

# Design Document HW2

**Sarthak Sharma (23M0789)**

**Kumaran Kartikeyan (23M0803)**

**Pulkit Suhag (23M0782)**

In our source code we use 11 python files and modules such as numpy, random, simpy, matplotlib, datetime, networkx, csv and hashlib. A brief information of the 11 files are given below:

- 1) **graph.py**: It is creating a graph which has all the required properties with minimum 3 other nodes and maximum of 6 nodes connected to it.
- 2) **networkproperties.py**: It is having an object of the network which is calling a method called `create_nodes` which is creating nodes and adding nodes into the `nodes_list` variable. Overall it creates the nodes accordingly as given in the problem statement as low/high CPU and fast/slow, 2 selfish nodes and `selfish_nodes` list selected while creating the network. It also has an `lvc` variable which is longest visible chain.
- 3) **node.py**: In this there is a transaction class and a node class. Transaction classes have different parameters as the one required. In node class we have the parameters and data structures which are used by a node, that is a genesis block, transaction list, blockchain, block queue and selfish queue (which is only used when the node is selfish).
- 4) **block.py**: It is having a block class which defines the structure of the block object.
- 5) **plot.py**: it is simply saving the topography of the network.
- 6) **transmitttransactions.py**: In this a node creates a transaction in exponential time distribution with the mean of `Tx`.
- 7) **transmit transactions.py**: In this a node is sending the transaction, which it creates to its neighbouring nodes.
- 8) **createblocks.py**: In this the nodes create blocks depending on the interarrival time and hashing power. It does this by checking the transactions in the blocks.
- 9) **handleblocks.py**: in this we are handling blocks by transmitting them according to the latencies, updating the longest value chain, and adding the blocks from the block queue. Checking whether transactions are valid.
- 10) **checktransactionqueue.py**: It is used to simulate latencies for transmitting the transaction from one node to another.
- 11) **main.py**: This is the main file which is used to create the environment to run the blockchain simulation. The variously asked command line parameters are given as parameters in this file.

Flowchart:

