**# MULTI HOP NETWORK IMPLEMENTATION**

**## PROBLEM STATEMENT**

Client/Server program to exchange data packets across the multi-hop network. Implemented in python using socket library.

The attributes of the packet are as follows.

(i) Client Id (Host Name/Any Name which defines the identification of machine from where a

network packets will be shared over the network)

(ii) Client IP Address (IP address of the source machine if single machine is used to test the

program then loopback (127.0.0.1) address may be used

(iii) Destination IP Address (IP address of the destination machine if single machine is used to

test the program then loopback (127.0.0.1) address may be used

(iv) Payload of Packet (How many bytes of information a packet will carry. This parameter will

be variable and a user can define at the runtime of the program)

(v) Total Number of Packets for the message (Divide the message size with payload and find the

number of packet for the message. If division is perfect otherwise payload of last packet will

be different. Message will be of any kind- it may be executable/image/text file, etc.)

(vi) Current Packet Id (It shows the packet number out of the total packets)

(vii) Name of the message (What kind of name will be given to message using which it will be

delivered at the recipient side. If it is a file then name of the file with its extension.)

(viii) Security Certificate.

**# NOTE: THE MACHINES MUST BE ON THE SAME NETWORK FOR THE PROGRAM TO WORK**

**## SOLUTION**

ABSTRACT WORKING OF THE PROGRAM

The program runs on 4 machines/nodes the data(an image named 'the matrix has you.png' in our case) is first

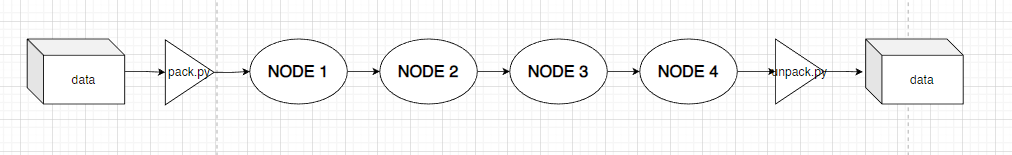
converted into packets which are objects of PacketClass dumpled as pickle files. These files can later be

unpickled and reconverted back to original data.

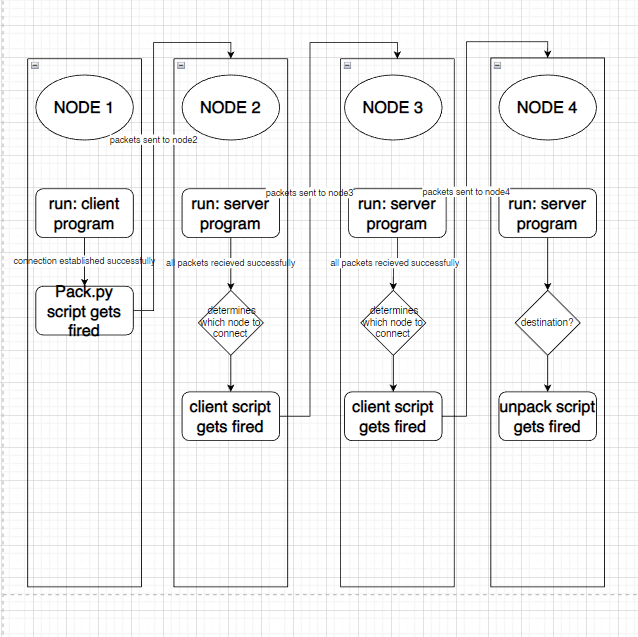
the packets travel through machine/node 2,3 and finally reconverted to original data in machine 4.

**### The program uses python and socket programming to implement multi hop network. Sokect programming itself implements**

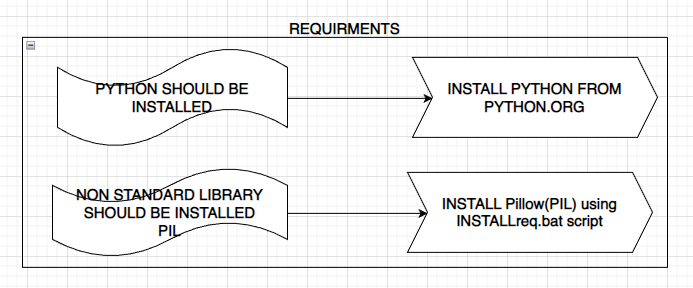
TCP/IP protocol to implement the network.



**### FLOWCHART**



**## REQUIRMENTS**



**## HOW TO USE THE PROGRAM**

**### FIRST MAKE A FOLDER NAMED 'packets',  IN EACH OF THE DIRECTORIES**

STEP 1: OPEN THE SERVER ON MACHINE 4, `serverm4.bat` and note down the listening as ip adress/ip adress of macihne 4

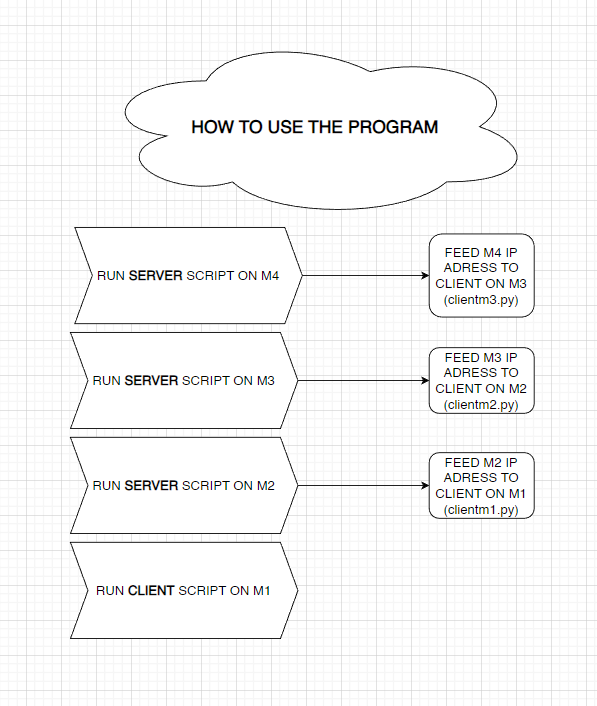
STEP 2: OPEN CLIENT ON MACHINE 3, LOOK FOR HOST VARIABLE AND UPDATE THAT VALUE WITH THE IP ADRESS FOUND ABOVE

STEP 3: OPEN THE SERVER ON MACHINE 3, `serverm3.bat` and note down the listening as ip adress/ip adress of macihne 3

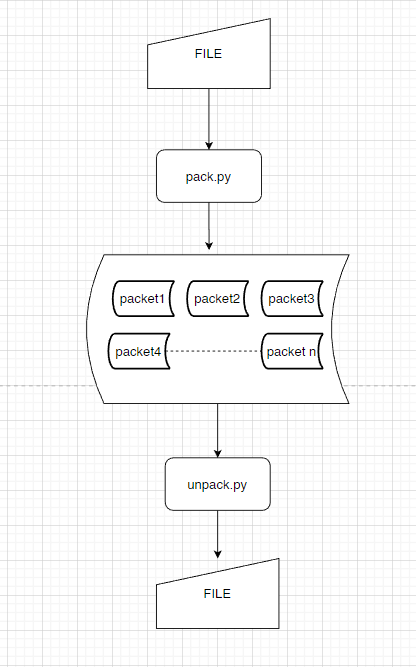
STEP 4: OPEN CLIENT ON MACHINE 2, LOOK FOR HOST VARIABLE AND UPDATE THAT VALUE WITH THE IP ADRESS FOUND ABOVE

STEP 5: OPEN THE SERVER ON MACHINE 2, `serverm2.bat` and note down the listening as ip adress/ip adress of macihne 2

STEP 6: OPEN CLIENT ON MACHINE 1, LOOK FOR HOST VARIABLE AND UPDATE THAT VALUE WITH THE IP ADRESS FOUND ABOVE



**## FUNCTIONING OF PACK AND UNPACK FUNCTIONS**



**## Flow Control**

When a data frame is sent from one host to another over a single medium, it is required that the sender and receiver should work at the same speed. That is, sender sends at a speed on which the receiver can process and accept the data. What if the speed (hardware/software) of the sender or receiver differs? If sender is sending too fast the receiver may be overloaded, (swamped) and data may be lost.

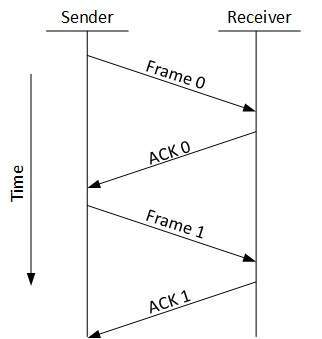
Two types of mechanisms can be deployed to control the flow:

Stop and Wait

This flow control mechanism forces the sender after transmitting a data frame to stop and wait until the acknowledgement of the data-frame sent is received.

**## WE HAVE IMPLEMENTED STOP AND WAIT FOR FLOW CONTROL**

**## THERE IS NO ERROR DETECTION AND RECOVERY MECHANISM**



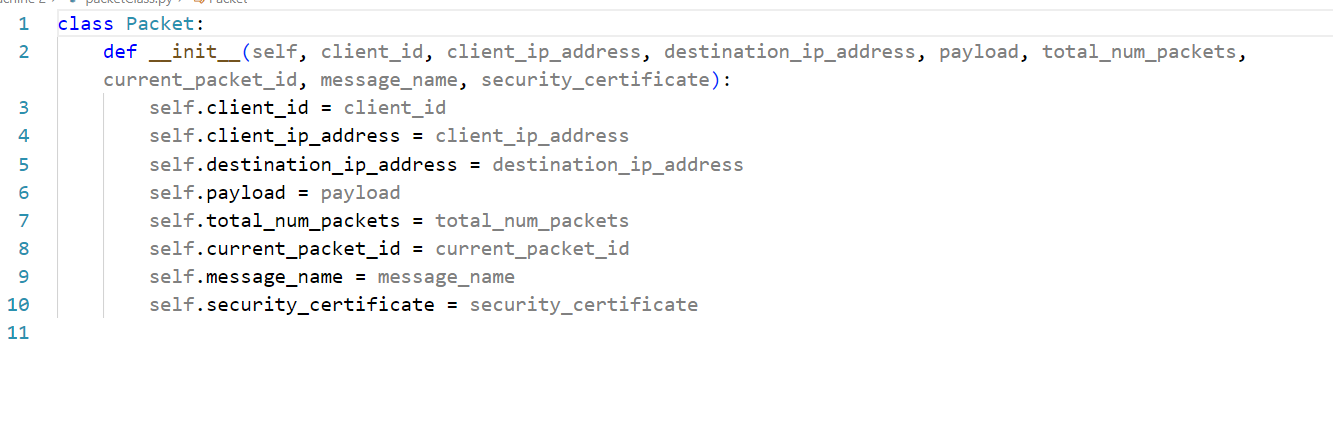
**### IT MAY BE POSSIBLE THAT THE SENDER AND RECIEVER WORK AT DIFFERENT SPEEDS, BUT IN OUR IMPLEMENTATION WE HAVE USED clientsocket.recv()**

**### If there is no data available to receive, then clientsocket.recv() will block the program execution until some data is received or**

**### the connection is closed. WHICH MEANS THE PROGRAM IS GURANTEED TO EXECUTE IRRESPECTIVE OF THE TIME IT WILL TAKE FOR THE PACKETS TO LOAD**

**### FROM THE SERVER END..**

**## PACKET CLASS**



THE PROGRAM USES VARIOUS BATCH SCRIPTS TO TRIGGER PYTHON PROGRAMS. WE ALSO FIRE SCRIPTS FROM PYTHON CODE

FINALLY ON THE MACHINE 4 THE PACKETS ARE UNPACKED AND CONVERTED INTO ORIGINAL IMAGE AND THEN OPENED