Building a layer-2 DAPP on top of Blockchain

Shalabh Gupta 180050095
Vrinda Jindal 180050120

Graph/Network Sampling

The node network forms a graph which follows a power law distribution with coefficient 3. All real networks generally follow the power law degree distribution. Not only this, almost all some-what real bigger networks are scale free.

We have used the Barabasi-Albert model to sample our graphs (scale-free). We have used the networkx python library to implement the graph sampling.

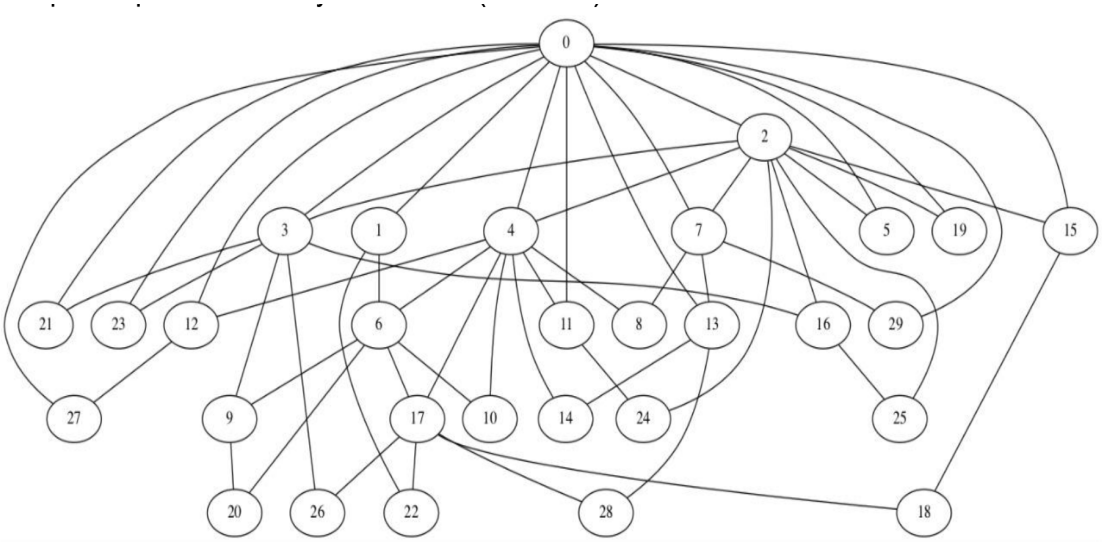
The reference mentioned below (Baumann et. al.) also concludes that the real bitcoin network converges to a scale-free graph.

In such graphs, the fraction P(k) of nodes in the network having k connections to other nodes goes for large values of k as

$$P(k) \sim k^{-3}$$

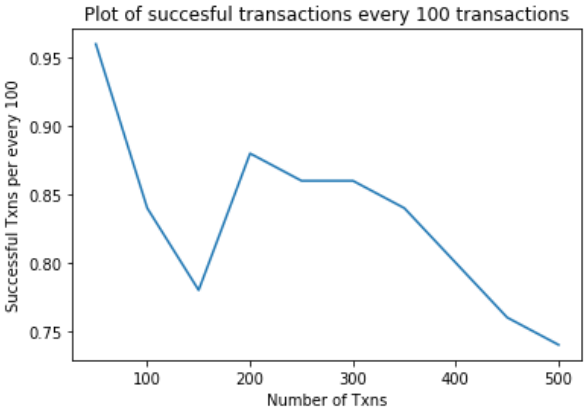
Also, when a new node is added to the network it first attaches itself to ‘m’ existing nodes. We have this parameter as 2.

Sample Graph Generated by this method (30 nodes):

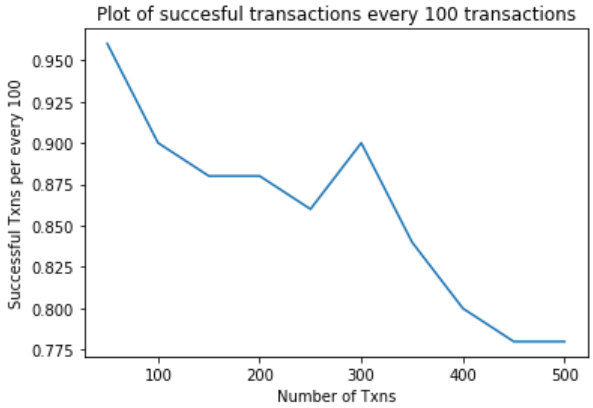


We could not run 1000 transactions due to lack of time. We ran 500 transactions in each of our machines parallelly. (Run1 and Run2) The graph obtained of ratio of successful transactions per every 50 transactions vs total number of transactions is given below:

Run 1



Run 2



With more and more transactions, the edges will block out (some nodes will have very less balance than the amount being transferred), and if even 1 edge blocks in the path between sender and receiver, it will lead to failure of the new transaction. Initially very few edges will be blocked, but with random transactions more and more edges block and

the probability of a transaction being successful reduces. This is why the number of successful transactions per 100 transactions show a decreasing trend against the total number of transactions. Sometimes there is a rise also, which is less probable but occurs due to some other edges unblocking also (balance rises in some nodes due to transaction).

Final Submission Details

- Solidity code : mainFunc.sol
- Python script.py
- README.txt
- requirements.txt
- genesis.json
- Report.pdf