CS765 Project Part 1
Simulation of P2P Cryptocurrency Network

Magham Dipen Anjan - 210050092
Patil Vipul Sudhir - 210050115
Hari Prakash Reddy - 210050119

January 2024

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

Let's try to model the inter arrival time.
Let I be the random variable which models inter arrival time between blocks, β be the parameter proportional to the likelihood that a transaction is generated at any instant and let Δ be the very small time interval.
The probability of an event(probability of success) occurring can be taken as $\beta\Delta$. Probability of arrival time = $n\Delta$ is given by -

$$P(I = n\Delta) = (1 - \beta\Delta)^{n-1}(\beta\Delta)$$

Let x denote Δn. Probability of inter arrival time being greater that $n\Delta$ is equivalent to Probability that there is no block for time Δ -

$$P(I > x) = \left(1 - \frac{\beta x}{n}\right)^n$$

Now apply limit Δ to zero and to signify the event can happen at any instant and thus having n tending to infinity to have a non zero time interval we get-

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

$P(I > x)$ is Cumulative Distribution Function for the exponential distribution. So the reason for choosing the exponential distribution is that it models the inter arrival time of a system where at each instant in time there is some probability of an event occurring, in this case a transaction being generated by any peer.

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

The queuing delay (d_{ij}) is sampled from exponential distribution which has distribution $P(X = x) = \beta e^{-\beta x}$, where β is the rate parameter. For exponential distribution rate parameter (β) is related to the mean as $1/mean$. Given that mean of queuing delay (d_{ij}) is $96kbits/c_{ij}$. So-

$$\beta = 1/mean = c_{ij}/96kbits$$

So β is directly proportional to c_{ij}. Also we know that higher the β higher is the probability of obtaining a lower value on sampling the random variable. The justification we can give is that when the link speed (c_{ij}) is higher, packets are transferred more quickly across the link, resulting in shorter waiting times at the queue that is shorter queuing delay (d_{ij}).

Q: Given interarrival time between blocks on average is I, and node k has fraction h_k (where $0 < h_k < 1$) of the total hashing power. Then T_k is drawn from an exponential distribution with mean equal to I/h_k. Give justification for choice of a particular mean.

A ndode with more hashing power should have a lesser mean inter arrival time. And also we want T_k be the average of the inter arrival time between any two blocks from any 2 nodes is I. Probability that none of the peers genertae a block in some time is given by-

$$P(I_1, I_2, ...I_n > x) = P(I_1 > x)P(I_2 > x)...$$

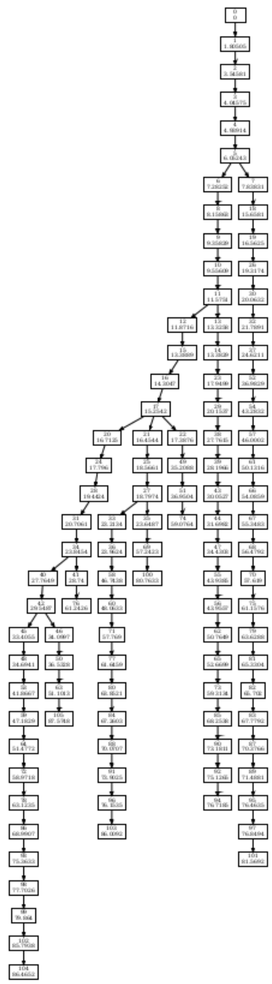
As we know that each miner can generate block independently so using Independence of the Random variables-

$$\prod_{i=1,2,...n} e^{-h_i x/I} = e^{-\sum h_i x/I} = e^{-x/I}$$

This is the result we expected as per definition of I. So the mean time is chosen as I/h_i.

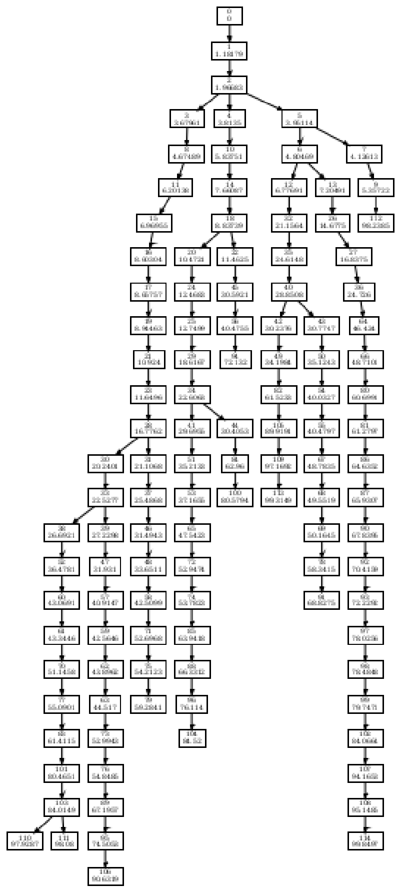
Q8: Observations:

Figure 1: Peers: 20, slow%: 10, lowcpu% :90, Blk interarr. time: 1



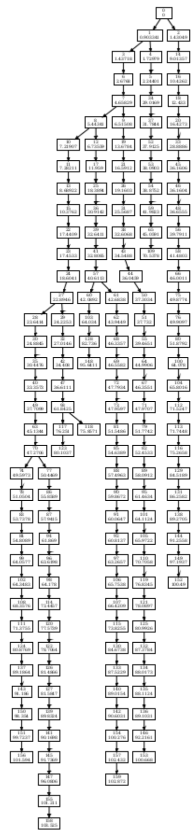
We can see moderate branching for given values of parameters as higher block inter arrival time is compensated by higher num of peers.

Figure 2: Peers: 20, slow%: 10, lowcpu% :10, Blk interarr. time: 1



Here we can see high branching although block inter arrival time and num peers remain constant because of low cpu percent. As we have decreased low cpu percentage implies that it take less time to propagate the block so probability of forks increases so branching is increased.

Figure 3: Peers: 15, slow%: 10, lowcpu% :70, Blk interarr. time: 0.7



In this case number of peers decreased from 20 to 15 and block inter arrival time is reduced to 0.7 seconds. As inter arrival time of block is less we can see branches.

Figure 4: Peers: 10, slow%: 30, lowcpu% :90, Blk interarr. time: 6

Here block inter arrival time is increased to 6 from 0.7 seconds. Since inter arrival time is increased, it gives more time to propagate through network so branching is less.

Figure 5: Peers: 10, slow%: 30, lowcpu% :90, Blk interarr. time: 6

Here block interarrival time is almost doubled (from 6 to 13 seconds) and we can see that there are very few brances (None here). (The given image is half image of generated blockchain)

Varying different parameters

Varying Txn Time:
The txns and blocks are independent so any change in the txn time will not change the block chain much.

Varying Number of Peers:
As number of peers increases the branching should increase in general, since each block now has to pass through a larger topology i.e. network so it takes more time than in the case of lower number of peers, and so, the probability of another block being generated during this time becomes higher so branching increases.
Other Fixed Quantities: slowpercent = 50, lowcpupercent = 50, intertxntime = 600 , interblktime = 0.5 seconds.

Varying Number of Peers

Num peers	Blocks in longest Chain (C)	Total Blocks (T)	C/T
5	185	585	0.3163
10	55	230	0.2391
15	37	153	0.2418
20	22	105	0.2095

Varying CPU power:
On increasing number of low hashing power nodes, each block takes some more time to genertae and hence, gives more time to resolve the created forks and hence branching should be less on increasing low CPU peers.

Other Fixed Quantities: numpeers = 10, slowpercent = 50, intertxntime = 600 , interblktime = 1 second.

Varying CPU power

CPU power	Blocks in longest Chain (C)	Total Blocks (T)	C/T
10	22	83	0.2651
30	29	95	0.3053
50	37	110	0.3364
70	44	113	0.3894

Varying slow CPU peers:
On increasing the number of slow peers, the branching should increase in general as it will take longer time to propogate through network, and so, probability of another block being generated during this time becomes higher. Other Fixed Quantities: numpeers = 10, lowcpupercent = 50, intertxntime = 600 , interblktime = 1 second.

Varying slow CPU peers

slowpercent	Blocks in longest Chain (C)	Total Blocks (T)	C/T
10	27	106	0.2547
30	26	109	0.2385
50	27	112	0.2411
70	22	98	0.2245

Varying Block Inter arrival time:

As interarrival time between blocks increases the generated block can propagate through network before next block is generated so the branching should be decrease. Other Fixed Quantities: numpeers = 10, slowpercent = 50, lowcpupercent = 50, intertxntime = 600.

Varying Block Inter arrival time

Blk inter arrival time	Blocks in longest Chain (C)	Total Blocks (T)	C/T
0.1	17	119	0.1429
0.5	29	109	0.2661
1	27	99	0.2727
10	61	83	0.7349