# Selfish Node

Constructor Argument (is a fast node?)

START

METHODS  (Other than inherited)

-Use static var node_id_counter for node_id assignment
-Use add_peer method to create p2p connections to the node
-sample selfish node id, connect it to \zeta fraction of nodes
-Sample cpu power from Gaussian distri
-Set cpu power of selfish node
-Initialise other state variables

Initialisation (inherit Node class, set is_selfish to 1, initialise own event queue)

Call receive block function to broadcast a block that is to be revealed from the private chain

Broadcast block

Expecting Method Call

Add Private Blocks

Add the blocks in private chain to the tree after they have been revealed, or at the end of the simulation

- Mine on private chain
- Initialise new block object
- set is_selfish_mined flag to 1
- Don't add any transactions except coinbase
-Schedule creation of blk according to exponential interarrival time
- If in 0', schedule broadcasting of block

Create Block

Receive Block

- Verify txns and validate blk if not own
- Forward the blk only if it's own
- If honest blk, then find the earlier lead
- if lead<=0, start mining on honest blk
- if lead=1, reveal the chain to reach 0'
- if lead=2, reveal the whole chain
- if lead>2, reveal one blk