NETWORK LAB REPORT

NAME: ANURAN CHAKRABORTY
ROLL NO.: 20
CLASS: BCSE-III
SECTION: A1

ASSIGNMENT NUMBER: 4

PROBLEM STATEMENT:
Implement CDMA for multiple access of a common channel by n stations. Each sender uses a unique code word, given by the Walsh set, to encode its data, send it across the channel, and then perfectly reconstruct the data at n stations.

DEADLINE: 14TH MARCH, 2019
SUBMITTED ON: 14TH MARCH, 2019
REPORT SUBMITTED ON: 28TH MARCH, 2019

CDMA:

DESIGN

The program implements CDMA algorithm for multiple access. The program consists of one module.

1. **cdma.py**
   For denoting a station, a class is created. Every station has a station number, its corresponding Walsh code, and its corresponding data. For every station an object of station class is created which sends the data. The station class contains a sender method which takes the current bit to be sent, multiplies it with the corresponding Walsh code and returns the data. For every station, for every bit the data is sent, added in the channel and then decoded.

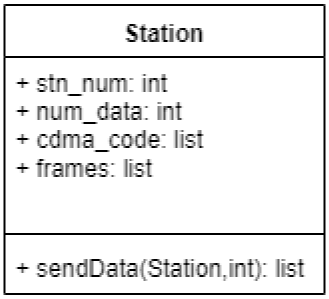


Fig 1 Class diagram for the station class

Some important parameters for the design of the program are:

Frame format: The frames are sent bit by bit.

Input format: The input for the program is a text file consisting of a string of only 0s, 1s and ‘i’s. ‘i’ denotes that the station is idle at that point of time.

Output format: The program output simulates the CDMA protocol.

IMPLEMENTATION

The assignment has been implemented in Python3. The detailed description is given below.

cdma.py:

This module simulates the whole process.

First the station class is defined as described in the design.

```
class Station: # Class for denoting every station

        def __init__(self, stn_num, num_data, cdma_code, frames):
                self.stn_num=stn_num
                self.num_data=num_data
                self.cdma_code=cdma_code
                self.frames=frames

        # Send data corresponding to index of frame list
        def sendData(self,index):

                if(self.frames[index]=='i'): # Meaning idle
                        data=0
                elif(self.frames[index]=='0'):
                        data=-1
                else:
                        data=1

                codeword=[data*self.cdma_code[i] for i in range(len(self.cdma_code))]
                print('Station '+str(self.stn_num)+' sending '+str(codeword))

                return codeword
```

createWalsh(r):

Function to create Walsh table given the number of stations.

```
# Function to create walsh tables
def createWalsh(r):
        global walsh
        walsh=[[int(bin(x&y),13)%2 or -1 for x in range(r)]for y in range(r)]
```

decode_cdma(codeword, num_stn):

Function to decode the data from all stations

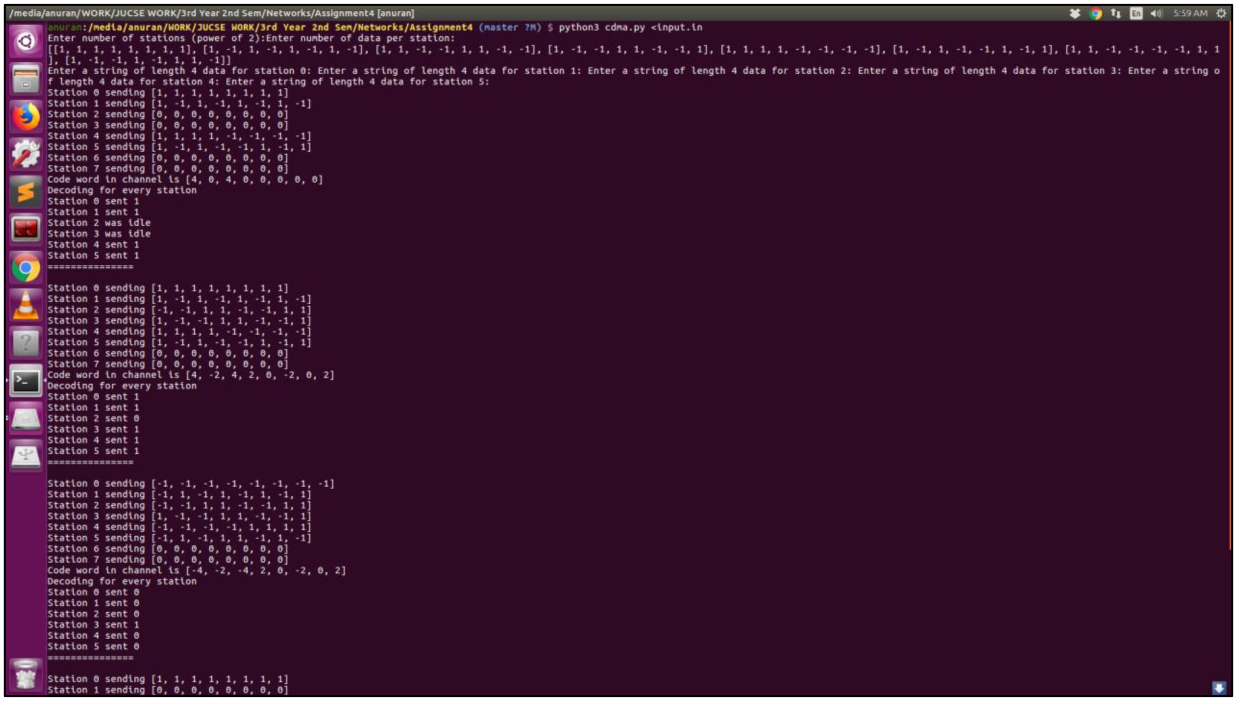
```
# Decode dataword for every station
def decode_cdma(codeword, num_stn):
       for i in range(len(walsh)):
               data=[codeword[j]*walsh[i][j] for j in range(len(walsh[i]))]
               data=sum(data)
               data=int(data/num_stn)

               if(data==-1):
                       data=0
                       print('Station '+str(i)+' sent '+str(data))
               elif(data==0):
                       print('Station '+str(i)+' was idle')
               else:
                       print('Station '+str(i)+' sent '+str(data))
```

__simulate():__

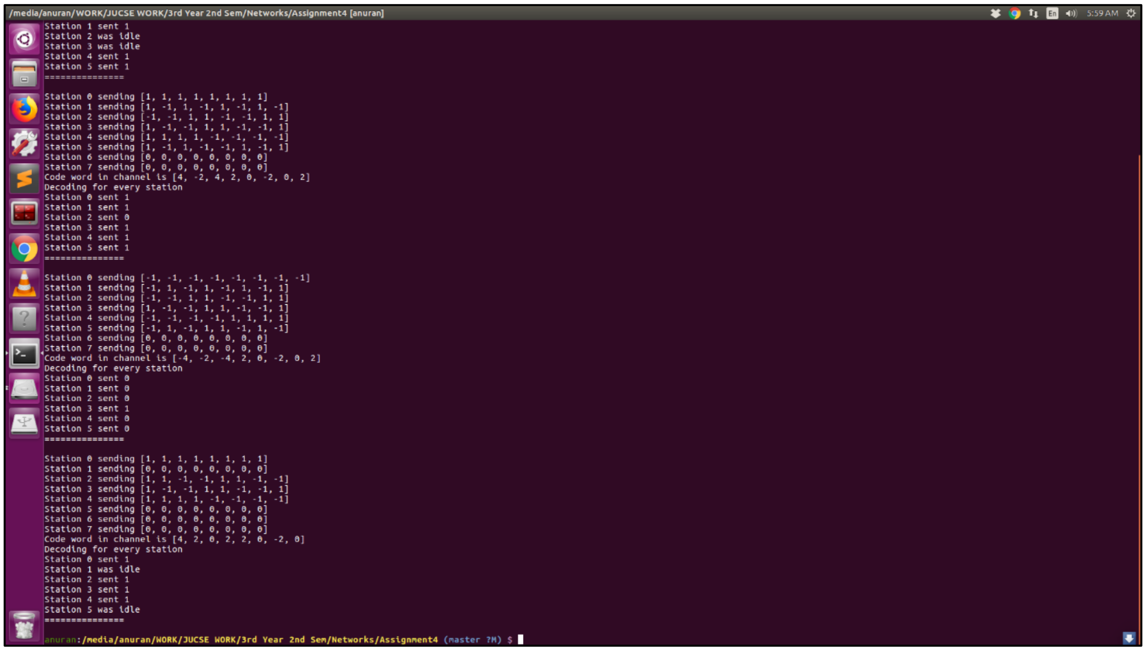
Function to simulate the whole procedure.

```
def simulate():


        num_stn = input('Enter number of stations (power of 2):')
        num_stn=int(num_stn)

        max_num_stn=2**(ceil(log(num_stn,2)))

        num_data = input('Enter number of data per station:')
        num_data=int(num_data)

        createWalsh(max_num_stn)
        print()
        print(walsh)

        stns=[]

        for i in range(num_stn):
                frames = input('Enter a string of length '+str(num_data)+'
data for station '+str(i)+': ')
                frames=list(frames)

                # Now create the station object
                tempstn=Station(i,num_data,walsh[i],frames)
                stns.append(tempstn)

        for i in range(num_stn,max_num_stn):
                frames=num_data*'i'
                frames=list(frames)
                tempstn=Station(i,num_data,walsh[i],frames)
                stns.append(tempstn)

        print()

        # Send data for every data
        for i in range(num_data):
                code=[0 for i in range(max_num_stn)]
                # Send for every station
                for j in range(max_num_stn):
                        # code=code+stns[j].sendData(i)
                        code = [x+y for x,y in zip(code, stns[j].sendData(i))]

                print('Code word in channel is '+str(code))

                print('Decoding for every station')
                decode_cdma(code,num_stn,max_num_stn)
                print(15*'=')
                print()
```

OUTPUTS


/media/anuran/WORK/JUCSE WORK/3rd Year 2nd Sem/Networks/Assignment4 [anuran]
En
5:59 AM

```
anuran:/media/anuran/WORK/JUCSE WORK/3rd Year 2nd Sem/Networks/Assignment4 (master ?M) $ python3 cdma.py <input.in
Enter number of stations (power of 2):Enter number of data per station:
[[1, 1, 1, 1, 1, 1, 1, 1], [1, -1, 1, -1, 1, -1, 1, -1], [1, 1, -1, -1, 1, 1, -1, -1], [1, -1, -1, 1, 1, -1, -1, 1], [1, 1, 1, 1, -1, -1, -1, -1], [1, -1, 1, -1, -1, 1, -1, 1], [1, 1, -1, -1, -1, -1, 1, 1
], [1, -1, -1, 1, -1, 1, 1, -1]]
Enter a string of length 4 data for station 0: Enter a string of length 4 data for station 1: Enter a string of length 4 data for station 2: Enter a string of length 4 data for station 3: Enter a string o
f length 4 data for station 4: Enter a string of length 4 data for station 5:
Station 0 sending [1, 1, 1, 1, 1, 1, 1, 1]
Station 1 sending [1, -1, 1, -1, 1, -1, 1, -1]
Station 2 sending [0, 0, 0, 0, 0, 0, 0, 0]
Station 3 sending [0, 0, 0, 0, 0, 0, 0, 0]
Station 4 sending [1, 1, 1, 1, -1, -1, -1, -1]
Station 5 sending [1, -1, 1, -1, -1, 1, -1, 1]
Station 6 sending [0, 0, 0, 0, 0, 0, 0, 0]
Station 7 sending [0, 0, 0, 0, 0, 0, 0, 0]
Code word in channel is [4, 0, 4, 0, 0, 0, 0, 0]
Decoding for every station
Station 0 sent 1
Station 1 sent 1
Station 2 was idle
Station 3 was idle
Station 4 sent 1
Station 5 sent 1
===============

Station 0 sending [1, 1, 1, 1, 1, 1, 1, 1]
Station 1 sending [1, -1, 1, -1, 1, -1, 1, -1]
Station 2 sending [-1, -1, 1, 1, -1, -1, 1, 1]
Station 3 sending [1, -1, -1, 1, 1, -1, -1, 1]
Station 4 sending [1, 1, 1, 1, -1, -1, -1, -1]
Station 5 sending [1, -1, 1, -1, -1, 1, -1, 1]
Station 6 sending [0, 0, 0, 0, 0, 0, 0, 0]
Station 7 sending [0, 0, 0, 0, 0, 0, 0, 0]
Code word in channel is [4, -2, 4, 2, 0, -2, 0, 2]
Decoding for every station
Station 0 sent 1
Station 1 sent 1
Station 2 sent 0
Station 3 sent 1
Station 4 sent 1
Station 5 sent 1
===============

Station 0 sending [-1, -1, -1, -1, -1, -1, -1, -1]
Station 1 sending [-1, 1, -1, 1, -1, 1, -1, 1]
Station 2 sending [-1, -1, 1, 1, -1, -1, 1, 1]
Station 3 sending [1, -1, -1, 1, 1, -1, -1, 1]
Station 4 sending [-1, -1, -1, -1, 1, 1, 1, 1]
Station 5 sending [-1, 1, -1, 1, 1, -1, 1, -1]
Station 6 sending [0, 0, 0, 0, 0, 0, 0, 0]
Station 7 sending [0, 0, 0, 0, 0, 0, 0, 0]
Code word in channel is [-4, -2, -4, 2, 0, -2, 0, 2]
Decoding for every station
Station 0 sent 0
Station 1 sent 0
Station 2 sent 0
Station 3 sent 1
Station 4 sent 0
Station 5 sent 0
===============

Station 0 sending [1, 1, 1, 1, 1, 1, 1, 1]
Station 1 sending [0, 0, 0, 0, 0, 0, 0, 0]
```


/media/anuran/WORK/JUCSE WORK/3rd Year 2nd Sem/Networks/Assignment4 [anuran]
En
5:59 AM

```
Station 1 sent 1
Station 2 was idle
Station 3 was idle
Station 4 sent 1
Station 5 sent 1
===============

Station 0 sending [1, 1, 1, 1, 1, 1, 1, 1]
Station 1 sending [1, -1, 1, -1, 1, -1, 1, -1]
Station 2 sending [-1, -1, 1, 1, -1, -1, 1, 1]
Station 3 sending [1, -1, -1, 1, 1, -1, -1, 1]
Station 4 sending [1, 1, 1, 1, -1, -1, -1, -1]
Station 5 sending [1, -1, 1, -1, -1, 1, -1, 1]
Station 6 sending [0, 0, 0, 0, 0, 0, 0, 0]
Station 7 sending [0, 0, 0, 0, 0, 0, 0, 0]
Code word in channel is [4, -2, 4, 2, 0, -2, 0, 2]
Decoding for every station
Station 0 sent 1
Station 1 sent 1
Station 2 sent 0
Station 3 sent 1
Station 4 sent 1
Station 5 sent 1
===============

Station 0 sending [-1, -1, -1, -1, -1, -1, -1, -1]
Station 1 sending [-1, 1, -1, 1, -1, 1, -1, 1]
Station 2 sending [-1, -1, 1, 1, -1, -1, 1, 1]
Station 3 sending [1, -1, -1, 1, 1, -1, -1, 1]
Station 4 sending [-1, -1, -1, -1, 1, 1, 1, 1]
Station 5 sending [-1, 1, -1, 1, 1, -1, 1, -1]
Station 6 sending [0, 0, 0, 0, 0, 0, 0, 0]
Station 7 sending [0, 0, 0, 0, 0, 0, 0, 0]
Code word in channel is [-4, -2, -4, 2, 0, -2, 0, 2]
Decoding for every station
Station 0 sent 0
Station 1 sent 0
Station 2 sent 0
Station 3 sent 1
Station 4 sent 0
Station 5 sent 0
===============

Station 0 sending [1, 1, 1, 1, 1, 1, 1, 1]
Station 1 sending [0, 0, 0, 0, 0, 0, 0, 0]
Station 2 sending [1, 1, -1, -1, 1, 1, -1, -1]
Station 3 sending [1, -1, -1, 1, 1, -1, -1, 1]
Station 4 sending [1, 1, 1, 1, -1, -1, -1, -1]
Station 5 sending [0, 0, 0, 0, 0, 0, 0, 0]
Station 6 sending [0, 0, 0, 0, 0, 0, 0, 0]
Station 7 sending [0, 0, 0, 0, 0, 0, 0, 0]
Code word in channel is [4, 2, 0, 2, 2, 0, -2, 0]
Decoding for every station
Station 0 sent 1
Station 1 was idle
Station 2 sent 1
Station 3 sent 1
Station 4 sent 1
Station 5 was idle
===============

anuran:/media/anuran/WORK/JUCSE WORK/3rd Year 2nd Sem/Networks/Assignment4 (master ?M) $
```

RESULTS

The data was successfully decoded for every station.

ANALYSIS

Overall the implementation of the assignment is more or less correct

COMMENTS

Overall the lab assignment was a great learning experience as we got to implement the well-known CDMA protocol ourselves. The assignment can be rated as easy.