Block Access Control in Wireless Blockchain Network: Design, Modelling and Analysis

Now-a-days everywhere wireless sensors are using for monitoring such as patient health monitoring, Road Traffic monitoring, area surveillance and many more. Wireless network senor sense their surrounding data and then send to single centralized server for monitoring and if this centralized server attacked or crashed then data will be lost and to overcome from this problem author of this paper employing Blockchain technology on WLAN (wireless local area network) as this Blockchain support decentralized data storage (means data stored at multiple nodes). If one node is attacked or crashed then data can be gather from other working nodes. Blockchain store or mine each data as transaction or block and associate each block with unique hash code and this hash code get verified before storing new block to avoid data tamper. This verification is called as POW (proof of work).

Due to above advantages author employing Blockchain in WLAN network but this will raise forking issue if one node sending data and Blockchain mining that data and at the same time if other node send data then data corruption will occur which will raise forking problem.

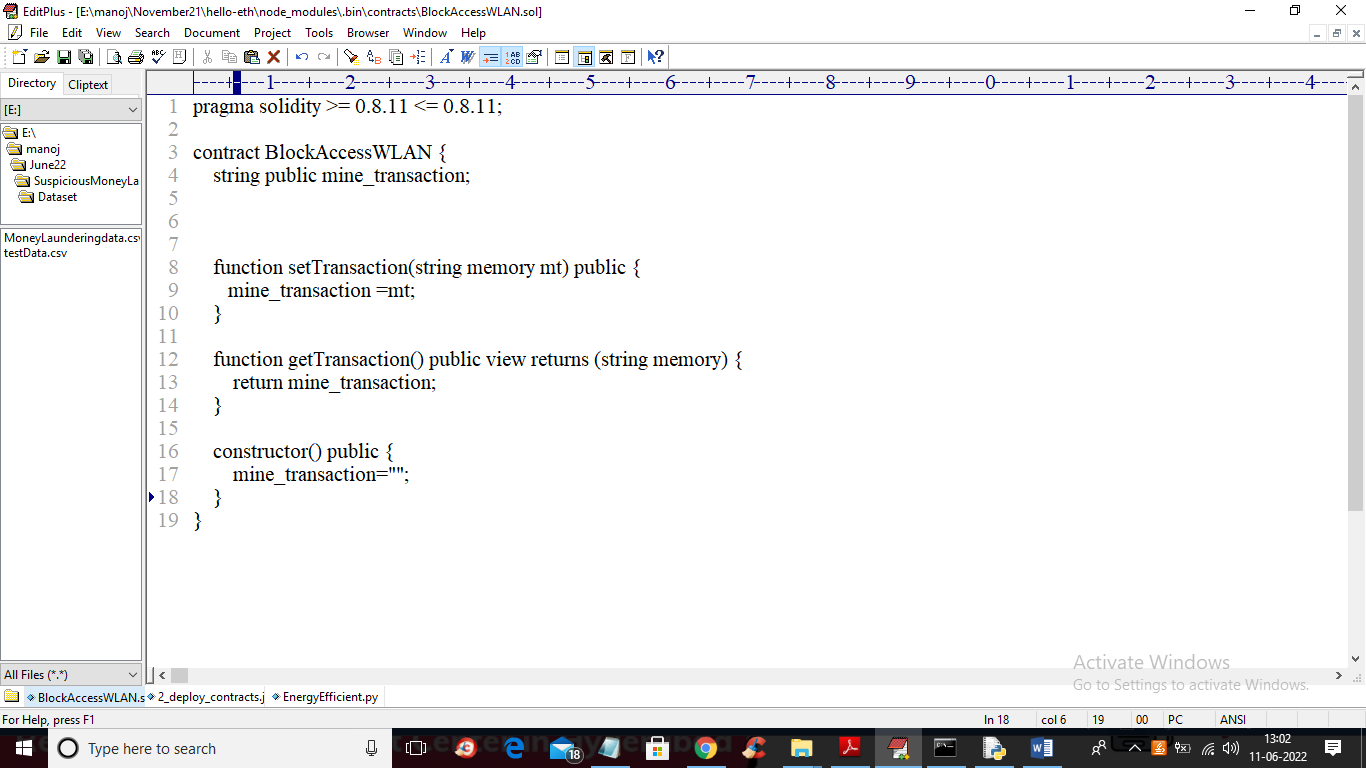
In propose paper author has consider 3 different types of nodes

1. Light Node (LN): this is a battery constrained (low computing power nodes) sensor nodes which sense data but cannot mine for Blockchain storage and this sensor will send sense data to Full Node
2. Full Node (FN): this node receive data from Light Node and then create Blockchain storage block as transaction mining and send to Access Point
3. Access Point: this node receive block from Full Node and then store in Blockchain and other Full nodes who don’t have data can download from Access Point.

To avoid forking problem author introduce 4 access blocks techniques such as BC1, BC2, BC3 and BC4 which ensures that if one node sending data then other node has to wait till the transmission completed. All nodes will follow strategy as No Block (data is not available), Block backoff (wait till other transmission completed) and TRANSMISSION (sending data). Below are the description of each block access

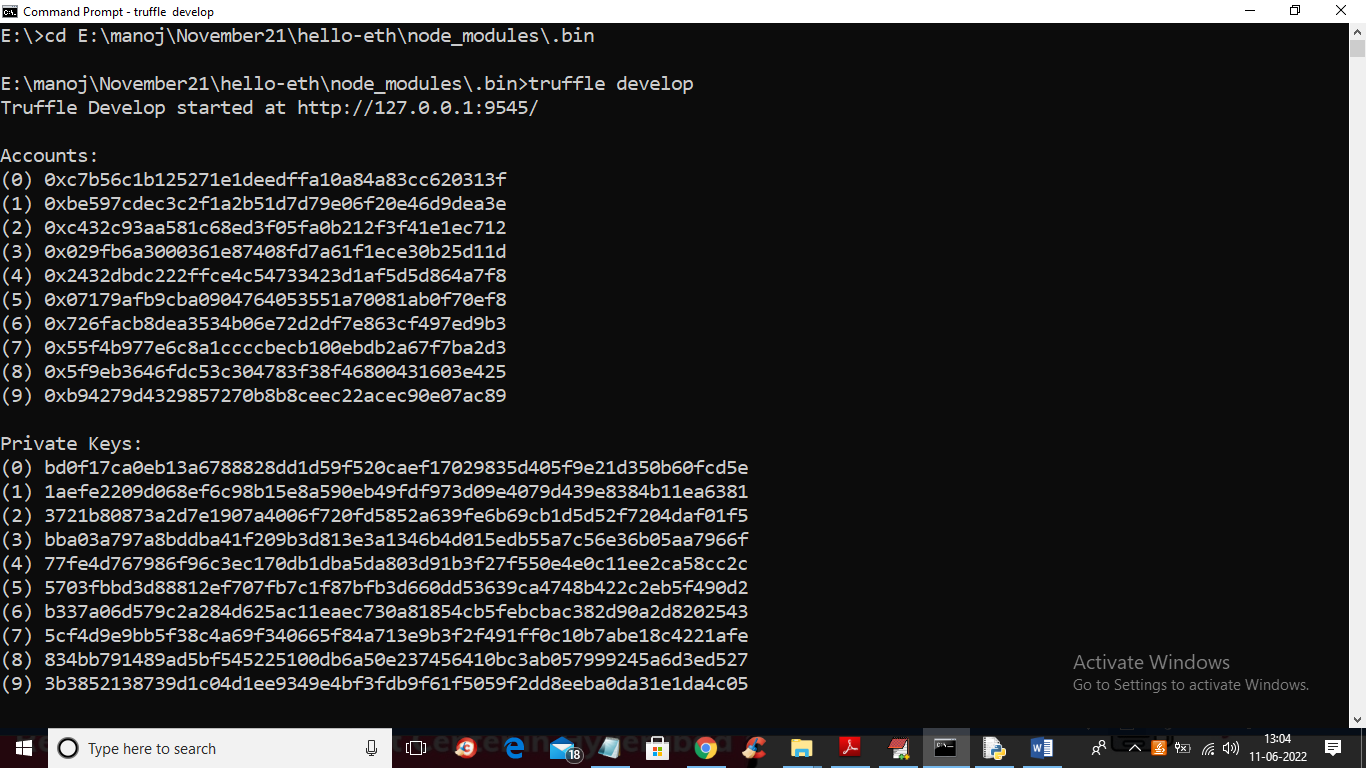
1. BC1 approach: only contains discard strategy, thus a FN will keep mining all the time. In no block state, if the FN generates a new block while a successful block transmission occurs on the channel, the FN discards its own block based on the discard strategy. If the new block is generated during the other information transmission, collision or channel idle time, the FN schedules its block transmission based on CSMA/CA. Once the channel remains idle more than a DIFS, the FN starts the backoff procedure by selecting a random initial value as the backoff counter.
2. BC2 approach: contains discard strategy and mining strategy I, thus the mining of a FN will be paused whenever a block transmission of the other FN is detected on the channel until the channel is idle more than a DIFS, which is shown in Fig. 2(b). Due to the mining pause, the expected mining time of FN in BAC-2 is less than that in BAC-1, which results in two differences between BAC-2 and BAC-1.
3. BC3 approach: contains discard strategy and mining strategy BC2, thus the mining of a FN will be paused when the FN has a new block until the new block is transmitted successfully or discarded. Since the mining is paused during the block backoff and block transmitting states, block queuing does not exist in BAC-3. So after a new block is transmitted successfully or discarded, a FN will return to no block state and restart mining. Except that BAC-3 has no block queuing, the other behaviours of BAC-3 is similar to BAC-1.
4. BAC-4: contains all the strategies. There are two differences between BAC-4 and BAC-1. The first is that, for a randomly chosen time slot, the probability to leave no block state in BAC-4 is lower than that in BAC-1. The second is that the block queuing does not exist in BAC-4, thus a FN will return to no block state and restart mining after the FN transmits or discards its block.

To implement above project we have used python TRUFFLE tool with ETHEREUM to store sensor data. We don’t have any sensors so we have designed simulation based application which generate random data and this data will stored in Blockchain. To store data in Blockchain we need to designed SOLIDITY SMART CONTRACT code which has to deploy in Ethereum and this contract will be called from python program to store or retrieve data. Below is the SMART CONTRACT code

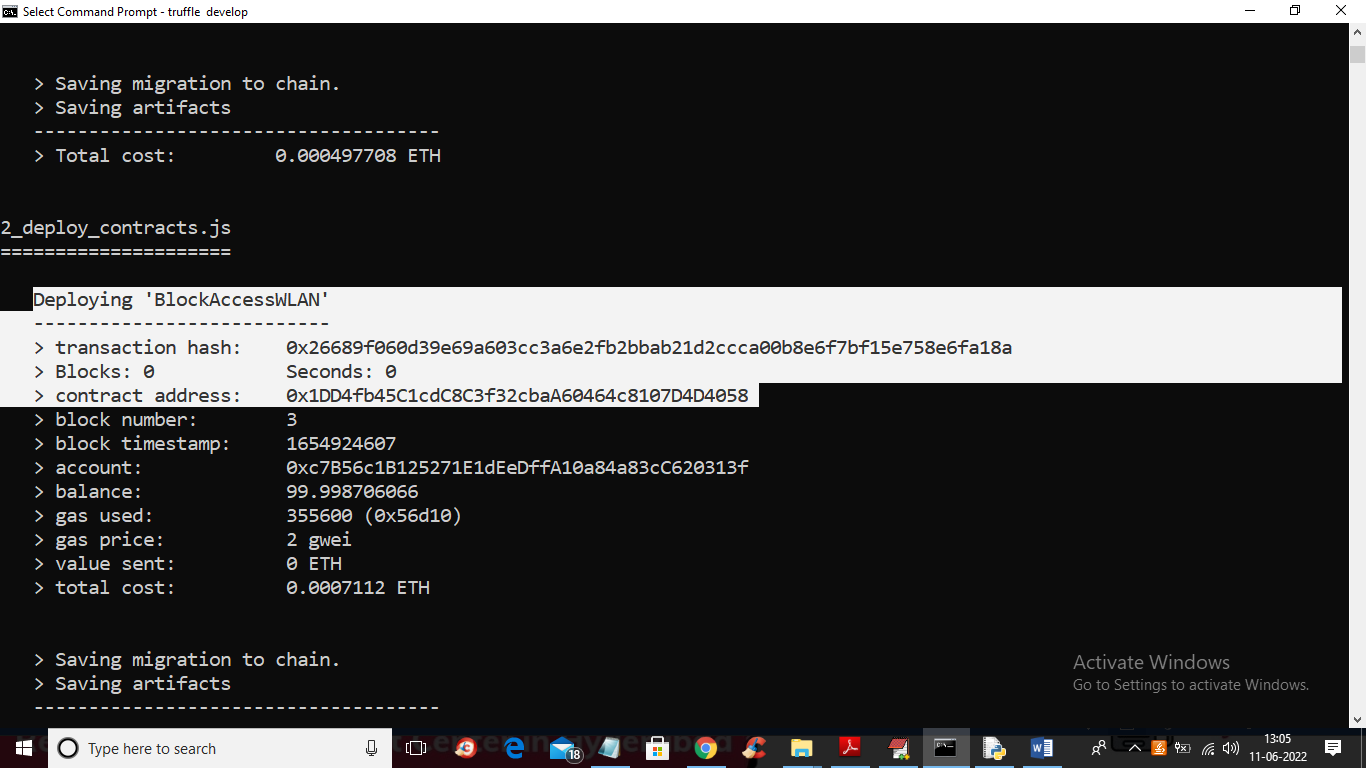


In above screen we have designed function to mine transaction by calling setting and getting method and to deploy above code in Ethereum we need to follow below steps

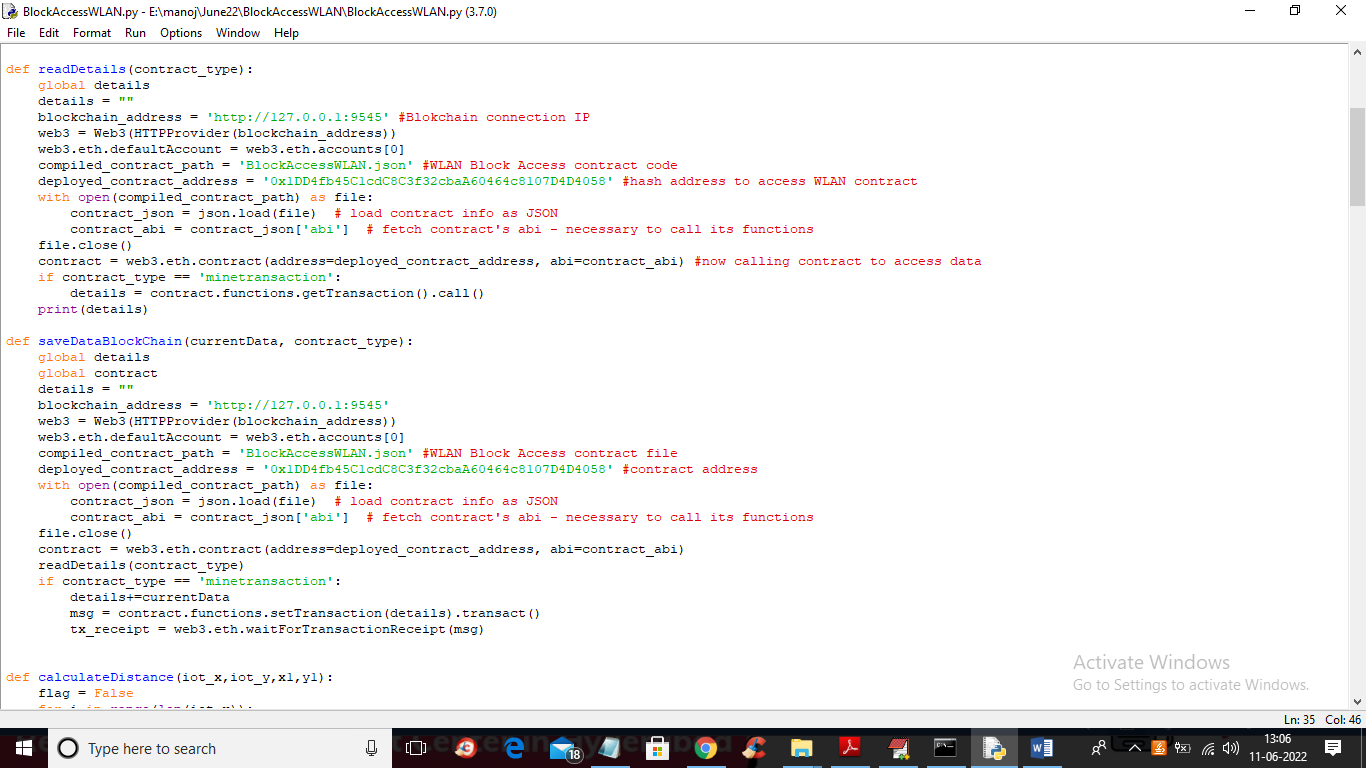
First go inside ‘hello-eth/node-modules/.bin’ folder and double click on ‘runBlockchain.bat’ file to get below screen



In above screen we can see Ethereum generated some default keys and accounts and now type ‘truffle migrate’ and press enter key to deploy contract and get below output



In above screen we can see ‘BlockAccessWLAN’ smart contract deployed and we got contract address and this address we can specify in python program to call that contract to store and retrieve data



In above python code you can see we have specified contract address and using that address python will store sensors data to Blockchain

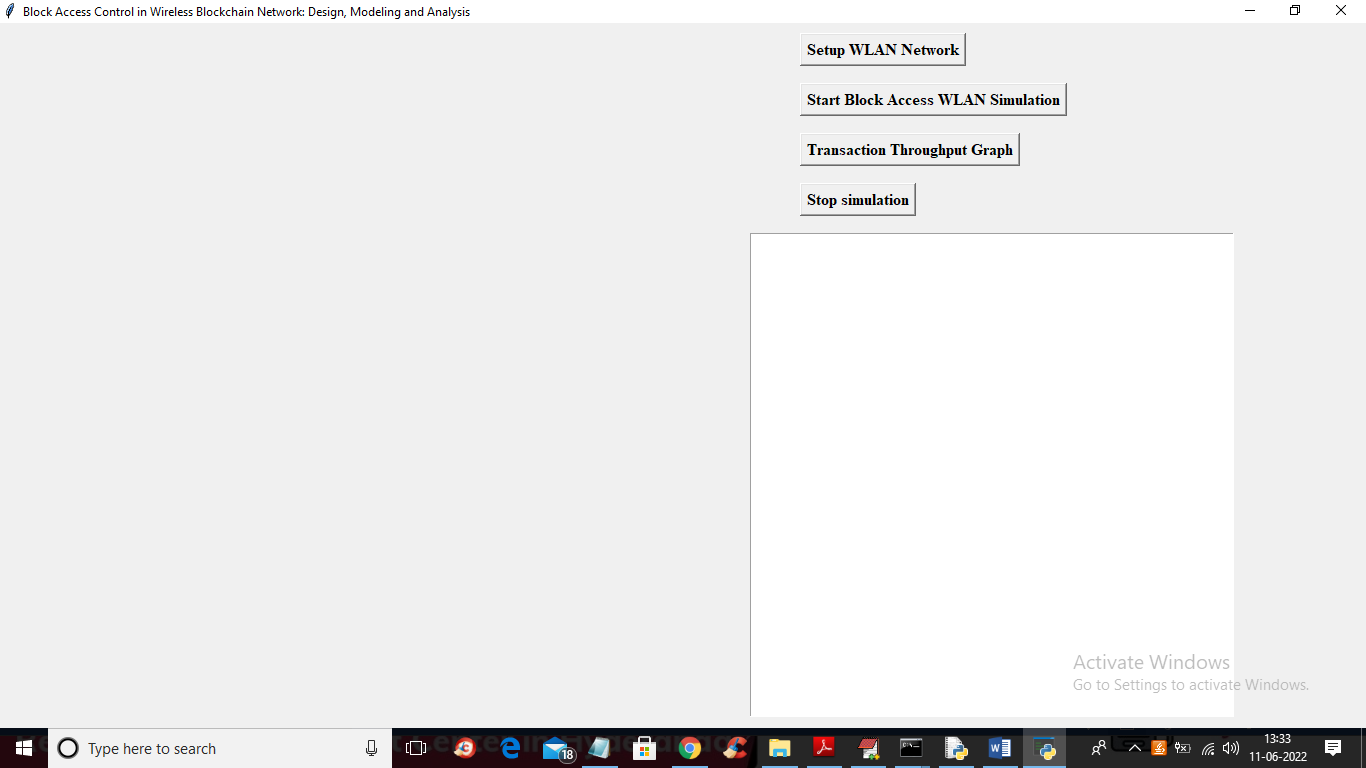
Modules Information

To implement this project we have designed following modules

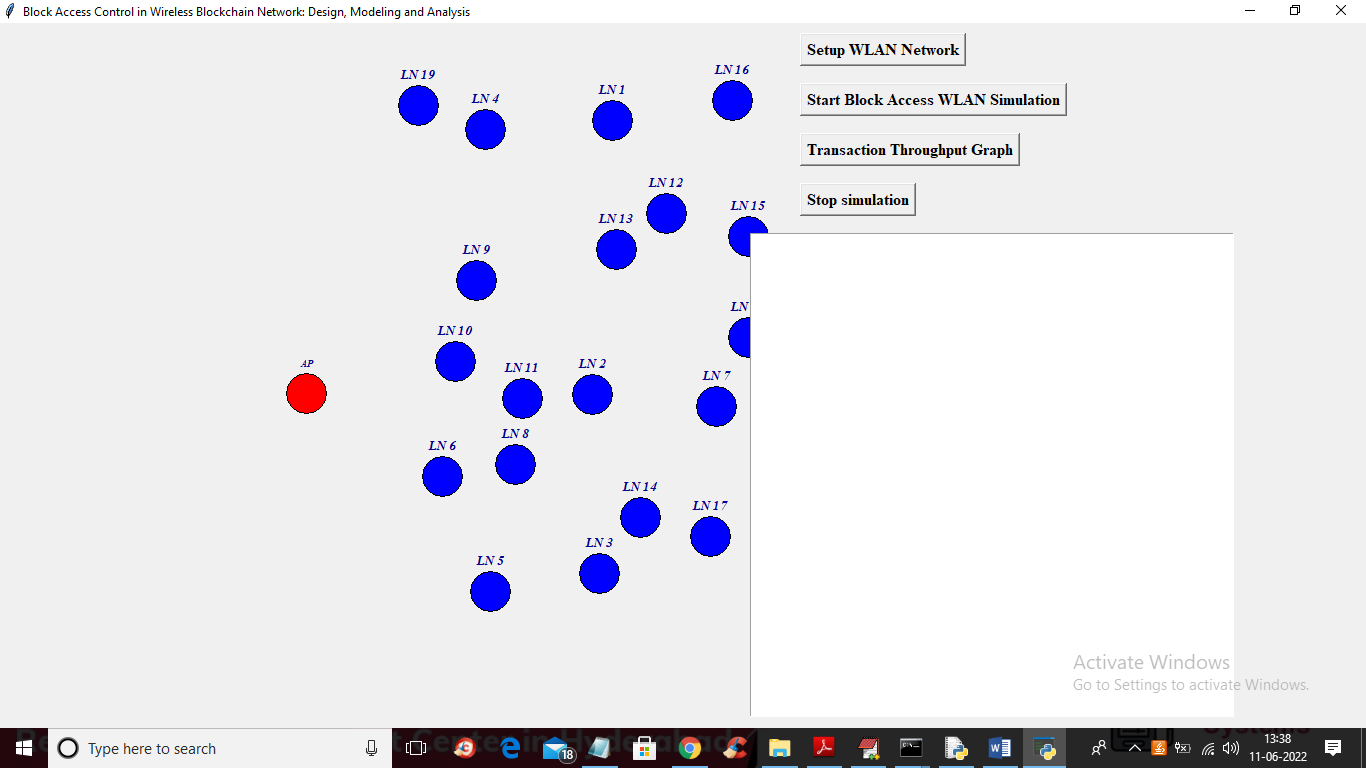
1. Setup WLAN Network: using this module we will generate network with dummy sensors placed at random location
2. Start Block Access WLAN Simulation: using this module each Light Node will sense random temperature and send to Full Node and full node will generate Block and send to Access Point for storage. Which sending data all BAC1, Bac2, Bac3 and Bac4 strategies will be applied to allow only one node to transfer data and other will wait till transfer complete
3. Transaction Throughput Graph: Each node will send data by using above strategies so by using this module we will calculate throughput for each strategy which refers to successful data transmission using any strategy.
4. Stop simulation: using this module we can stop simulation or sending data

SCREEN SHOTS

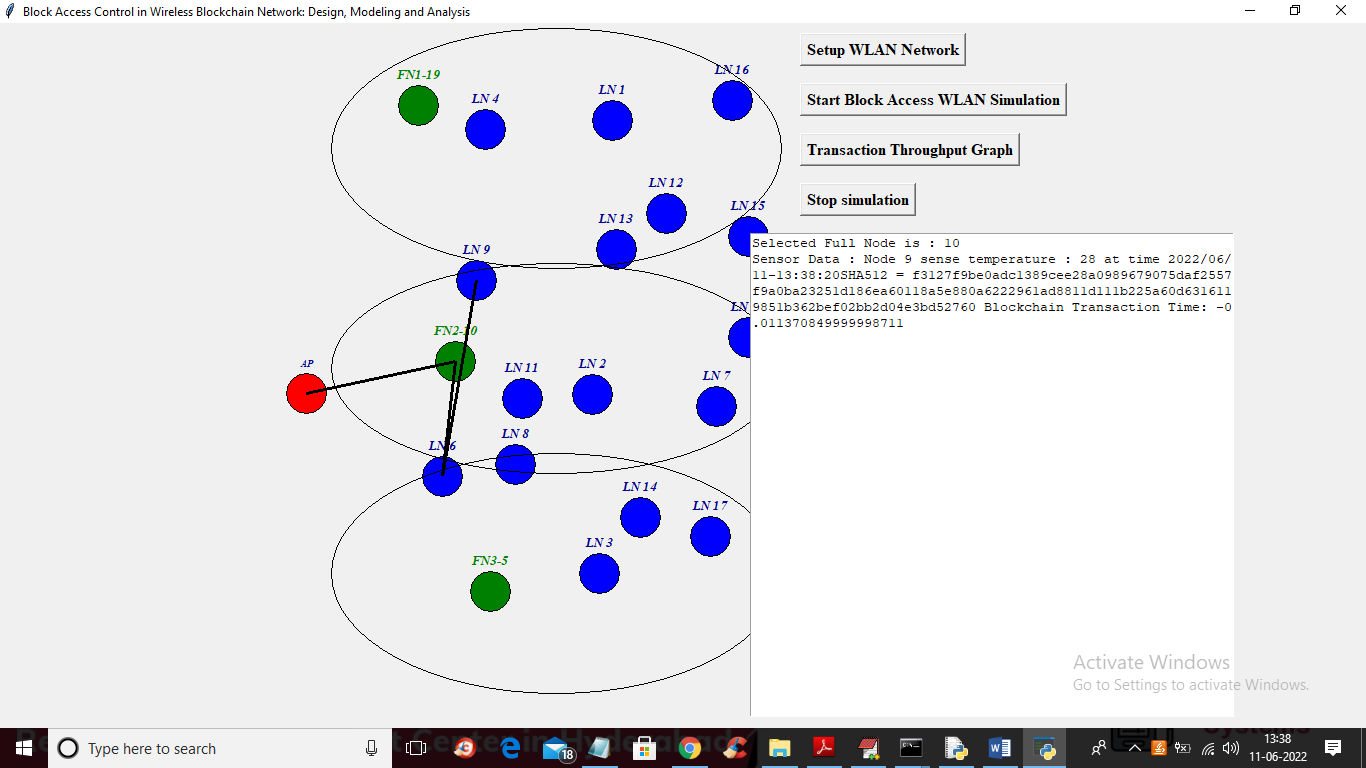
To run project double click on ‘run.bat’ file to get below screen

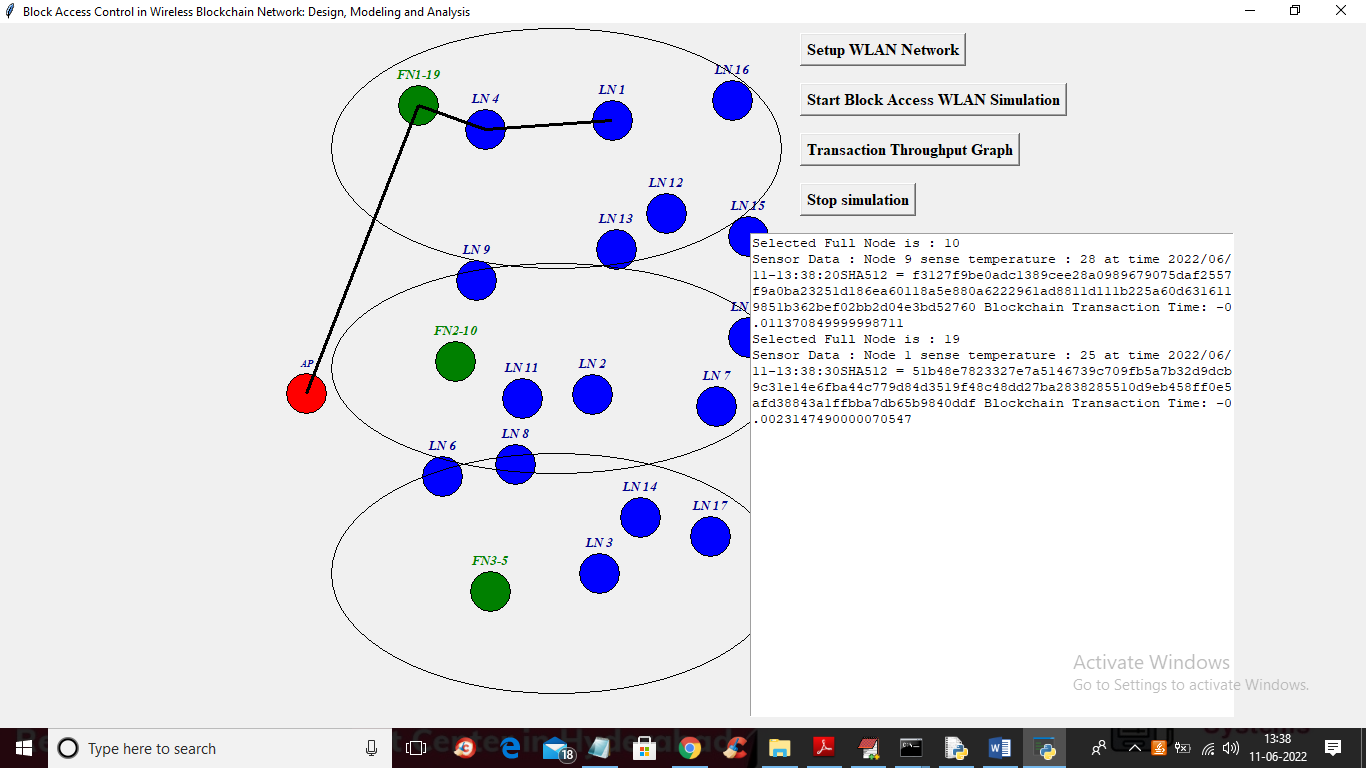


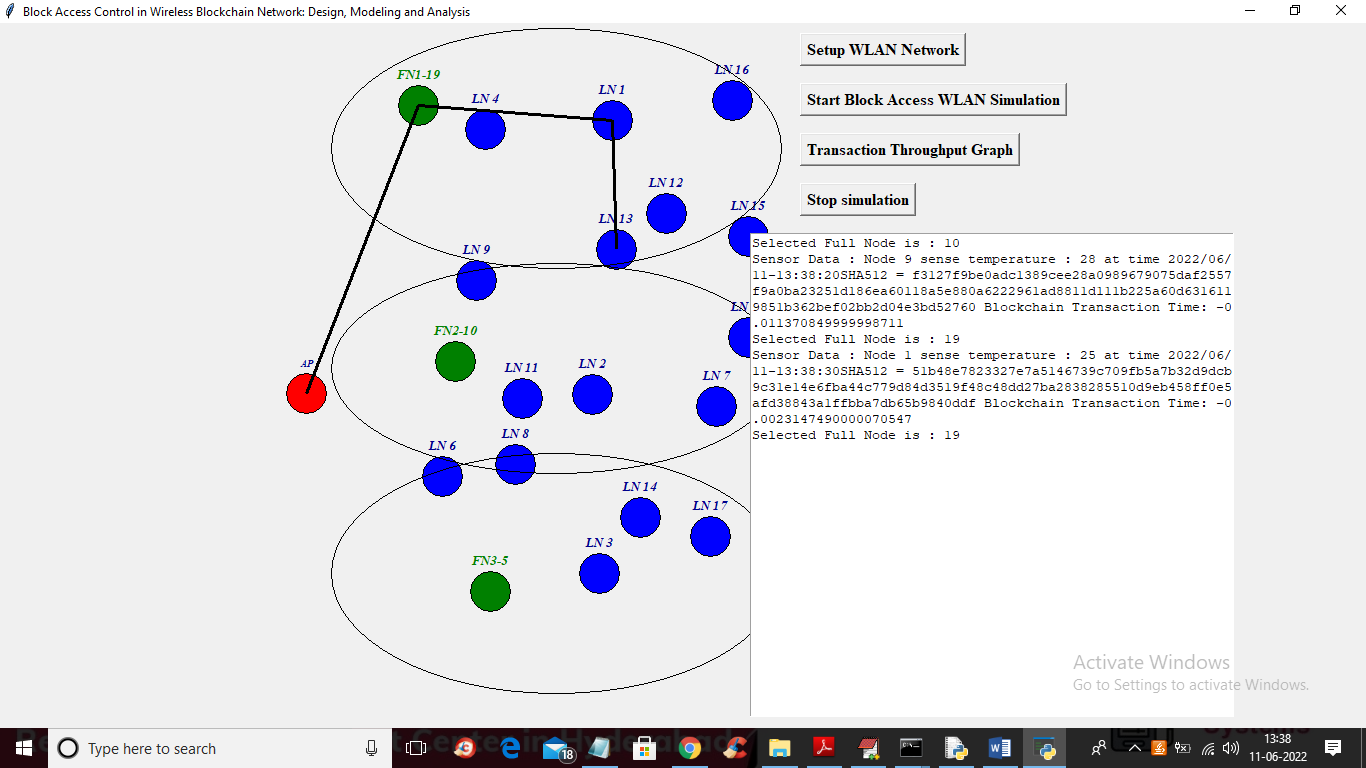
In above screen click on ‘Setup WLAN Network’ button to setup network and get below output



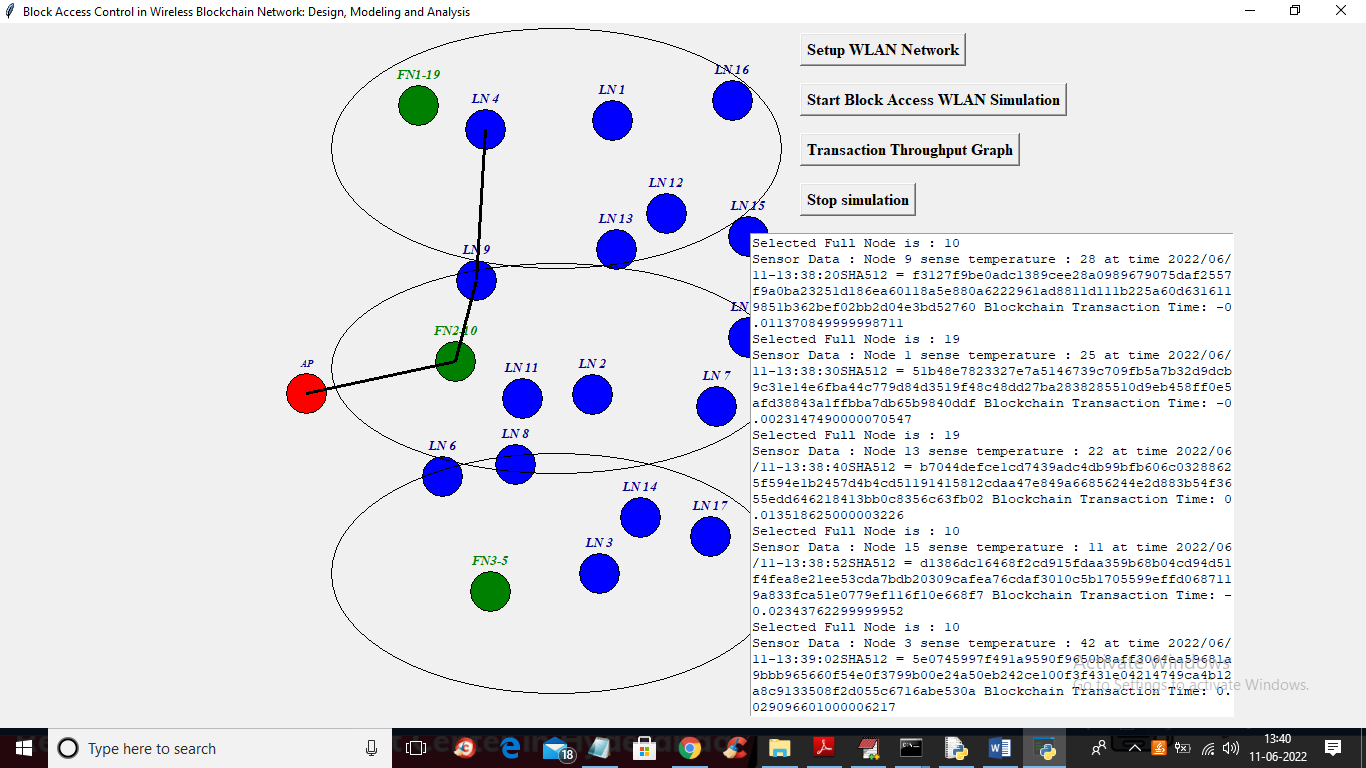
In above screen all blue colour circles consider as Light Node which send data to Full Node and the nodes which are nearer to AP red colour node is called as Full Node. Red colour circle is called as Access Point and all Light Node send data to Full Node and Full Node send to Access Point. Now click on ‘Start Block Access WLAN Simulation’ button to start sending data to access point



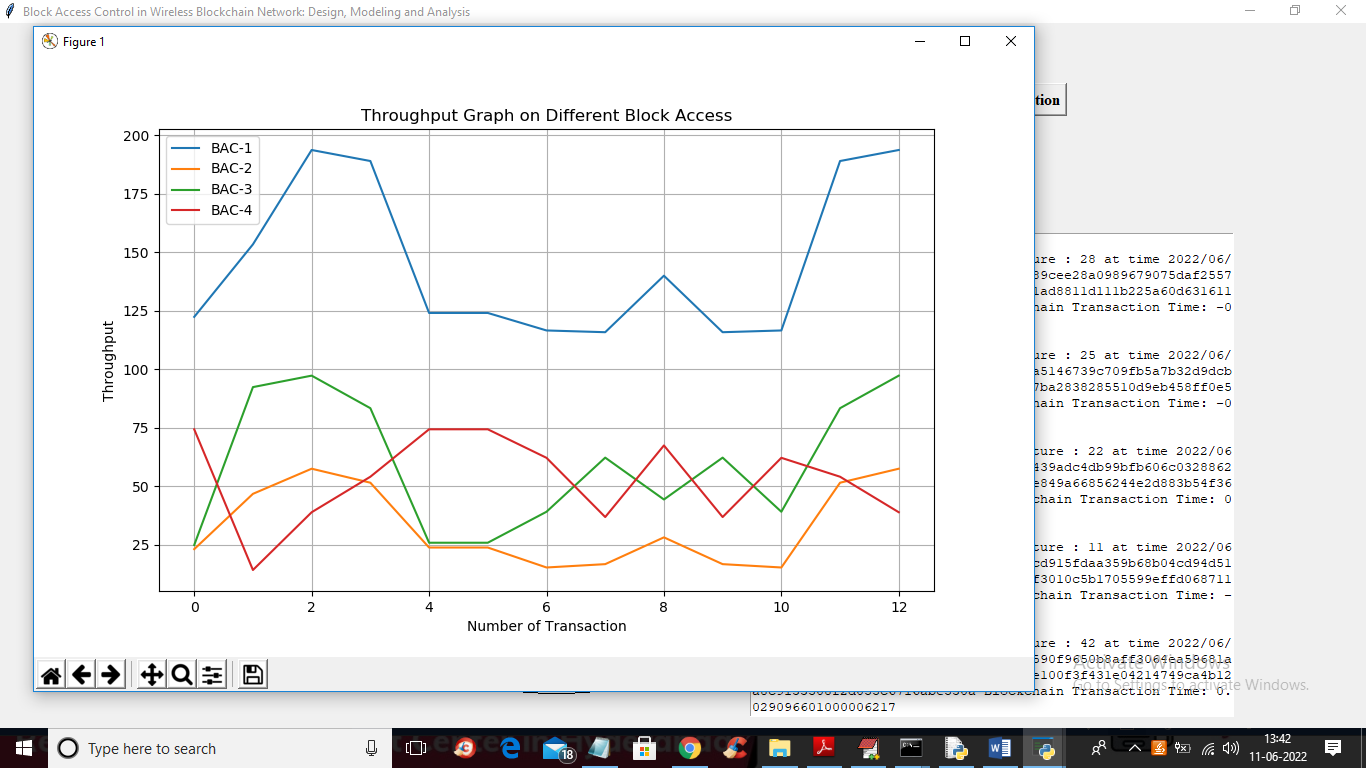




In above screen each blue light node will be randomly selected as source and it will sense some random data and send to green colour FULL Node and Full Node send to access point and in above screen we can see data sending via black colour line and if one node sending data then other nodes will wait and continue only after first one completes and this simulation will run in infinite loop and to stop simulation click on ‘Stop Simulation’ button.



In above screen in text area we can see which node is sending what data and which node is selected as FN. Now click on ‘Transaction Throughput Graph’ button to get below output



In above graph x-axis represents number of transactions and y-axis represents throughput of sending that transaction data. Throughput refers to amount of data send from start to end tie. Blue line represents BAC1 strategy throughput and similarly different lines represents different strategy throughput. In all techniques BAC-1 got high throughput