Computer Networks - Exp 5

Kartik Jolapara

60004200107 - B1

Aim

To study and implement different framing techniques.

Theory

Character Count

This method uses a field in the header to specify the number of characters in the frame. When the data link layer at the destination sees the character count, it knows how many characters follow and hence where the end of the frame is.

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.

Here the data is stuffed at start and end with characters not present in the data word itself. Thus, we return the data by adding the unique start and ending characters or a 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 the receiver removes this ESC and keeps the data part that causes some problems or issues. In simple words, we can say that character stuffing is an addition of 1 additional byte if there is presence of ESC or flag in text.

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 a transmission unit or message to be transmitted as a simple way to provide and give signaling information and data to the receiver and to avoid or ignore appearance of unintended or unnecessary control sequences. It is a type of protocol management simply performed to break up a bit pattern that results in transmission to go out of synchronization. Bit stuffing is a very essential part of the transmission process in network and communication protocol. It is also required in USB.

Code

```
#include <iostream>

#include <string>


using namespace std;


int main()

{

   int opt;

   do

   {

      cout << "1. Character count\n2. Starting and ending with character stuffing\n3. Character stuffing\n4. Starting and ending Flag, with bit stuffng.\n5. Exit";

      cout << "\n\nEnter an option: ";

      string data, finalData, temp;
```

```
cin >> opt;
switch (opt)
{
case 1:
{
    cout << "Enter the number of frames: ";
    int frames;
    cin >> frames;
    while (frames--)
    {
        cout << "Enter frame data: ";
        cin >> temp;
        finalData += (std::to_string(temp.length()) + temp);
    }
    cout << "\nFinal data: " << finalData << "\n\n";
    break;
}
case 2:
{
    cout << "Enter the data: ";
    cin >> data;
```

```
    string stx = "STX", dle = "DLE";
    finalData = stx + dle + data + dle + stx;
    cout << "\nFinal data: " << finalData << "\n\n";
    break;
}
case 3:
{
    cout << "Enter the data: ";
    cin >> data;
    string stx = "STX", dle = "DLE";
    int i = 0, count = 0;
    finalData += stx + dle;
    while (i != data.length())
    {
        finalData += data[i];
        if (data[i] == 'D')
        {
            count++;
        }
        else if (count == 1 && data[i] == 'L')
        {
```

```
            count++;
        }
        else if (count == 2 && data[i] == 'E')
        {
            finalData += "DLE";
            count = 0;
        }
        else
        {
            count = 0;
        }
        i++;
    }
    finalData += dle + stx;
    cout << "\nFinal data: " << finalData << "\n\n";
    break;
}
case 4:
{
    cout << "Enter the data: ";
    cin >> data;
```

```
int i = 0, count = 0;
string flag = "01111110";
finalData += flag;
while (i != data.length())
{
    if (data[i] == '1' && count == 5)
    {
        count = 0;
        finalData += '0';
    }
    else if (data[i] == '1')
    {
        count++;
        finalData += data[i];
    }
    else if (count > 0)
    {
        count = 0;
        finalData += data[i];
    }
    i++;
```

```
            }
            finalData += flag;
            cout << "\nFinal data: " << finalData << "\n\n";
            break;
        }
        case 5:
        {
            cout << "Exiting..!\n";
            break;
        }
        default:
        {
            cout << "Enter an valid OPTION!!";
        }
        }
    } while (opt != 5);
    return 0;
}
```

Output

```
1. Character count
2. Starting and ending with character stuffing
3. Character stuffing
4. Starting and ending Flag, with bit stuffng.
5. Exit

Enter an option: 1
Enter the number of frames: 4
Enter frame data: 12
Enter frame data: 3451
Enter frame data: 25
Enter frame data: 123563

Final data: 212434512256123563

1. Character count
2. Starting and ending with character stuffing
3. Character stuffing
4. Starting and ending Flag, with bit stuffng.
5. Exit

Enter an option: 2
Enter the data: fjiem#T53jkfmf88

Final data: STXDLEfjiem#T53jkfmf88DLESTX

1. Character count
2. Starting and ending with character stuffing
3. Character stuffing
4. Starting and ending Flag, with bit stuffng.
5. Exit
```

```
Enter an option: 3
Enter the data: fsjSml4joSTXfjldemDLEfes

Final data: STXDLEfsjSml4joSTXfjldemDLEDLEfesDLESTX

1. Character count
2. Starting and ending with character stuffing
3. Character stuffing
4. Starting and ending Flag, with bit stuffng.
5. Exit

Enter an option: 4
Enter the data: 1101100011101110101100111111000111111111

Final data: 0111111011011011101110101101111101111011101111110

1. Character count
2. Starting and ending with character stuffing
3. Character stuffing
4. Starting and ending Flag, with bit stuffng.
5. Exit

Enter an option: 5
Exiting..!
```

Conclusion

We have Successfully Implemented Different Framing Techniques like Character Count, Character Stuffing and Bits Stuffing used during Communication.