CS 765 : Report - Project Part 1

Simulation of a P2P Cryptocurrency Network

Aakriti - 190050002
Aditya Badola - 190050006
Gaurang Dev - 19D070024

September 7, 2021

1 What are the theoretical reasons of choosing the exponential distribution for inter-arrival time between transactions?

Let T denote the time we need to wait, before the next transaction occurs (i.e. the inter-arrival time between transactions). If T has an exponential distribution, that implies:

$$\mathcal{P}(\beta < T < (\beta + \Delta\beta)) \propto \Delta\beta$$

This assumption is very practical and hence exponential distribution is a good estimate for inter-arrival time between transactions.
Moreover, the exponential distribution has **memoryless property**, that is, the inter-arrival time between two consecutive transactions is independent of the previous occurrences, and the number of transactions in given time follows Poisson distribution.

2 Algorithm used for connecting the peers randomly such that the resultant network is connected

Define p to be the probability with which two nodes are connected in an arbitrary network. For every two nodes i and j ($i, j \in [n]$), sample a random variable from a uniform distribution $\mathcal{U}(0, 1)$, call it r. If $r < p$, connect the nodes i and j. After iterating it over all the node pairs, we get a network. If the network is connected, we are done. Else, repeat the same algorithm (from scratch) until the network gets connected. *Note:* p determines the sparsity of the network.

3 Why is the mean of d_{ij} inversely related to c_{ij}?

d_{ij} denotes the queuing delay at node i to forward a message to node j. Whereas, c_{ij} is the link speed between nodes i and j. c_{ij} determines how fast or slow the link propagates its queue. Higher the link speed, it will take lesser time to propagate the message and hence the queue will be cleared faster. Therefore, d_{ij} is inversely proportional to c_{ij}.

4 Justify the chosen mean value for T_k.

T_k is the inter-arrival time between blocks. Higher value of T_k would generate less number of blocks and transaction limit for each block will get saturated. On the other hand, lower value of T_k would mean huge number of blocks being generated and there will be fewer transactions in each block. We need an optimal value of T_k, hence mean value of T_k has to be justifiably chosen.

5 Variation with fraction of slow nodes (z)

When we increase the fraction of slow nodes, the fraction of slow links increases as a result of which lesser number of blocks are broadcasted amongst the nodes in the given time of simulation. Therefore, the length of blockchain decreases.
Following are the blockchains for peer 0:

- **Case 1:** $z = 0\%$ (All peers are fast)

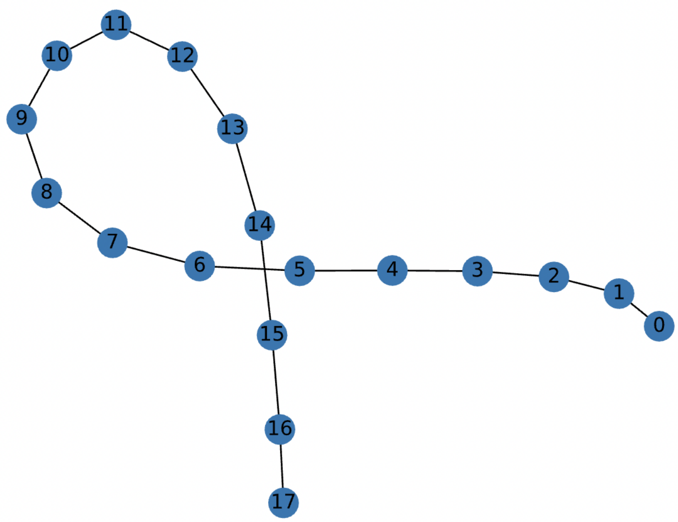


Figure 1: length of blockchain: 18

- **Case 2:** $z = 50\%$ (Half of the nodes are fast)

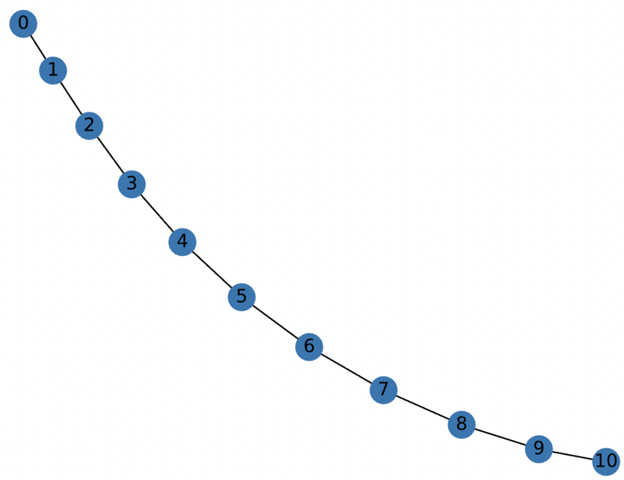


Figure 2: length of blockchain: 11

- **Case 3:** $z = 100\%$ (All nodes are slow)

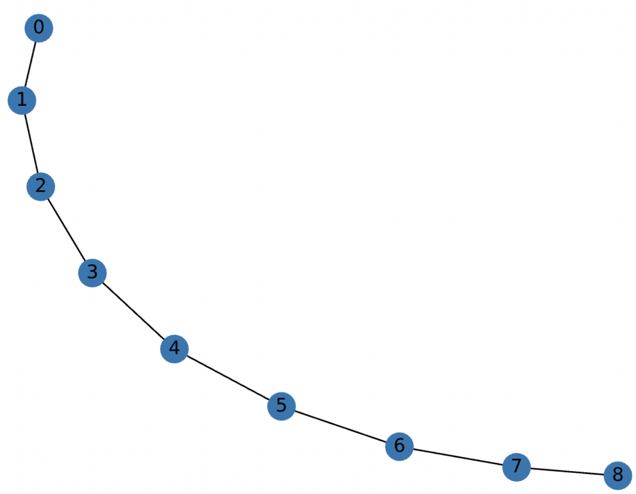


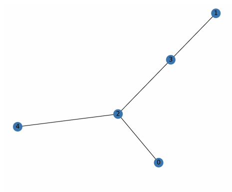
Figure 3: length of blockchain: 9

6 Variation with number of nodes (n):

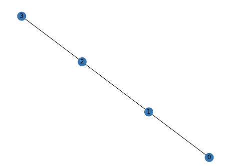
When we increase the number of peers, more peers will be mining for the blocks. So, there will be an increase in the number of block generation events, that is, there will be more blocks in the blockchain.

Following are the blockchains for peer 0:

- **Case 1:** $n = 5$

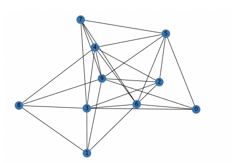


(a) Peer network

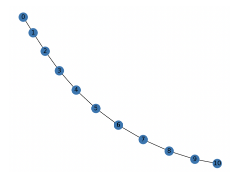


(b) length of blockchain: 4

- **Case 2:** $n = 10$

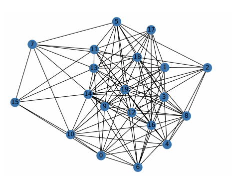


(a) Peer network

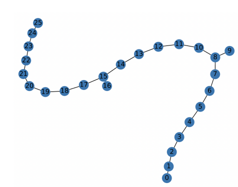


(b) length of blockchain: 11

- **Case 3:** $n = 20$



(a) Peer network



(b) length of blockchain: 25, **forking**

7 Variation with the mean transaction inter-arrival time (T_{tx}):

As $T_{tx}(\propto \frac{1}{\lambda_{tx}})$ increases, the interarrival time between two consecutive transactions increases. So, less number of transactions will be generated given the simulation time, and therefore, each block will have lesser number of transactions.
On the other hand, as T_{tx} decreases, more transactions will be generated by every node, and hence number of transactions in each block increases.
Following are the blockchains for peer 0:

- **Case 1:** $\lambda_{tx} = 0.8$

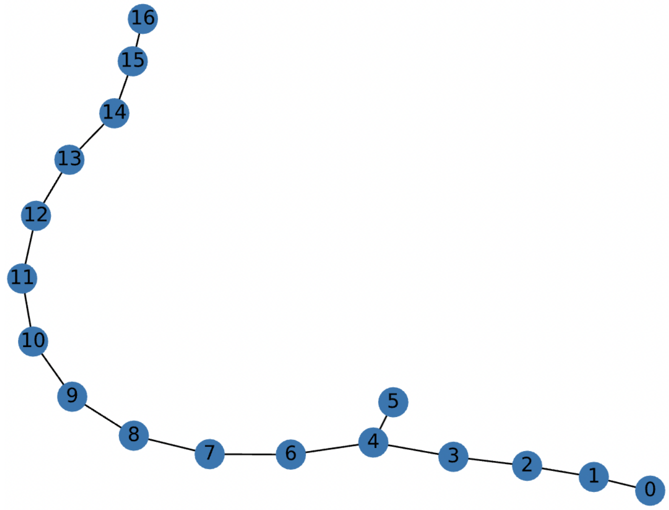


Figure 7: length of blockchain: 17

- **Case 2:** $\lambda_{tx} = 2$

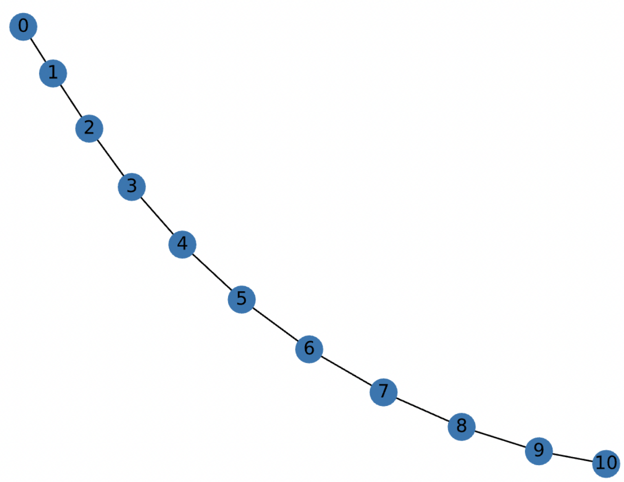


Figure 8: length of blockchain: 11

- **Case 3:** $\lambda_{tx} = 4$

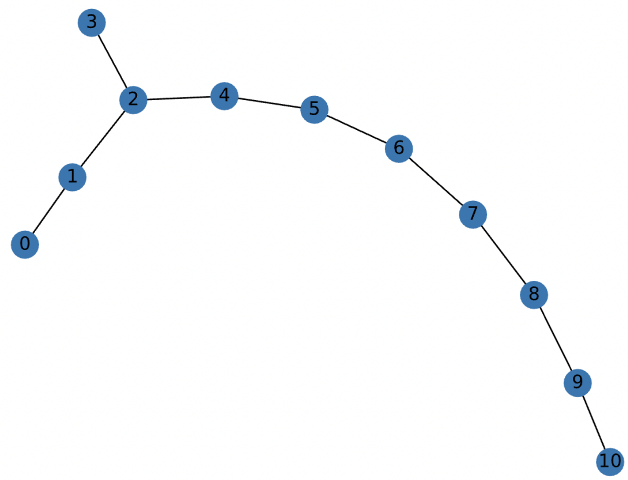


Figure 9: length of blockchain: 11, **forking**

8 Variation with the mean block inter-arrival time (T_k):

As $T_k(\propto \frac{1}{\lambda_k})$ increases, the interarrival time between two consecutive blocks increases. So, less number of blocks will be generated given the simulation time, and therefore, there will be lesser branching.
On the other hand, as T_k decreases, more blocks will be generated by every node, so nodes are extending their longest chain in a very short duration of time. As a result, there will be more forking.
Following are the blockchains for peer 0:

- **Case 1:** $\lambda_k = 0.001$

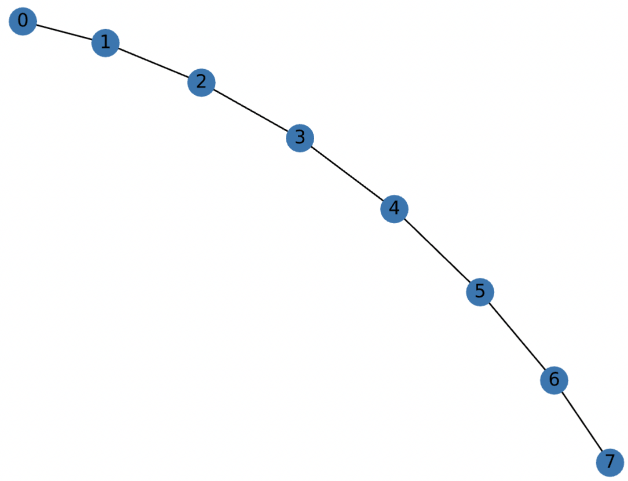


Figure 10: length of blockchain: 8

- **Case 2:** $\lambda_k = 0.0025$

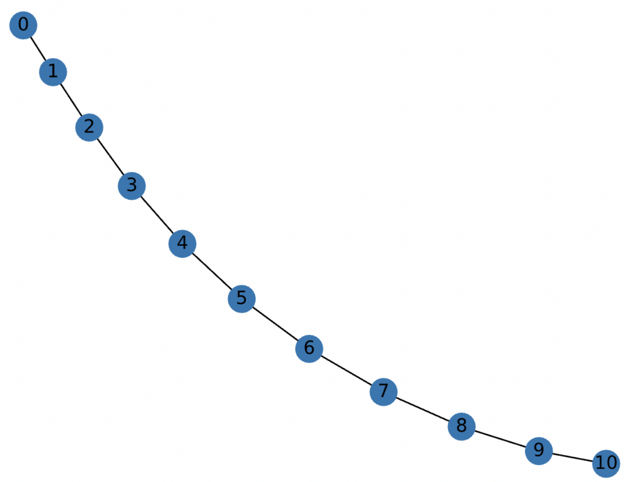


Figure 11: length of blockchain: 11

- **Case 3:** $\lambda_k = 0.005$

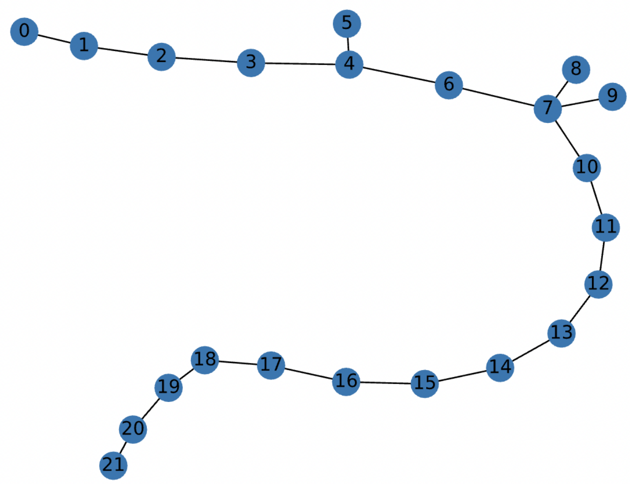


Figure 12: length of blockchain: 22, **forking**

9 Variation of ratio of the number of blocks generated by each node to total number of blocks generated

- **Case 1:** $z = 50, \lambda_k = 0.005$

Node ID	Type of Node	Ratio
0	slow	0.0217391
1	fast	0.152174
2	slow	0.0652174
3	fast	0.108696
4	slow	0.0869565
5	slow	0.0434783
6	fast	0.173913
7	fast	0.152174
8	fast	0.108696
9	slow	0.0434783

- **Case 2:** $z = 50, \lambda_k = 0.0025$

Node ID	Type of Node	Ratio
0	slow	0.0588235
1	fast	0.176471
2	slow	0
3	fast	0.117647
4	slow	0
5	fast	0.117647
6	slow	0
7	fast	0.0588235
8	fast	0.176471
9	slow	0.0588235

- **Case 3:** $z = 50, \lambda_k = 0.001$

Node ID	Type of Node	Ratio
0	fast	0.2
1	fast	0.2
2	slow	0
3	fast	0.4
4	slow	0
5	slow	0
6	fast	0.2
7	slow	0
8	fast	0
9	slow	0

As λ_k increases (or CPU power increases), the mean inter-arrival time between blocks T_k decreases. So, blocks are generated more often. Therefore, the ratio of blocks generated by slow nodes to the total number of blocks becomes almost same as the ratio of blocks generated by fast nodes to the total number of blocks.
On the other hand, when λ_k is low, blocks are generated at a slow rate. So, the latency of propagation between the nodes play an important role. The faster nodes will generate blocks much more often as compared to slow nodes (look at table 3).
As z denotes the percentage of slow nodes, if z is at extreme values, i.e. 0 or 1, the ratio is equal for all nodes regardless of it being fast or slow. On the other hand, when z is say 50%, the ratio is higher for the fast nodes, given that time of simulation and peers are constant.