GOA COLLEGE OF ENGINEERING

Experiment No: 4

ALOHA

Aim: To write and execute a program implementing ALOHA

Theory:

ALOHA is a multiple access protocol for transmission of data via a shared network channel. It operates in the medium access control sublayer (MAC sublayer) of the open systems interconnection (OSI) model. Using this protocol, several data streams originating from multiple nodes are transferred through a multi-point transmission channel.

In ALOHA, each node or station transmits a frame without trying to detect whether the transmission channel is idle or busy. If the channel is idle, then the frames will be successfully transmitted. If two frames attempt to occupy the channel simultaneously, collision of frames will occur and the frames will be discarded. These stations may choose to retransmit the corrupted frames repeatedly until successful transmission occurs.

There are 2 major versions of ALOHA namely, Pure ALOHA and Slotted ALOHA

Pure ALOHA

In pure ALOHA, the time of transmission is continuous. Whenever a station has an available frame, it sends the frame. If there is collision and the frame is destroyed, the sender waits for a random amount of time before retransmitting it.

Slotted ALOHA

Slotted ALOHA reduces the number of collisions and doubles the capacity of pure ALOHA. The shared channel is divided into a number of discrete time intervals called slots. A station can transmit only at the beginning of each slot. However, there can still be collisions if more than one station tries to transmit at the beginning of the same time slot.

Code:

```python
import matplotlib.pyplot as plt
import numpy as np
color=["red","blue","orange","green"]

# Declaring a figure "gnt"
fig, gnt = plt.subplots()
gnt.set_yticklabels([])
maxTime=-1
numSenders=int(input("Enter number of senders: "))
print("Enter data in format TIME LENGTH")
```

```python
base=10
senders={}
for senderNum in range(numSenders):

    sender=chr(ord("A")+senderNum)
    senderSchedule=[]
    print("For sender number",sender,": ")
    while True:

        ip=input()
        if ip:

            ip=[int(x.strip()) for x in ip.split(" ")]
            senderSchedule.append((ip[0],ip[0]+ip[1]))
            maxTime=max(maxTime,ip[0]+ip[1])

else: break

    senders[sender]=senderSchedule
    gnt.broken_barh(senderSchedule,(base,10),facecolors=('tab:'+color[senderNum]))
    base+=10

def checkOverlap(senders):
    globalSch=np.zeros(maxTime+1)
    for sender in senders:

        schedule=senders[sender]
        for slot in schedule:

            if np.any(globalSch[slot[0]:slot[1]+1]):
                print("Overlap exists")
                return

            else:
                globalSch[slot[0]:slot[1]+1]=1

    print("No overlap")
checkOverlap(senders)
plt.savefig("gantt1.png")
```

Output:

Enter number of senders: 2
Enter data in format TIME LENGTH For sender number A :
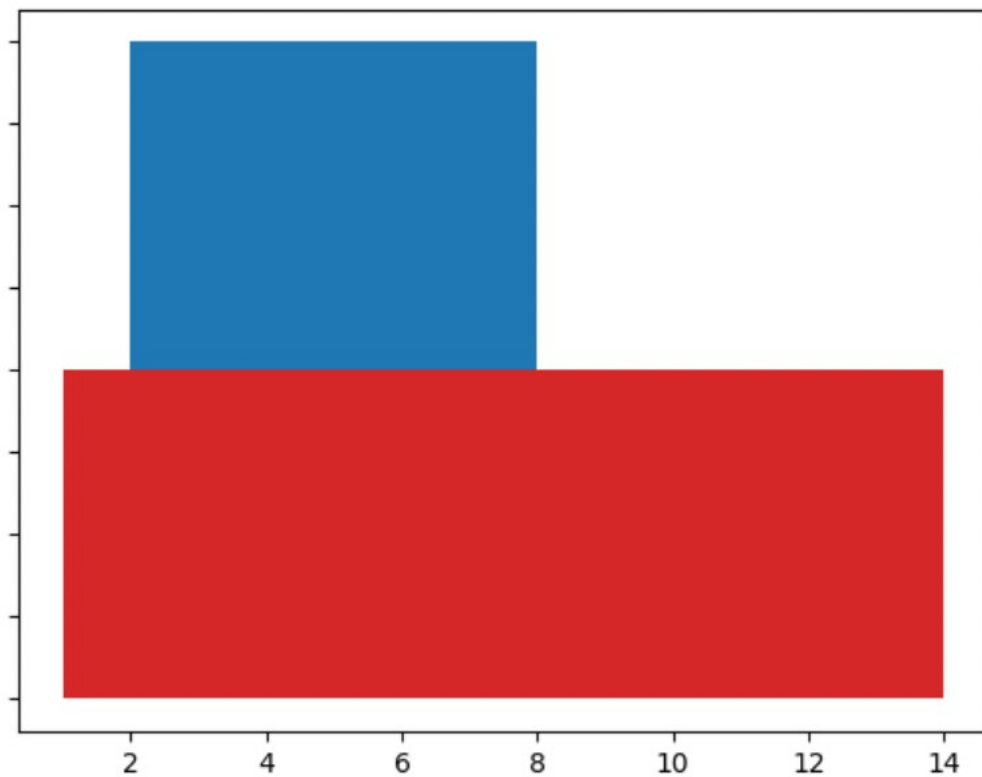1 2

5 4

For sender number B :

3 2
6 3

Overlap exists

Graph Output:

Conclusion: A program to implement ALOHA was successfully written and executed.

Deepraj Bhosale Roll Number: 181105016 Batch-A Semester VIII