Computer Networks

Experiment 5

Name: Ayush Jain

SAPID: 60004200132

Batch: B2

Branch: Computer Engineering

Aim: To study and implement different framing techniques.

Theory: Frames are the units of digital transmission, particularly in computer networks and telecommunications. Frames are comparable to the packets of energy called photons in the case of light energy. Frame is continuously used in Time Division Multiplexing process.

1) **Bit Stuffing:** Bit stuffing is also known as bit-oriented framing or bit-oriented approach. In bit stuffing, extra bits are being added by network protocol designers to data streams. It is generally insertion or addition of extra bits into transmission unit or message to be transmitted as simple way to provide and give signalling information and data to receiver and to avoid or ignore appearance of unintended or unnecessary control sequences.
   It is type of protocol management simply performed to break up bit pattern that results in transmission to go out of synchronization. Bit stuffing is very essential part of transmission process in network and communication protocol. It is also required in USB.

   Bit sequence: 110101111101011111101011111110 (without bit stuffing)

   Bit sequence: 1101011111**0**01011111**0**101011111**0**110 (with bit stuffing) After 5 consecutive 1-bits, a 0-bit is stuffed. Stuffed bits are marked bold.

2) **Character Stuffing:** Character stuffing is also known as byte stuffing or character oriented framing and is same as that of bit stuffing but byte stuffing actually operates on bytes whereas bit stuffing operates on bits. In byte stuffing, special byte that is basically known as ESC (Escape Character) that has predefined pattern is generally added to data section of the data stream or frame when there is message or character that has same pattern as that of flag byte.

But receiver removes this ESC and keeps data part that causes some problems or issues. In simple words, we can say that character stuffing is addition of 1 additional byte if there is presence of ESC or flag in text.

3) **Character Count:** This method is rarely used and is generally required to count total number of characters that are present in frame. This is be done by using field in header. Character count method ensures data link layer at the receiver or

destination about total number of characters that follow, and about where the frame ends.
There is disadvantage also of using this method i.e., if anyhow character count is disturbed or distorted by an error occurring during transmission, then destination or receiver might lose synchronization. The destination or receiver might also be not able to locate or identify beginning of next frame.

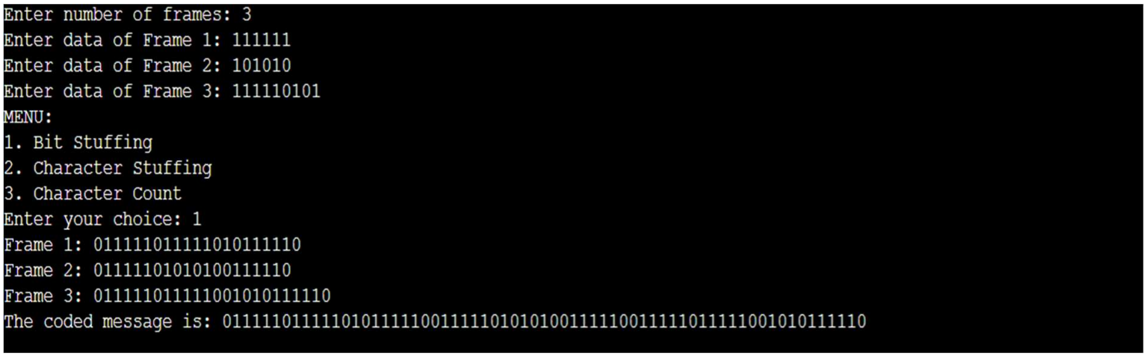
Code:

```
import java.util.Scanner;
class Main
{
 static char flag='/';
 static char esc='#';
 static String bit="0111110";
 public static void main(String args[])
 {
 int x, n, i, j, l;
 Scanner sc = new Scanner(System.in);
 System.out.print("Enter number of frames: ");
 n=sc.nextInt();
 String[] data = new String[n];
 for(i=0; i<n; i++)
 {
 System.out.print("Enter data of Frame "+(i+1)+": ");
 data[i]=sc.next();
```

```
}

System.out.println("MENU:");

System.out.print("1. Bit Stuffing\n2. Character Stuffing\n3. Character Count\n");

System.out.print("Enter your choice: ");

x=sc.nextInt();

switch(x)

{

case 1:

for(i=0; i<n; i++)

{

for(j=0; j<data[i].length()-5; j++)

{

if(data[i].substring(j,j+5).equals("11111"))

data[i]=data[i].substring(0,j+5)+"0"+data[i].substring(j+5);

}

data[i]=bit+data[i]+bit;

System.out.println("Frame "+(i+1)+": "+data[i]);

}

break;

case 2:

for(i=0; i<n; i++)

{

for(j=0; j<data[i].length(); j++)

{

if(data[i].charAt(j)==flag || data[i].charAt(j)==esc)

{ data[i]=data[i].substring(0,j)+esc+data[i].substring(j);

j++;

}

}
System.out.println("Frame "+(i+1)+": "+flag+data[i]+flag);
}
break;
case 3:
for(i=0; i<n; i++)
{
l=data[i].length()+1;
data[i]=Integer.toString(l)+data[i];
System.out.println("Frame "+(i+1)+": "+data[i]);
}
break;
default:
System.out.println("Invalid Choice!");
}
System.out.print("The coded message is: ");
for(i=0; i<n; i++)
System.out.print(data[i]);
}
}
```

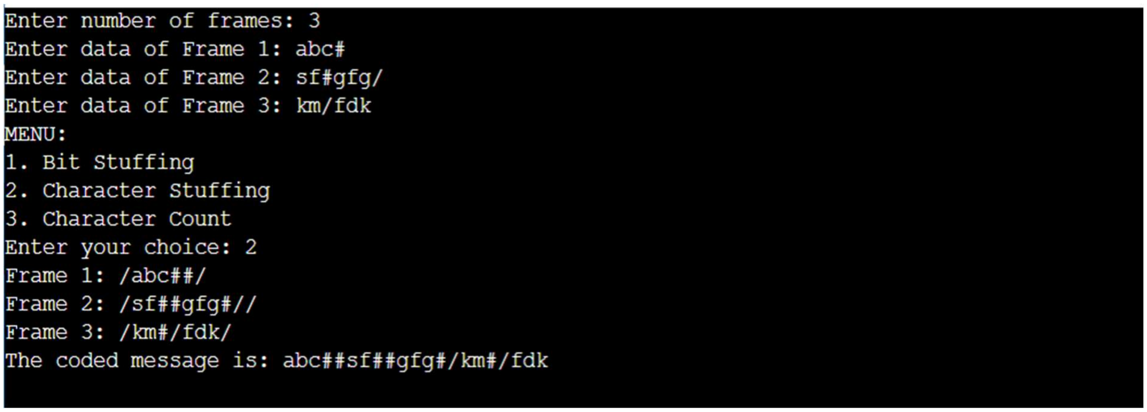
Output:

Bit Stuffing:



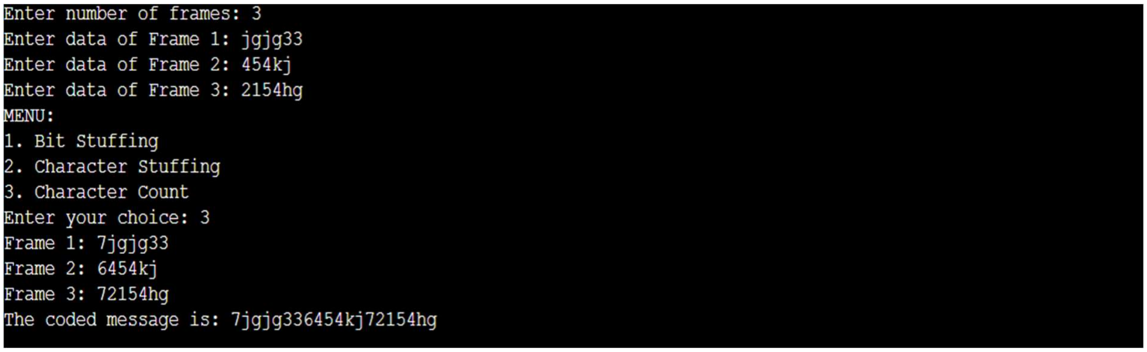
```
Enter number of frames: 3
Enter data of Frame 1: 111111
Enter data of Frame 2: 101010
Enter data of Frame 3: 111110101
MENU:
1. Bit Stuffing
2. Character Stuffing
3. Character Count
Enter your choice: 1
Frame 1: 011111011111010111110
Frame 2: 01111101010100111110
Frame 3: 011111011111001010111110
The coded message is: 01111101111101011111001111101010100111110011111011111001010111110
```

Character Stuffing:



```
Enter number of frames: 3
Enter data of Frame 1: abc#
Enter data of Frame 2: sf#gfg/
Enter data of Frame 3: km/fdk
MENU:
1. Bit Stuffing
2. Character Stuffing
3. Character Count
Enter your choice: 2
Frame 1: /abc##/
Frame 2: /sf##gfg#//
Frame 3: /km#/fdk/
The coded message is: abc##sf##gfg#/km#/fdk
```

Character Count:



```
Enter number of frames: 3
Enter data of Frame 1: jgjg33
Enter data of Frame 2: 454kj
Enter data of Frame 3: 2154hg
MENU:
1. Bit Stuffing
2. Character Stuffing
3. Character Count
Enter your choice: 3
Frame 1: 7jgjg33
Frame 2: 6454kj
Frame 3: 72154hg
The coded message is: 7jgjg336454kj72154hg
```

Conclusion: Hence we have implemented different framing techniques.