Report on: Simulation of a P2P Cryptocurrency Network

Assignment - 1

Akash Kumar (213050020)

Hrishikesh Saloi (213050057)

Manoj Kumar Maurya(213050067)

Contents

1 Why the interarrival between transactions generated by any peer is chosen from an exponential distribution ?

So we are dealing with the time where we have to wait for a given event (generate transaction) to occur. If this given waiting time is unknown, it is somehow appropriate to think of it as a random variable with the having an exponential distribution.The time let's say 'X' we need to wait before an event occurs has an exponential distribution. if there is probability that the event occurs during a certain time interval is proportional to the length of that time interval. On applying probability, X has an exponential distribution if the conditional probability:

$$P(t < X \leq t + \Delta t | X > t)$$

is approximately proportional to the length of Δt the time interval in between the times t and t + Δt , for any time instant t . In the practical view of simulation using this exponential distribution is more realistic that's why we are using this distribution for waiting time generation of next event (transaction).

One of the most important properties of the exponential distribution is the memoryless property. When the event can occur more than once and the time elapsed between two successive occurrences is exponentially distributed and independent of previous occurrences, the number of occurrences of the event within a given unit of time has a Poisson distribution. The exponential distribution is also tightly coupled with poisson distribution too.

2 Justification for n peers randomly connected graph using a chosen distribution

We used Networkx module of python to generate connected graph of nodes.A random number is generated from an uniform distribution between N-1and (N*(N-1))/2 i.e the maximum number of edge in a graph of n nodes.We treat this number as the number of edge in the graph.This will make sure that the graph always remain connected

3 Latency calculations

3.1 Justification for why is the mean of $\mathbf{d_{ij}}$ inversely related to $\mathbf{c_{ij}}$

c_{ij} is the link speed between i and j node and d_{ij} is the queuing delay at node i to forward the message to node j. So the speed of c_{ij} will decide how fast a node is able to process it's queue in resultant it will give inverse proportionality among c_{ij} and d_{ij}

the latency is calculated using the below formula

$$\rho_{ij} + |m|/c_{ij} + d_{ij}$$

where,
c_{ij} is the link speed between nodes i and j in bits per second

d_{ij} is the queuing delay at node i to forward the message to node j.d_{ij} is randomly chosen from an exponential distribution with some mean 96kbits/c_{ij}. d_{ij} is randomly chosen for each message transmitted from i to j.

ρ_{ij} is always chosen from a uniform distribution between 10ms and 500ms at the start of the simulation. c_{ij} is set to 100 Mbps if both i and j are fast, and 5 Mbps if either of the nodes is slow.

$|m|$ denotes the length of the message in bits. where length of each transaction size is 1KB and maximum length of Block is 1MB.

4 Simulation of Proof of work

4.1 Justification for the chosen mean value for T_k from an exponential distribution

T_k (or B_m) is the mean wait time for generation of blocks. So higher value of T_k would result in fewer blocks being generated and vice-versa respectively.If we have too many blocks the number of transaction per block is reduced, whereas if we have too few blocks then node transaction queue is filled up. So we need a balance among these two

5 Summary of effects of choosing different values of simulation parameters:

Simulation is done for 200 events,

For simplicity ,we have considered the fast nodes as high cpu node and slow nodes as low cpu node

Let,

$$R_d = \frac{\text{Ratio of blocks generated in the longest chain}}{\text{Ratio of blocks generated ot the end of simulation}}$$

N=No. of nodes

T_x=Mean block inter arrival time

T_{tx}=Mean transaction inter arrival time

Z=Percentage of fast nodes

We are considering the following factors:

- **Fraction of Slow Nodes (Z)**
- **Mean transaction interarrival time (T_x)**
- **Mean block interarrival time (T_{tx})**

6 Observations

1) $N = 5, T_{\mathrm{x}} = 40, T_{\mathrm{tx}} = 20, Z = 50$
On fast nodes, $\mathrm{R_d}$=.67
On slow nodes, $\mathrm{R_d}$=.78

2) $N = 10, T_{\mathrm{x}} = 40, T_{\mathrm{tx}} = 20, Z = 50$
On fast nodes, $\mathrm{R_d}$=.52
On slow nodes, $\mathrm{R_d}$=.43

3) $N = 5, T_{\mathrm{x}} = 50, T_{\mathrm{tx}} = 20, Z = 50$
On fast nodes, $\mathrm{R_d}$=.83
On slow nodes, $\mathrm{R_d}$=.68

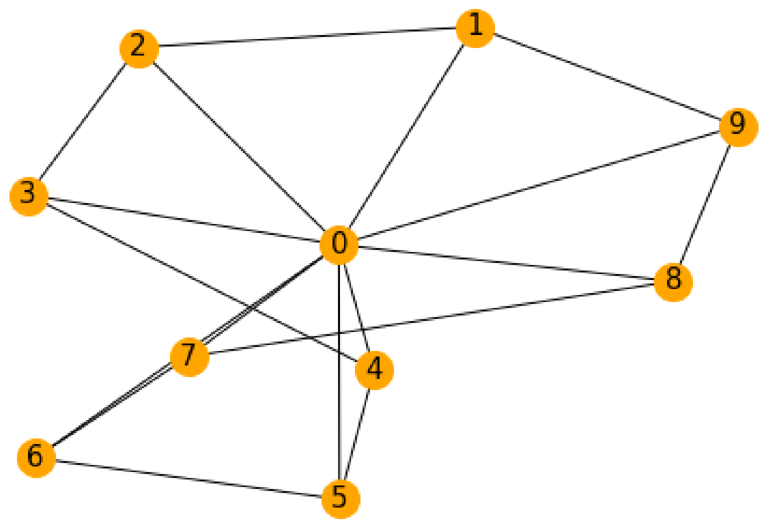
4) $N = 5, T_{\mathrm{x}} = 40, T_{\mathrm{tx}} = 30, Z = 50$
On fast nodes, $\mathrm{R_d}$=.87
On slow nodes, $\mathrm{R_d}$=.70

5) $N = 5, T_{\mathrm{x}} = 40, T_{\mathrm{tx}} = 20, Z = 80$
On fast nodes, $\mathrm{R_d}$=.74
On slow nodes, $\mathrm{R_d}$=.57

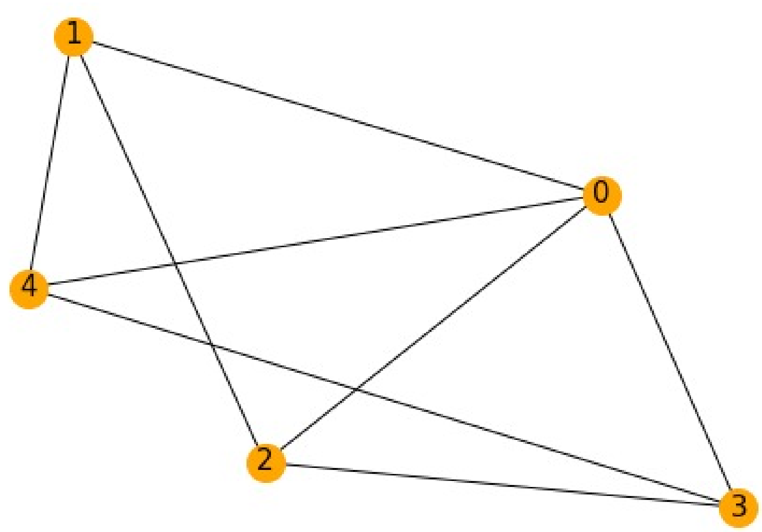
6) $N = 5, T_{\mathrm{x}} = 60, T_{\mathrm{tx}} = 10, Z = 50$
On fast nodes, $\mathrm{R_d}$=.78
On slow nodes, $\mathrm{R_d}$=.57

7 The Above Observation is based on following interconnection between the peer nodes

7.1 interconnection between the 10 peer nodes



7.1.1 interconnection between the 5 peer nodes



8 Conclusion of Observations

- R_d Value of fast nodes is almost in all cases greater than that of slow nodes.

- On increasing the number of nodes R_d value of both fast and nodes decreases to a great extent.

- As we increase the T_x (mean block interarrival time) R_d value of both fast and slow notes increase.

- As we increase the $T_{t\,x}$ (mean block interarrival time) R_d value of both fast and slow notes increase.

- As we increase the % of fast notes R_d value of fast note increases while R_d of slow notes decreases.

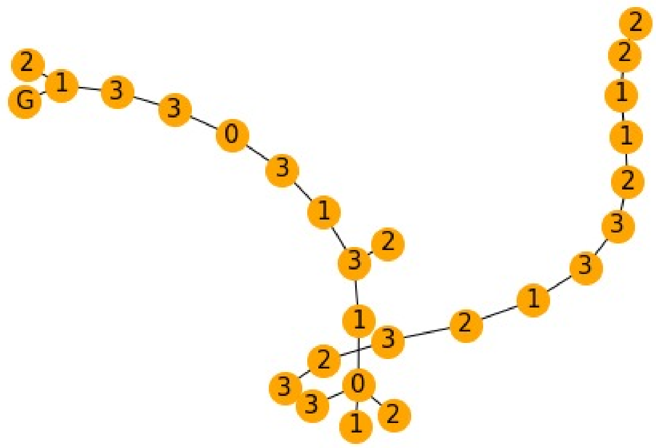
- As we increase the gap between T_x and $T_{t\,x}$ both R_d for fast and slow notes decreases.

9 Picture of typical trees

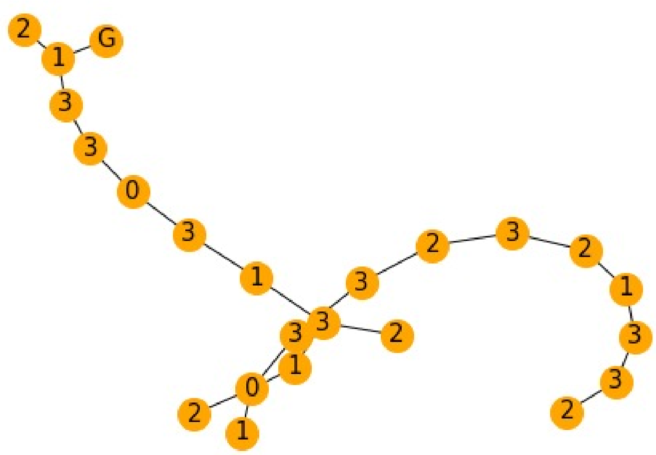
9.1 some snippets of the blockchain tree when N=5, Z=80,T_x=40, $T_{t\ x}$=20

Here the labels of blocks are node id of its creator

9.1.1 Fast nodes tree

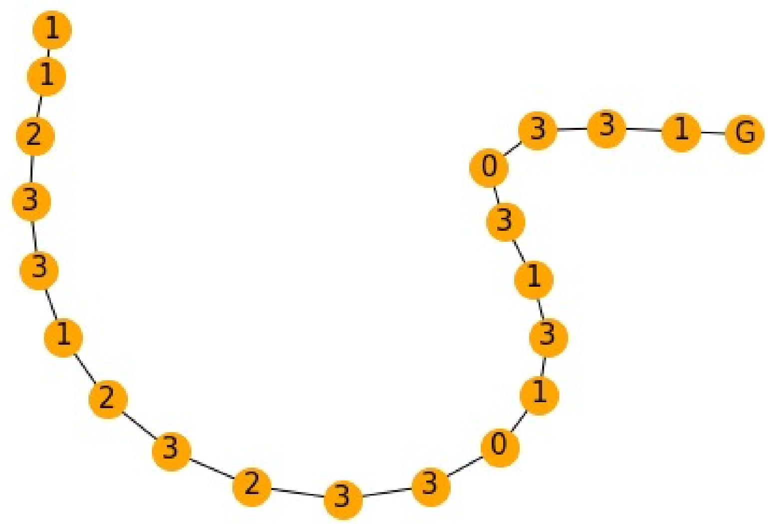


9.1.2 Slow nodes tree

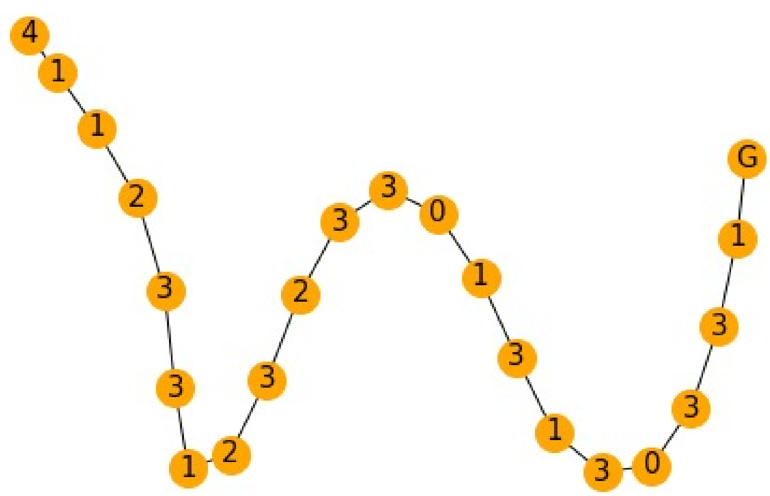


9.2 Some snippets of the longest chain

9.2.1 Fast nodes Longest chain tree



9.2.2 Slow nodes Longest chain tree



10 Bibliography

References

[1] https://github.com/dvf/blockchain/find/master

[2] https://jis-eurasipjournals.springeropen.com/track/pdf/10.1186/s13635-019-0085-3.pdf

[3] https://medium.com/@amannagpal4/how-to-create-your-own-decentralized-file-sharing-service-using-python-2e00005bdc4a

[4] https://pypi.org/project/pyp2p/

[5] https://www.youtube.com/watch?v=b81Ib_oYbFk&list=PLBOh8f9FoHHhNmKlVW8TYKnNoCCf8srPh

[6] https://github.com/dufferzafar/crypto-simulation/blob/master/Problem%20Statement.pdf

[7] https://github.com/macsnoeren/python-p2p-network