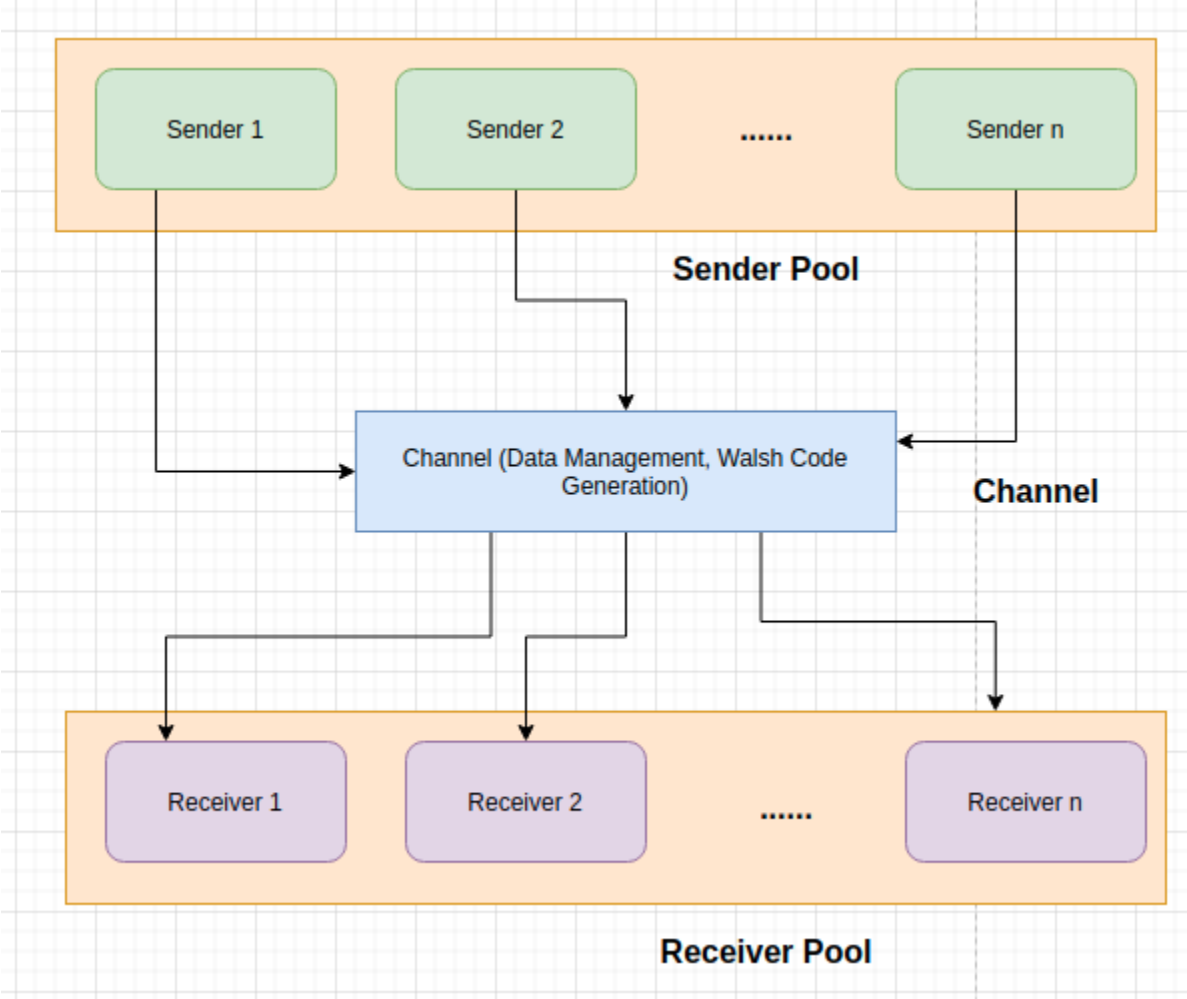
Network Assignment 4

Name :- Debargha Mukherjee Roll :- 001910501067

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

Solution :

System Design



(Above figure shows the system architecture followed throughout this assignment)

Design Elements

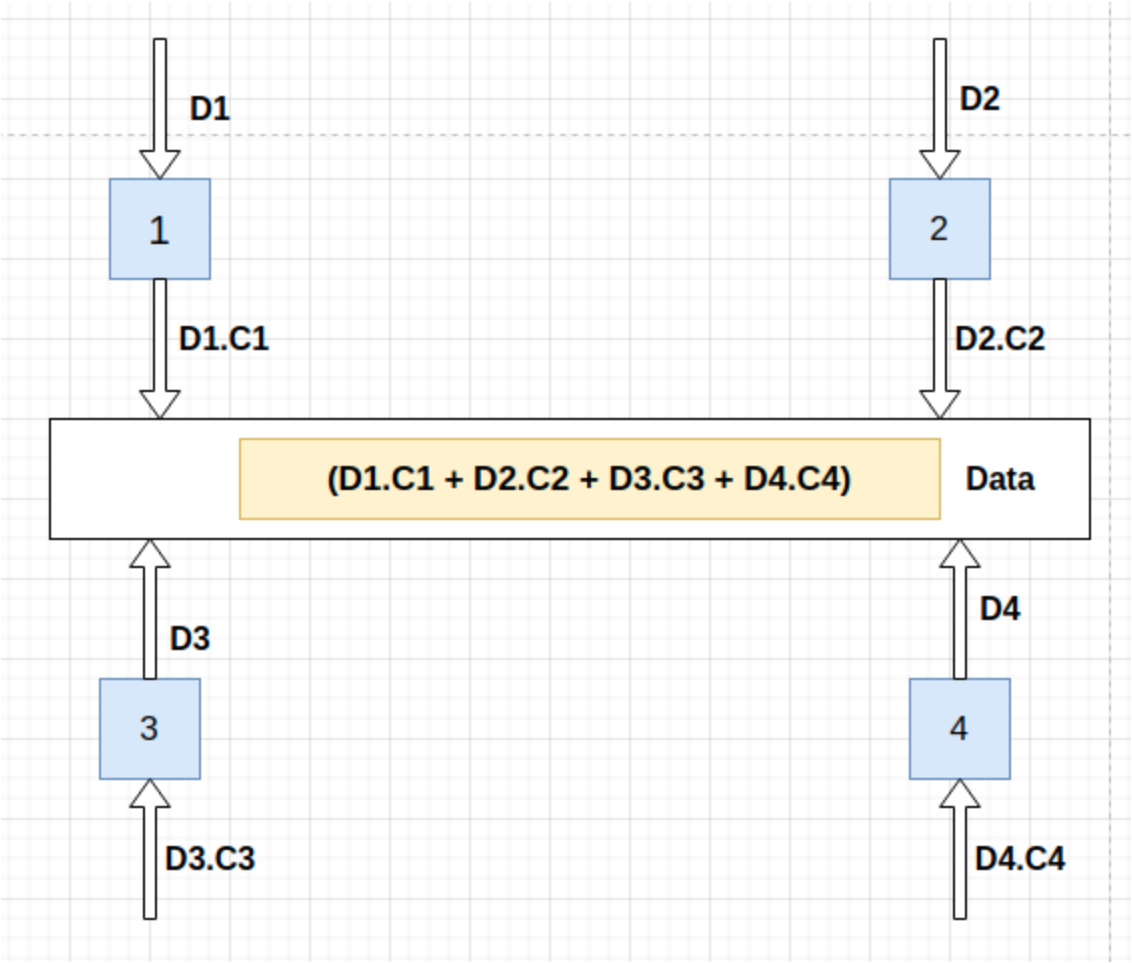
Sender :- It is responsible for reading each character (byte) from the data file and sending bit by bit for every character. For every bit, multiply it with the walsh code for the respective sender.

Channel :- Receive data from the senders in respective threads. Manage the received data in another thread, i.e. generate the Walsh code for all the data received . Send the cumulated Walsh code to the respective receiver in the sender threads.

Receiver :- Connect with the channel. Receive data(cumulated Walsh code) from the channel. Calculate the multiplication and summation of the data received and the Walsh code of the respective receiver, and divide it by no. of stations to get the data bit. Generate the characters from each 8 data bits, and store them in the output file.

Implementation

CDMA employs analog-to-digital conversion (ADC) in combination with spread spectrum technology.
Audio input is first digitized into binary elements. The frequency of the transmitted signal is then made
to vary according to a defined pattern (code), so it can be intercepted only by a receiver whose frequency response is programmed with the same code, so it follows exactly along with the transmitter
frequency.
CDMA has a number of distinguishing features that are key to spread spectrum transmission technologies:
● Use of wide bandwidth: CDMA, like other spread spectrum technologies uses a wider bandwidth
than would otherwise be needed for the transmission of the data. This results in a number of advantages including an increased immunity to interference or jamming, and multiple user access.
● Spreading codes used: In order to achieve the increased bandwidth, the data is spread by the use
of a code that is independent of the data.
● Level of security: In order to receive the data, the receiver must have a knowledge of the spreading code, without this it is not possible to decipher the transmitted data, and this gives a measure of security.
● Multiple access: The use of the spreading codes which are independent for each user along with
synchronous reception allows multiple users to access the same channel simultaneously.



(The idea of communication in CDMA)

(Data representation in CDMA)

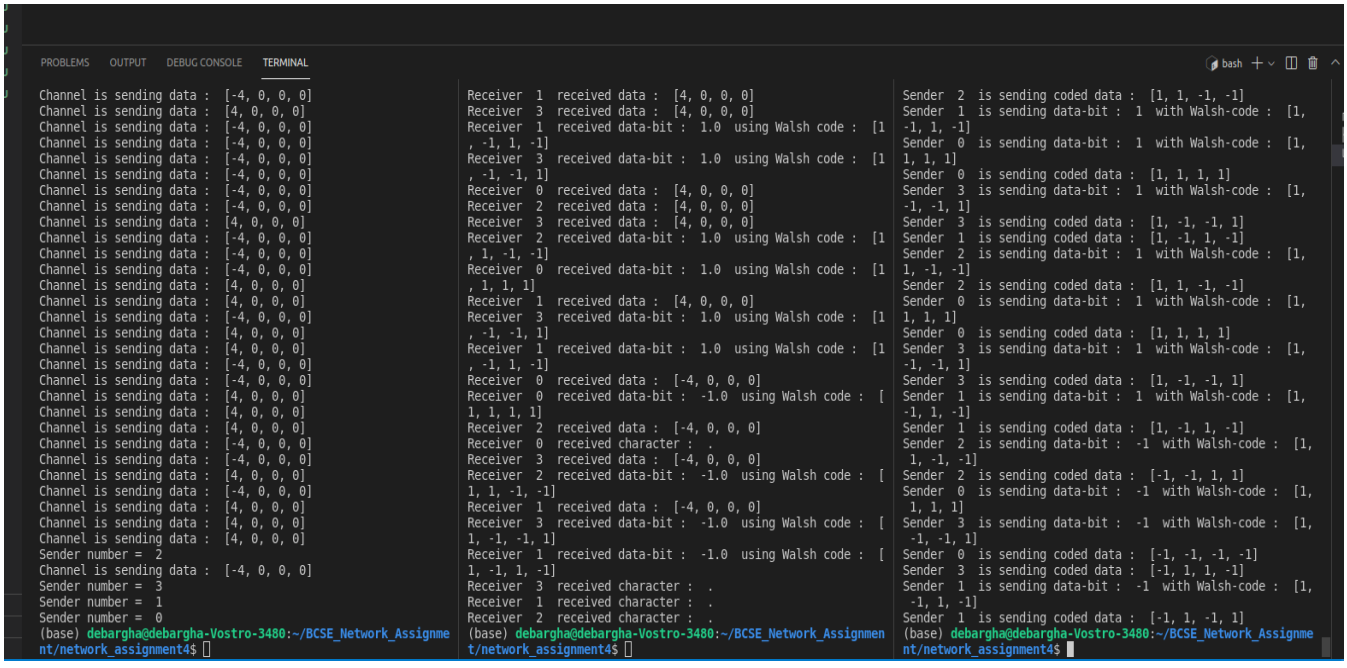
Function to build the Walsh Table:

```
def buildWalshTable(wTable, length, i1,i2, j1,j2, isComplement):
if length == 2:
      if not isComplement:
            wTable[i1][j1] = 1
            wTable[i1][j2] = 1
            wTable[i2][j1] = 1
            wTable[i2][j2] = -1
```

```
        else:
        wTable[i1][j1] = -1
        wTable[i1][j2] = -1
        wTable[i2][j1] = -1
        wTable[i2][j2] = 1
        return
midi = (i1+i2)//2
midj = (j1+j2)//2


buildWalshTable(wTable, length/2, i1, midi, j1, midj, isComplement)
buildWalshTable(wTable, length/2, i1, midi, midj+1, j2, isComplement)
buildWalshTable(wTable, length/2, midi+1, i2, j1, midj, isComplement)
buildWalshTable(wTable, length/2, midi+1, i2, midj+1, j2, not
isComplement)
```

Output Snapshots



Results :

We have used total time taken, effective bandwidth, and average successful transmission time of a data bit as the parameters to compare the CDMA with increasing no. of senders.

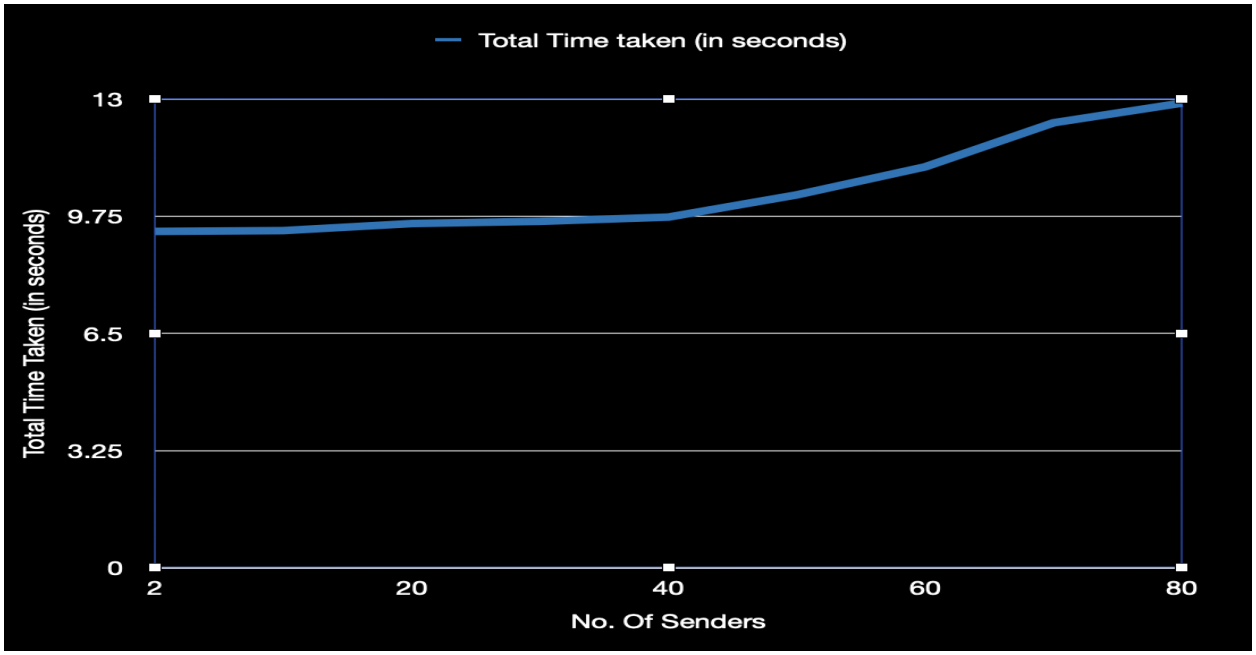
$$Effective\ Bandwidth = \frac{Total\ no.\ of\ data\ bits\ sent\ by\ all\ senders}{Total\ time\ taken\ for\ the\ transmission}$$

$$Average\ Successful\ Transmission\ Time = \frac{Total\ time\ taken\ for\ the\ transmission}{Average\ no.\ of\ data\ bits\ sent\ by\ a\ sender}$$

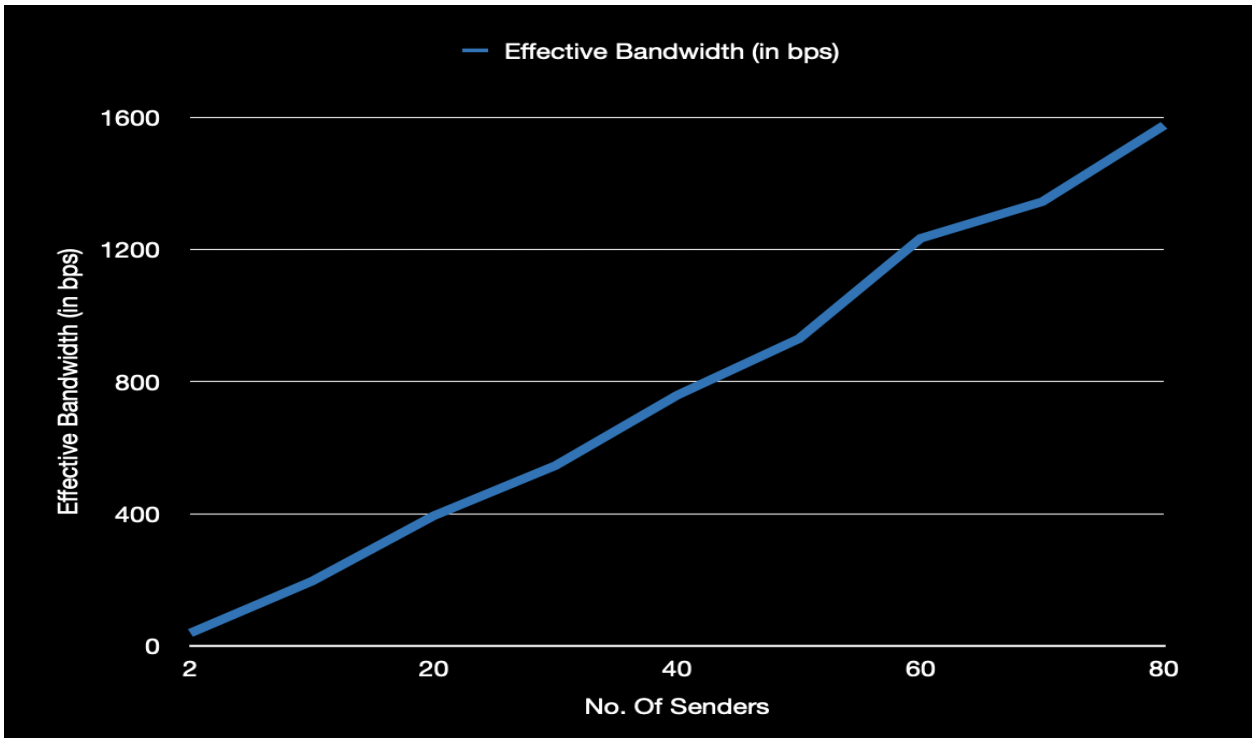
Table 1

No. Of Senders	Total Time taken (in seconds)	Effective Bandwidth (in bps)	Average Successful Transmission Time of a data bit
2	9.332	39	0.0507
10	9.354	196	0.0509
20	9.550	394	0.0506
30	9.611	546	0.0511
40	9.729	759	0.0515
50	10.345	930	0.0518
60	11.123	1234	0.0538
70	12.345	1345	0.0567
80	12.891	1576	0.0611

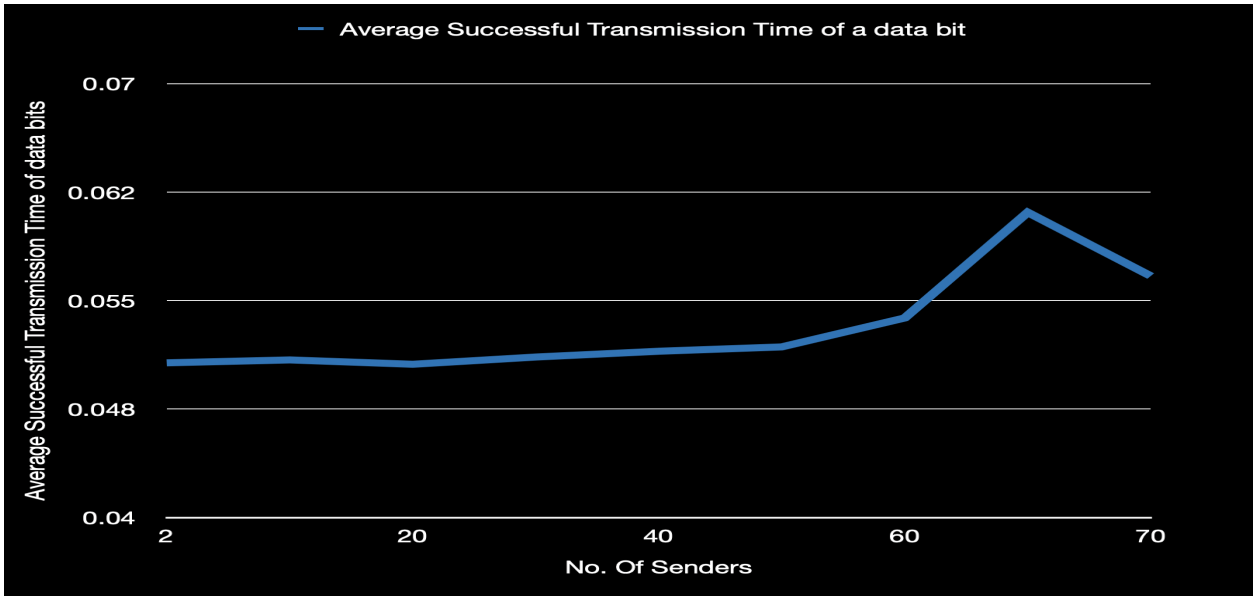
Total Time Taken: From the graph below, we can observe that up to the no. of senders of around 64, the total time taken remains almost constant, and then faces some fluctuations. Theoretically, CDMA should be taking constant time independent of no. of senders, which implies the curve should be a straight line parallel to the x-axis, but they fluctuate gradually with more no. of senders, possibly because of the time slot lapse that occurs due to channel processing time exceeds the time for a time slot.



Effective Bandwidth: Theoretically in CDMA, there is no chance of collision and hence no time slot loss occurs. So the curve should be a linear increasing curve, where the effective bandwidth or channel utilization increases with the no. of senders. This happens exactly the same for up to a certain no. of senders around 64. But, after that, the curve takes a dip, which is possibly due to the channel taking more time than a timeslot (i.e. 50ms here) to club the data. So, some of the timeslots get lapsed without sending any data. This thing is more common when no. of senders is high.



Average Successful Transmission Time: Average Successful Transmission time for a data bit remains almost constant for all types of sender counts, which should be the case for CDMA.



Further Improvements

➢ No flow control protocol is considered, hence an error-free channel is assumed, which is not a practical scenario.

➢ No frame format for the packets is also considered.

➢ This would have been more efficient if it was implemented in a language closer to the system such as C/C++.