

COMPUTER NETWORKS

NAME - KAPAROTU VENKATA SURYA THARANI

USN - 22BTRAD018

BRANCH - AIDE

Implement Data Link Layer Framing Methods

1. Implement the character framing method.

The Character Framing Method in computer networks is a technique used to ensure reliable data transmission between devices. It involves encapsulating data into frames, where each frame consists of a header, payload, and footer. The header and footer contain control information, such as the source and destination addresses, while the payload holds the actual data. This method allows for efficient transmission, error detection, and synchronization, ensuring that the data is received and interpreted correctly by the receiving device.

Code:

```
import java.io.*; import java.util.*;
class CharCount
{
    public static void main(String args[])
    {

        Scanner k=new Scanner(System.in);
        System.out.print("enter a string\t");
        String str=k.next();
        Character c = new Character(str.charAt(0));
        String s = c.toString();
        int p = Integer.parseInt(s);
        int i;

        int m=str.length(); for(i=0;(p+i)<=m;)
        {
            if(p==((str.substring(i,p+i).length())))
            {
```

```
Character c1 = new Character(str.charAt(p)); String s1 = c1.toString();  
i+=p; p=Integer.parseInt(s1);
```

```
}  
}
```

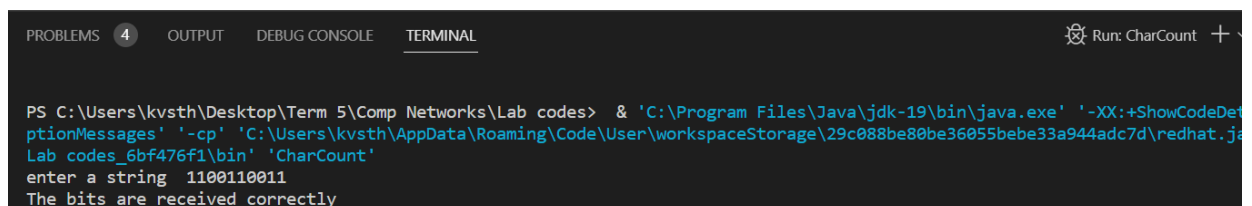
```
if(i==m)  
{
```

```
System.out.println("The bits are received correctly");
```

```
}  
else
```

```
{  
System.out.println("The bits are not received correctly");  
}  
}  
}
```

Output:



The screenshot shows a terminal window with the following content:

```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL Run: CharCount +  
PS C:\Users\kvsth\Desktop\Term 5\Comp Networks\Lab codes> & 'C:\Program Files\Java\jdk-19\bin\java.exe' '-XX:+ShowCodeDet  
ptionMessages' '-cp' 'C:\Users\kvsth\AppData\Roaming\Code\User\workspaceStorage\29c088be80be36055bebe33a944adc7d\redhat.ja  
Lab codes_6bf476f1\bin' 'CharCount'  
enter a string 1100110011  
The bits are received correctly
```

2. Implement the character-stuffing framing method.

Character stuffing uses the special start/end characters for framing and allows those characters in the message also.

This method is for the sender to stuff an extra special character whenever the start or end character occurs naturally so that within the message the special character always occurs in pairs.

The receiver recognizes the single special character as start/end and removes from the message the first special character from the pairs received.

Code:

```
import java.io.*; import java.util.*;
import java.lang.*;
class Stuffing
{

public static void main(String args[])

{

Scanner k =new Scanner (System.in);
System.out.println("enter the string\t");
String s=k.nextLine();
String str1;
String str2=""; int i,m,j;
m=s.length();
System.out.println("Data sent: "+s);

str1="dlestx"; for(i=0;i<=m-1;i++)
{

if((s.charAt(i)=='d')&&(s.charAt(i+1)=='l')&&(s.charAt(i+2)=='e'))

{

str1=str1+"dle";

}

str1=str1+s.substring(i,i+1);

}

str1=str1+"dleetx"; int p=str1.length();
System.out.println("Data transmitted: "+str1); for(i=6;i<p-6;i++)
{

if((str1.charAt(i)=='d')&&(str1.charAt(i+1)=='l')&&(str1.charAt(i+2)=='e')&&(str1.charAt(i+3)=='d')
&&(str1.charAt(i+4)=='l')&&(str1.charAt(i+5)=='e'))

{
```

```
i=i+3;

}

str2=str2+str1.substring(i,i+1);

}

System.out.println("Data received: "+str2);

}

}
```

OUTPUT:

```
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\kvsth\Desktop\Term 5\Comp Networks\Lab codes> & 'C:\Program Files\Java\jdk-19\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\kvsth\AppData\Roaming\Code\User\workspaceStorage\29c088be80be36055bebe33a944adc7d\redhat.java\jdt_ws\Lab codes_6bf476f1\bin' 'Stuffing'
enter the string
wxydlesfgh
Data sent: wxydlesfgh
Data transmitted: dlestdxwxydledlesfghdleetx
Data received: wxydlesfgh
PS C:\Users\kvsth\Desktop\Term 5\Comp Networks\Lab codes> 
```

3. Implement the bit stuffing framing method.

Whenever the sender's data link layer encounters five consecutive 1s in the data, it automatically stuffs a zero bit into the outgoing bit stream. This technique is called bit stuffing. When the receiver sees five consecutive 1s in the incoming data stream, followed by a zero bit, it automatically destuffs the 0 bit. The boundary between the two frames can be determined by locating the flag pattern.

Code:

```
package p2;

import

java.util.*;

public class BitStuffing

{

    public static void main(String[] args)

    {

        Scanner sc=new Scanner(System.in);

        System.out.print("Enter the value: ");

        String d1 = sc.nextLine();

        String remaining = new String();

        String output=new String();

        int counter = 0;

        for(int i=0;i<d1.length();i++)

        {

            if (d1.charAt(i)!='1' && d1.charAt(i)!='0')

            {

                System.out.println("Enter valid Binary values");

                return;

            }

            if(d1.charAt(i) == '1')
```

```

        {
            counter++;

            remaining = remaining + d1.charAt(i);
        }
    else
    {
        remaining = remaining + d1.charAt(i);

        counter = 0;
    }

    if(counter == 5)
    {
        remaining = remaining + '0';

        counter = 0;
    }
}

String new1="|01111110 | "+remaining+" | 01111110|";

System.out.println("Stuffed data:");

for(int k=0;k<=(28+d1.length());k++)

{

    System.out.print("-");

}

System.out.println(); System.out.println(" "+new1);
for(int k=0;k<=(28+d1.length());k++)

{

    System.out.print("-");

}

System.out.println();

counter=0;

for(int i=0;i<remaining.length();i++)

{

```

```

if(remaining.charAt(i) == '1')
{

    counter++;

    output = output + remaining.charAt(i);

}
else
{
    output = output + remaining.charAt(i);
    counter = 0;
}
if(counter == 5)
{
    if((i+2)!=remaining.length())
    {
        output = output + remaining.charAt(i+2);
    }
    else
    {
        output=output + '1';
    }
    i=i+2;
    counter =
    1;
}
}

System.out.println("Destuffed BIT is: "+output);

```

```
}  
  
}
```

Output:

```
PROBLEMS 9 OUTPUT DEBUG CONSOLE TERMINAL  
  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  
  
PS C:\Users\kvsth\Desktop\Term 5\Comp Networks\Lab codes> & 'C:\Program Files\Java\jdk-19\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\kvsth\AppData\Roaming\Code\User\workspaceStorage\29c088be80be36055bebe33a944adc7d\r  
edhat.java\jdt_ws\Lab codes_6bf476f1\bin' 'BitStuffing'  
Enter the value: 001101111110011  
Stuffed data:  
-----  
|01111110 | 00110111111010011 | 01111110|  
-----  
Destuffed BIT is: 001101111110011  
PS C:\Users\kvsth\Desktop\Term 5\Comp Networks\Lab codes> |
```