CS765 Assignment1

Team members :
Gadde Nitish Samanth - 180050031 (180050031@iitb.ac.in)
Tekuri Sai Akhil - 180050112 (180050112@iitb.ac.in)
Urabavi Revanth Kumar - 180050114 (180050114@iitb.ac.in)

Point 2:
There should be a mean interarrival time for transactions generated by a peer but the distribution should be memoryless since the probability that a transaction will be made in the next some amount of time will not and should not depend on the transactions made till now. Since the only distribution which is memoryless is exponential, we choose this distribution to model the interarrival time of transactions. Moreover, the probability that a transaction is generated in a small amount of time can be thought of depending linearly on the time interval. So, the resulting distribution is exponential.

Point 4:
We used this research as a reference which models the degree of connectivity of nodes as a power law distribution. This paper studies the degree distribution of the bitcoin network containing 7025 nodes. From the analysis, it is shown from the data that the out-degree distributions follow power-law. In this distribution, few nodes have high degree of connectivity and more nodes have low degree of connectivity. Two algorithms (first-mover wins and fittest-gets-richer) are given for network generation in the paper which outputs networks in which each node follows power-law. We have implemented the fittest-gets-richer algorithm in our simulation.

Point 5:
Transmission delay at a node = |m| / c_ij
So, if c_ij is high then a transaction/block is transmitted faster, which allows the node to transmit some other transaction/block earlier, which implies less queuing at the node. So, a block / transaction needs to wait for less time on an average to get transmitted when c_ij is high. So, the queuing delay(d_ij) is less when c_ij is high

Therefore, average of d_ij is inversely proportional to c_ij

Point 7:
We have taken 2 classes of nodes: one with high hashing power and one with low hashing power. The hashing power of HIGH CPU nodes is double the hashing power of LOW CPU nodes.Therefore, nodes with high hashing power wait half the time nodes with low hashing power waits on an average to transmit the block generated by it. We have taken the percent of high cpu nodes as simulation input. Generally, we have to set the mean interarrival time of blocks so that there won't be much branching (so high mean is preferrable) and there should not be transactions filling up the transaction queue (so low mean is preferrable). So, we should pick a value which balances both. For n=20,T_tx=5, it is better to have B_tx as 125, since there will be 500 transactions in the block approximately.

Implementation Details:

We used python and the library simpy to simulate the blockchain. There are mainly 6 types of events in the simulation namely transaction generation, transaction broadcasting to peers, transaction routing to peers, Block generation, Block broadcasting and Block routing to peers. The stack maintenance is done by using the library simpy.

We stopped the simulation after some certain specified amount of time(500 sec).At the end of simulation each node creates a file and writes the time at which the blocks arrived at the node along with some other information.

Summary of study of blockchain with different parameters:

Effect of n(with z=30,T_tx=3,B_tx=10,stop_time = 500):

- When n=8, the number of transactions in the block is approximately 30, but when n=20, the number of transactions went upto 100 approximately.
- There is more branching in the blockchain as n increases. When n=8 , there is almost no branching , but when n is increased to 20, there is significant branching in the blockchain

Effect of z(with n=20,T_tx=3,B_tx=10,stop_time=500) :

- As z is increased, there is more branching in the blockchain since the broadcasted blocks take more time to reach all nodes in the network and some other node may broadcast its block in the meantime.
  For z=0, number of branches = 2 and for z=100, number of branches=5
- Number of transactions in the block decreased when z is increased since many transactions may be in the network when the block is generated.
  For z=0,number of transactions = 90 and for z=100, number of transactions = 50

Effect of T_tx (with n=20,B_tx=10,stop_time=500):

- As T_tx is decreased, there are more transactions included in a block since more transactions are generated by each node in an amount of time.
  For T_tx = 5, number of transactions = 27, for T_tx=2 number of transactions=65

Effect of B_Tx (with n=20,T_tx=3,stop_time=500):

- If B_Tx is low, then there is more branching of the blockchain. This may be due to the fact that the node transmits the block generated by it when some other node has already transmitted its block, since it may not get the transmitted block in such a small time due to network delay.
  For B_Tx = 5, number of branches are 12, for B_Tx=10, number of branches are 4

General Observation:

- Nodes which are slow have blockchains of shorter length than the nodes which are fast at an instant of time.

For inputs : n=20, z = 50, T_tx = 3, percent_high_cpu = 50, B_Tx = 10 and stop_time = 1000

Ratio of number of blocks generated in blockchain to total number of blocks created :
Average ratio for nodes with low speed and low hash power = 0.0276
Average ratio for nodes with low speed and high hash power = 0.0544
Average ratio for nodes with high speed and low hash power = 0.0425
Average ratio for nodes with high speed and high hash power = 0.0664

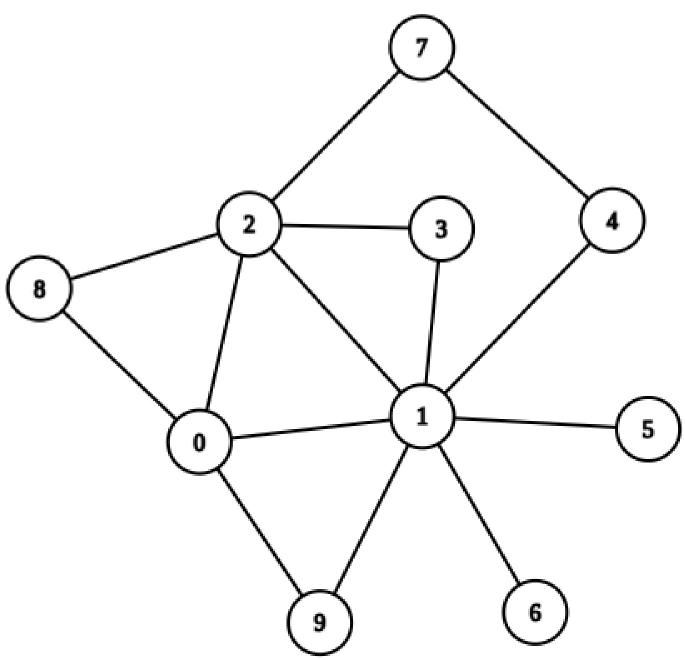
Lengths of branches of the tree : 77, 63, 61, 59, 38, 37, 21, 13, 3, 2

Explanation:

We took z = 50 and percent_high_cpu = 50 because only then the difference due to speed and hashing power is truly seen in the ratios. If it's not 50 then some nodes are more in number than the other due to which we can't really say that if the ratio is high, then it's due to high hashing power or high number of the nodes.

We can see that on increasing the hashing power, the ratio increases since a node has to wait for less time to broadcast the block created by it and there are less chances that a block is already broadcasted. Nodes with high speed has better ratio than nodes with low speed since at an instant of time high speed node mines on the high length blockchain but low speed node mines on the low length blockchain(due to network delays) and its mined block will mainly be a branch in the blockchain.

Typical network:



Typical blockchains:

We used gnuplot for visualization of blockchains. Below are some images of typical blockchains created.