Simulation on P2P Network of Selfish Miner & Stubborn Miner

Report on Assignment 2

CS765

Team Members

Siddharth Maurya
(213050033)

Yasaswini Dasari
(213050065)

Implementation:

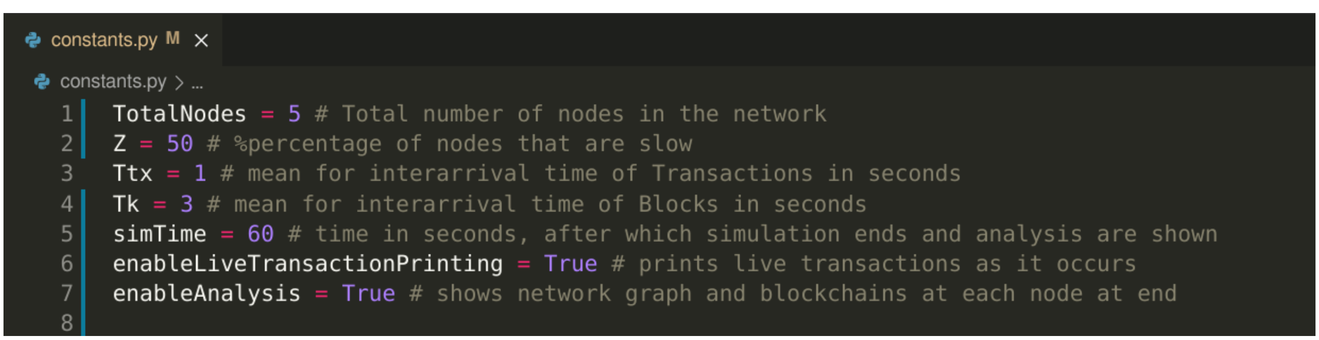
We have achieved the behaviour of a P2P network with the help of Discrete Event Simulation. To implement this discrete event simulation, and we used python language, due to its object-oriented programming capabilities along with SimPy Library to make programming simulation easier. We have separated all entities into classes. Each instance of Node does its work independently. Each node implements a `run()` subroutine in its class, which executes as a process for each node in the simulation environment. Further details have been given in the design document.

Instructions:

To start the simulation,

```
python3 simulation.py
```

To change the parameters (or to view default parameters), edit the *"constants.py"* file.


constants.py M ×
constants.py > ...

```
TotalNodes = 5 # Total number of nodes in the network
Z = 50 # %percentage of nodes that are slow
Ttx = 1 # mean for interarrival time of Transactions in seconds
Tk = 3 # mean for interarrival time of Blocks in seconds
simTime = 60 # time in seconds, after which simulation ends and analysis are shown
enableLiveTransactionPrinting = True # prints live transactions as it occurs
enableAnalysis = True # shows network graph and blockchains at each node at end
```

(Screenshot of constants.py)

Changing various parameters and analyzing their effects:

[Selfish Mining]

T_{tx}: Interarrival time for Transactions

We ran the simulation with following configuration,
TotalNodes = 10, Z = 50%, T_k = 20, HonestConnected = 70%, Alpha = 40%

As observed, interarrival time for transactions did not make much difference to affect the stubborn mining. The simulator does not have any minimum requirement limitations on building a block, so Blocks with very few transactions would be created, but the number of blocks wouldn't be affected.

T_k: Interarrival time for Blocks

We ran the simulation with following configuration,
TotalNodes = 5, Z = 50%, T_{tx} = 1, HonestConnected = 75%, Alpha = 30%

T_k	MPU_{adv}	$MPU_{overall}$
1	0.46	0.34
10	0.5	0.73

TotalNodes: Total number of peers in network

We ran the simulation with following configuration,
Tk = 8, Z = 50%, T_{tx} = 1, HonestConnected = 75%,
Alpha = 30%

TotalNodes	MPU_{adv}	$MPU_{overall}$
5	0.71	0.72
20	0.4	0.73

HonestConnected: Fraction of honest nodes directly connected to Adversary node

We ran the simulation with following configuration,
Tk = 8, Z = 50%, T_{tx} = 1, TotalNodes = 5, Alpha = 30%

TotalNodes	MPU_{adv}	$MPU_{overall}$
25%	0.25	0.52
50%	0.69	0.56
75%	0.8	0.72

As, more nodes are connected to adversary directly, it becomes easier for adversary to play its game. As it gets the latest update of main chain, and can send latest update of private chain quite faster bypassing network propagation delay.

Alpha: Mining Power of Adversary Node

We ran the simulation with following configuration,
Tk = 8, Z = 50%, T_{tx} = 1, TotalNodes = 5,
HonestConnected = 75%

TotalNodes	MPU_{adv}	$MPU_{overall}$
30%	0.42	0.33
50%	0.25	0.32
80%	0.13	0.14

For higher alpha, we can observe MPU to fall down drastically, the possible reason might be due to higher power at adversary, honest nodes suffer and fails to create new blocks, hence the less honest blocks comes to adversary, but private chain at adversary would still be quite long.

[Stubborn Mining]

T_{tx}: Interarrival time for Transactions

We ran the simulation with following configuration,
TotalNodes = 10, Z = 50%, T_k = 20, HonestConnected = 70%,
Alpha = 40%, simTime = 160

As observed in Selfish Miner, here too interarrival time for transactions did not make much difference to affect the stubborn mining. The simulator does not have any minimum requirement limitations on building a block, so Blocks with very few transactions would be created, but the number of blocks wouldn’t be affected.

T_k: Interarrival time for Blocks

We ran the simulation with following configuration,
TotalNodes = 5, Z = 50%, T_{tx} = 1, HonestConnected = 75%,
Alpha = 30%

T_k	MPU_{adv}	$MPU_{overall}$
1	0.51	0.46
10	0.75	0.77

TotalNodes: Total number of peers in network

We ran the simulation with following configuration,
Tk = 8, Z = 50%, T_{tx} = 1, HonestConnected = 75%,
Alpha = 30%

TotalNodes	MPU_{adv}	$MPU_{overall}$
5	0.66	0.83

20	0.5	0.45

HonestConnected: Fraction of honest nodes directly connected to Adversary node

We ran the simulation with following configuration,
Tk = 8, Z = 50%, T_{tx} = 1, TotalNodes = 5, Alpha = 30%

TotalNodes	**MPU_{adv}**	**$MPU_{overall}$**
25%	0.5	0.4
50%	0.83	0.66
75%	0.9	0.87

A similar trend, as discussed in Selfish Mining can be observed here.

Alpha: Mining Power of Adversary Node

We ran the simulation with following configuration,
Tk = 8, Z = 50%, T_{tx} = 1, TotalNodes = 5,
HonestConnected = 75%

TotalNodes	**MPU_{adv}**	**$MPU_{overall}$**
30%	0.33	0.58
50%	0.40	0.51
80%	0.25	0.20

A similar trend, as discussed in Selfish Mining can be observed here.

Selfish VS Stubborn:

Here we, look at the above data with a comparison aspect between selfish mining and stubborn mining. While analysing a parameter, others were kept constant as discussed earlier.

T_k: Interarrival time for Blocks

Parameter	*Stubborn Mining*		*Selfish Mining*	
T_k	**MPU_{adv}**	**$MPU_{overall}$**	**MPU_{adv}**	**$MPU_{overall}$**
1	0.51	0.46	0.46	0.34
10	0.75	0.77	0.5	0.73

TotalNodes: Total number of peers in network

Parameter	*Stubborn Mining*		*Selfish Mining*	
TotalNodes	**MPU_{adv}**	**$MPU_{overall}$**	**MPU_{adv}**	**$MPU_{overall}$**
5	0.66	0.83	0.71	0.72
20	0.5	0.45	0.4	0.73

HonestConnected: Fraction of honest nodes directly connected to Adversary node

Parameter	*Stubborn Mining*		*Selfish Mining*	
honestConn	$\mathbf{MPU_{adv}}$	$\mathbf{MPU_{overall}}$	$\mathbf{MPU_{adv}}$	$\mathbf{MPU_{overall}}$
25%	0.5	0.4	0.25	0.52
50%	0.83	0.66	0.69	0.56
75%	0.9	0.87	0.8	0.72

Alpha: Mining Power of Adversary Node

Parameter	*Stubborn Mining*		*Selfish Mining*	
Alpha	$\mathbf{MPU_{adv}}$	$\mathbf{MPU_{overall}}$	$\mathbf{MPU_{adv}}$	$\mathbf{MPU_{overall}}$
30%	0.33	0.58	0.42	0.33
50%	0.40	0.51	0.25	0.32
80%	0.25	0.20	0.13	0.14

As we can see here, in general, Stubborn Mining is performing better than Selfish Miner as discussed in the paper.