.

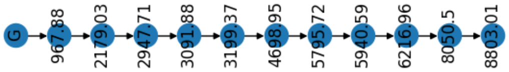

Figure 4: Blockchain formed for interarrival time = 600sec - No branching

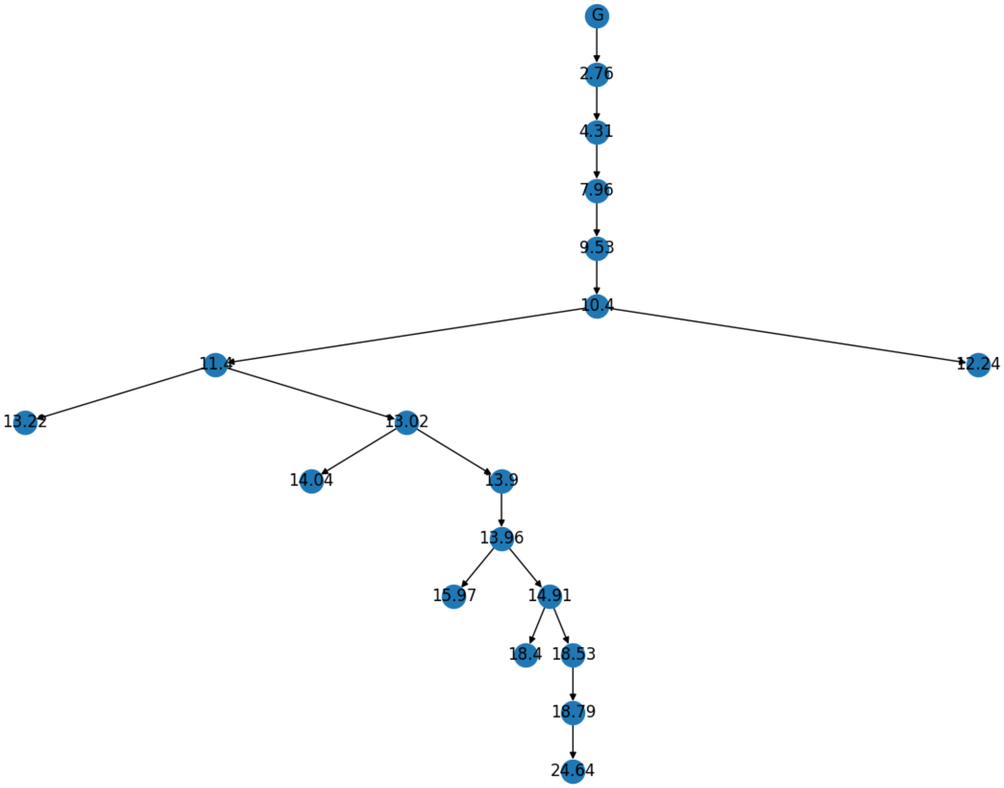

Figure 5: Blockchain formed for interarrival time = 1 sec - Branching