COMPUTER COMMUNICATION NETWORKS

LAB EXPERIMENT 14

NAME: RAHIL SHARMA
PRN: 18070123062
BATCH: 2018-2022
DIVISION: G2; EA 3

AIM: To code a Sender Receiver Program for Go Back N ARQ implementation.

Code of the Program:

Sender:

```
import time,socket,sys
import random

print("\n Welocme to receiver module \n")
print("initializing.....\n")
time.sleep(1)

s= socket.socket()
shost=socket.gethostname()
ip=socket.gethostbyname(shost)
print(shost, "(", ip, ")\n")
host= input(str("enter server address: "))
port=1232
print("\n trying to connect to ", host, "(", port, ")\n")
time.sleep(1)
```

```
s.connect((host,port))
print("connected....\n")

while True:

   m=s.recv(1024)
   m=m.decode()
   if m != "bye":
      k=s.recv(1024)
      k=k.decode()
      k=int(k)
      i=0
      a=""
      b=""
      f=random.randint(0,1)
      message=""
      while i != k:

         f= random.randint(0, 1)
         if(f==0):
            b="ACK LOST"
            message=s.recv(1024)
            message=message.decode()
            s.send(b.encode())

         elif (f==1):
            b="ACK" + str(i)
            message=s.recv(1024)
            message=message.decode()

            s.send(b.encode())
            a=a + message
            i= i+1
      print("Received Message: ", m)
   else:
      print("\n Conection Terminated")
```

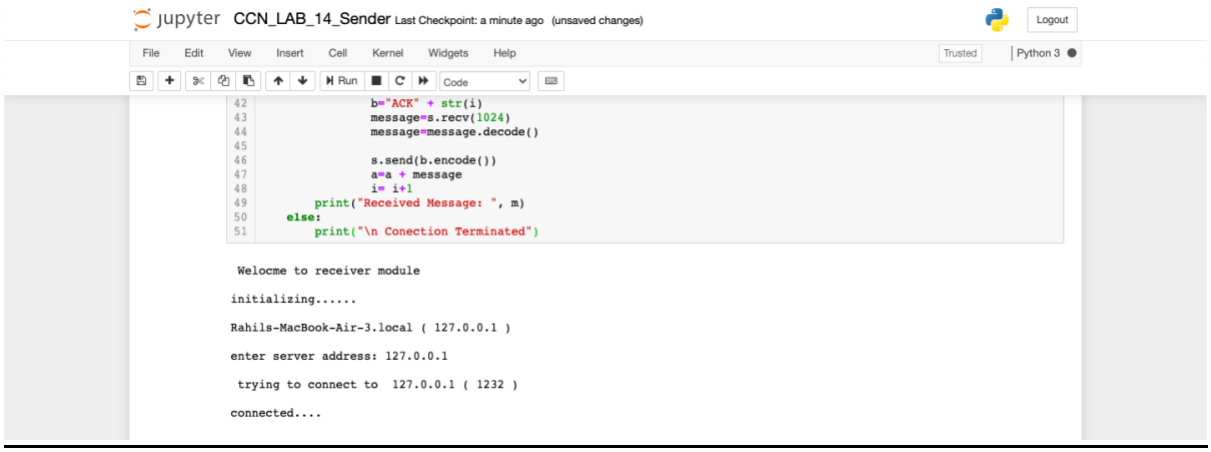
Receiver:

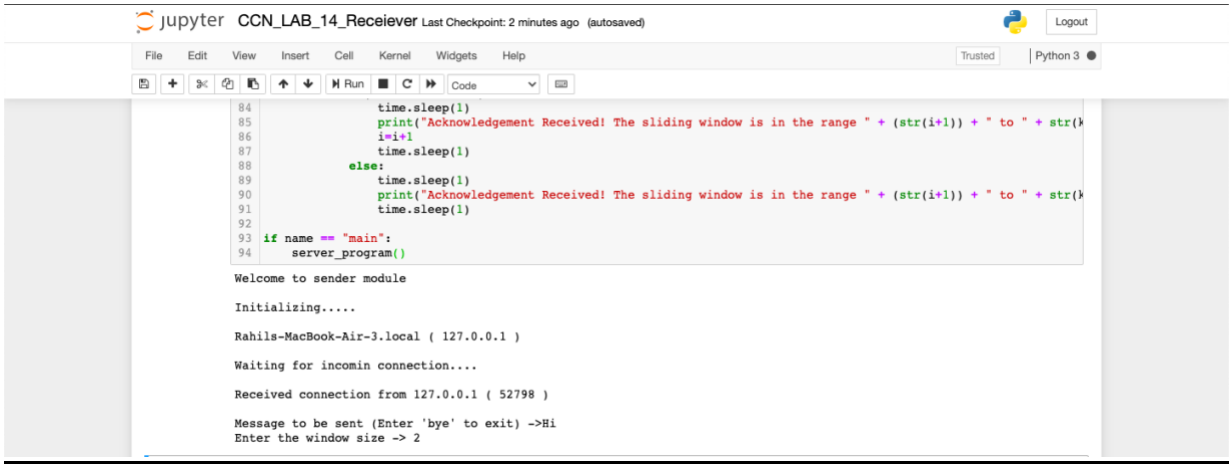
```
import time,socket,sys

def decimalTobinary(n):
   return n.replace("0b", "")

def binarycode(s):
   a_byte_array=bytearray(s,'utf8')

   byte_list=[]
```

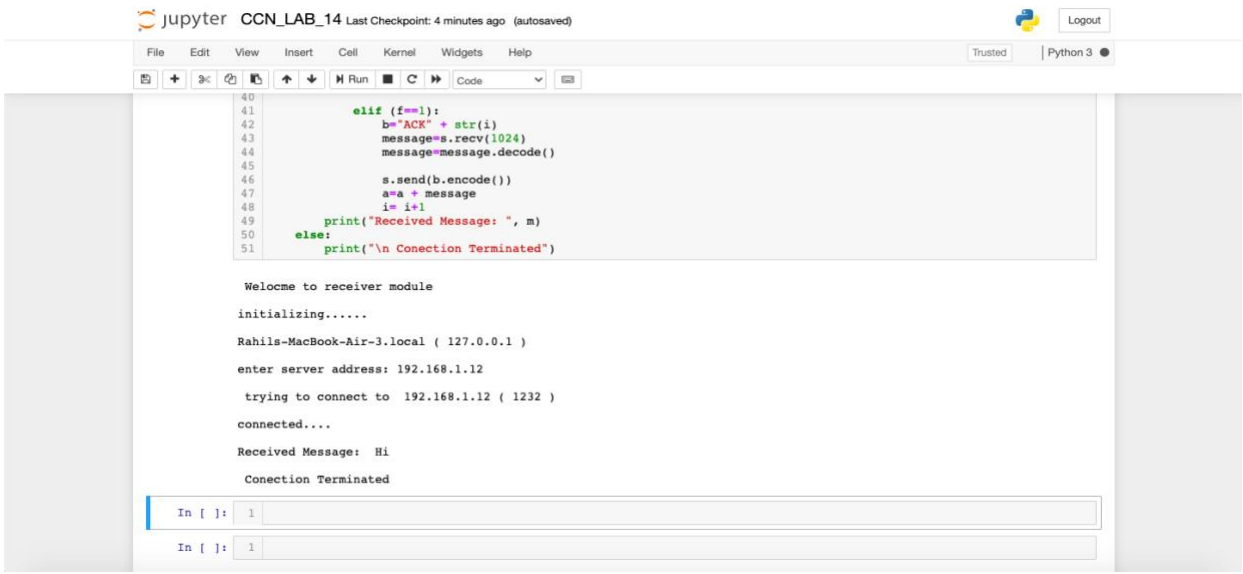
```
    for byte in a_byte_array:
        binary_representation=bin(byte)
        byte_list.append(decimalTobinary(binary_representation))

        a=""
        for i in byte_list:
            a=a+i
        return a

print("Welcome to sender module \n")
print("Initializing.....\n")
time.sleep(1)

s=socket.socket()
host= socket.gethostname()
ip=socket.gethostbyname(host)
port=1232
s.bind((host,port))
print(host, "(", ip, ")\n")


s.listen(1)
print("Waiting for incomin connection....\n")

conn,addr=s.accept()
print("Received connection from", addr[0], "(", addr[1], ")\n")

while True:
    message= input(str("Message to be sent (Enter 'bye' to exit) ->"))
    conn.send(message.encode())
    if message == "bye":
        message="connection terminated"
        print("\n Connection terminated")
        conn.send(message.encode())
        print("\n")
        break
    message = binarycode(message)
    f=str(len(message))
    conn.send(f.encode())

    i=0
    j=0
    j=int(input("Enter the window size -> "))

    b=""

    j=j-1
    f=int(f)
```

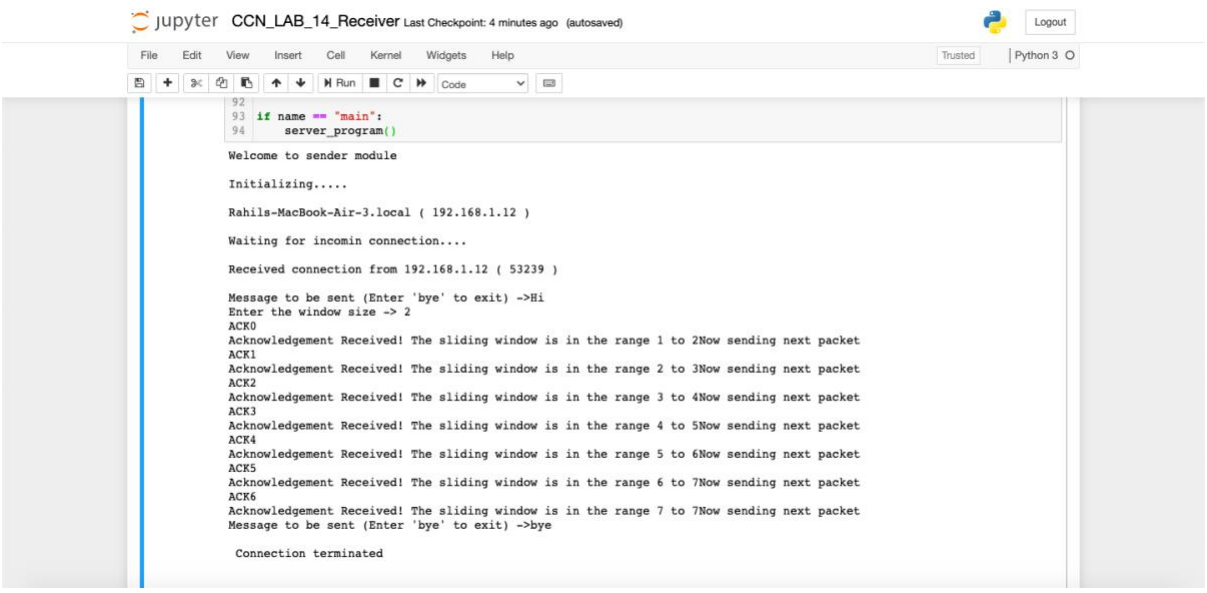
```
   k=j
   while i !=f:
      while(i != (f-j)):
         conn.send(message[i].encode())
         b=conn.recv(1024)
         b=b.decode()
         print(b)
         if(b != "ACK LOST"):
            time.sleep(1)
            print("Acknowledgement Received! The sliding window is in the range " + (str(i+1)) + " to " +
str(k+1) + "Now sending next packet")
            i =i + 1
            k=k + 1

            time.sleep(1)
         else:
            time.sleep(1)
            print("Acknowledgement Received! The sliding window is in the range " + (str(i+1)) + " to " +
str(k+1) + "Now Resending next packet")
            time.sleep(1)
      while(i !=f):

         conn.send(message[i].encode())
         b=conn.recv(1024)
         b=b.decode()
         print(b)
         if(b != "ACK LOST"):
            time.sleep(1)
            print("Acknowledgement Received! The sliding window is in the range " + (str(i+1)) + " to " +
str(k) + "Now sending next packet")
            i=i+1
            time.sleep(1)
         else:
            time.sleep(1)
            print("Acknowledgement Received! The sliding window is in the range " + (str(i+1)) + " to " +
str(k) + "Now Resending same packet")
            time.sleep(1)

if name == "main":
   server_program()
```

OUTPUTS AND SCREENSHOTS:









CONCLUSION: From this experiment we have an understanding of a Sender Receiver Program for Go Back N ARQ implementation.