

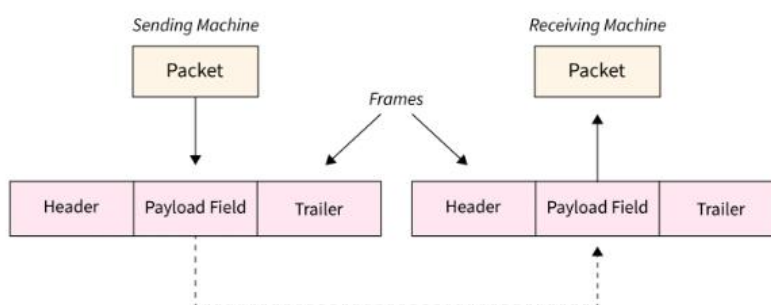


|                      |                                     |
|----------------------|-------------------------------------|
| Name:                | Perna Sunil Jadhav                  |
| Sap Id:              | 60004220127                         |
| Class:               | S. Y. B.Tech (Computer Engineering) |
| Course:              | Computer Networks (DJ12CEL405)      |
| Date of Performance: |                                     |
| Date of Submission:  |                                     |
| Experiment No.:      | 03                                  |
| Aim:                 | Data Link Layer (Framing Mechanism) |

### AIM: DATA LINK LAYER – FRAMING MECHANISM

#### THEORY:

- In the physical layer, data transmission involves synchronised transmission of bits from the source to the destination.
- Data-link layer takes the packets from the Network Layer and encapsulates them into frames. If the frame size becomes too large, then the packet may be divided into small sized frames.

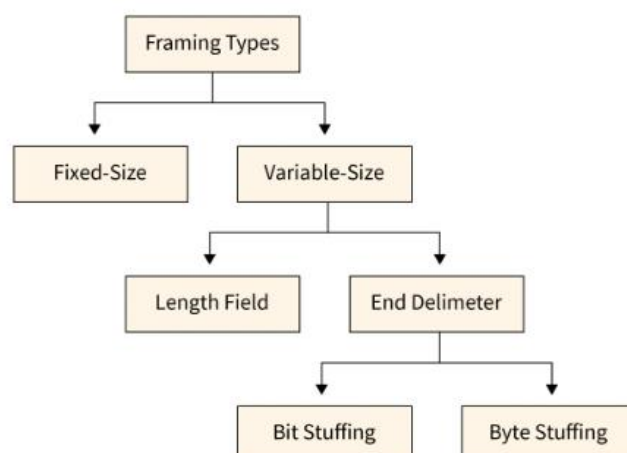


Smaller sized frames makes flow control and error control more efficient.

- Then, it sends each frame bit-by-bit on the hardware. At receiver's end, data link layer picks up signals from hardware and assembles them into frames.

#### Types of Framing in Data Link Layer

- The frame can be of fastened or variable size. founded on the size, the following are the types of framing in data link layers in computer networks
- Fixed Size Framing is used in ATMs, Wide area networks(WAN)
- In variable-size framing, we need a way to outline the tip of the frame and also the starting of the succeeding frame. This can be utilized in local area networks(LAN).

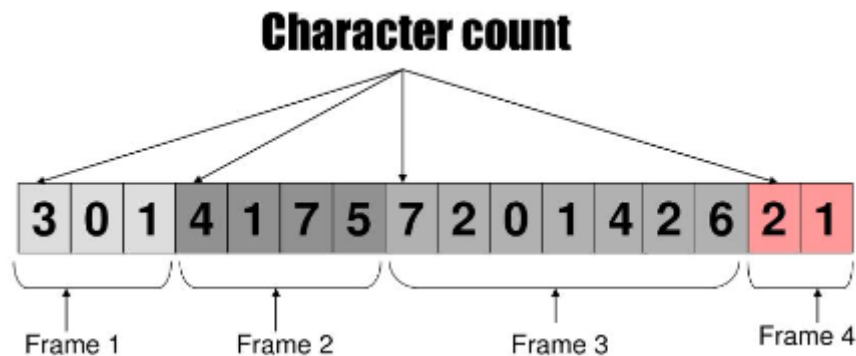




### Methods of Framing:

#### 1. Character Count

- Character count methodology makes sure that the Framing in the data link layer is at the receiver end regarding the total range of characters maintain, and wherever the frame ends.



- It also has its disadvantage conjointly of utilizing this methodology which is, if in any case, the character count is issued or bended by a miscalculation occurring throughout the transmission process, then at the receiver end it may drop synchronization.
- The receiver strength is ineffective to find or establish the start of the next frame.
- Program:

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <stdlib.h>

char *sender(int n)
{
    static char ch[100];
    char data_unit[30];
    char length[1];

    for (int i = 0; i < n; i++)
    {
        printf("Enter the Data Unit %d: ", (i + 1));
        scanf("%s", data_unit);

        length[0] = (strlen(data_unit) + 1) + '0';
        strcat(ch, length);
    }
    return ch;
}
```



```
void receiver(char *data)
{
    size_t i = 0;
    int count = 0;
    printf("\nThe Data Units transferred were: \n");
    while (data[i] != '\0')
    {
        int isADigit = isdigit(data[i]);
        if (isADigit == 1)
        {
            count++;
            printf("Data Unit %d: ", count);
            int dataUnitLen = 0;
            dataUnitLen = data[i] - '0';
            for (int j = 0; j < dataUnitLen - 1; j++)
            {
                i++;
                printf("%c", data[i]);
            }
            i++;
            printf("\n");
        }
    }
}

int main()
{
    int n = 0;
    char *data_To_Be_Transmitted;

    printf("Enter the number of data units: ");
    scanf("%d", &n);

    if (n <= 0)
    {
        printf("No data units received");
        return 1;
    }

    data_To_Be_Transmitted = sender(n);
    printf("Data String to be Transmitted: %s", data_To_Be_Transmitted);
    receiver(data_To_Be_Transmitted);
    return 0;
}
```

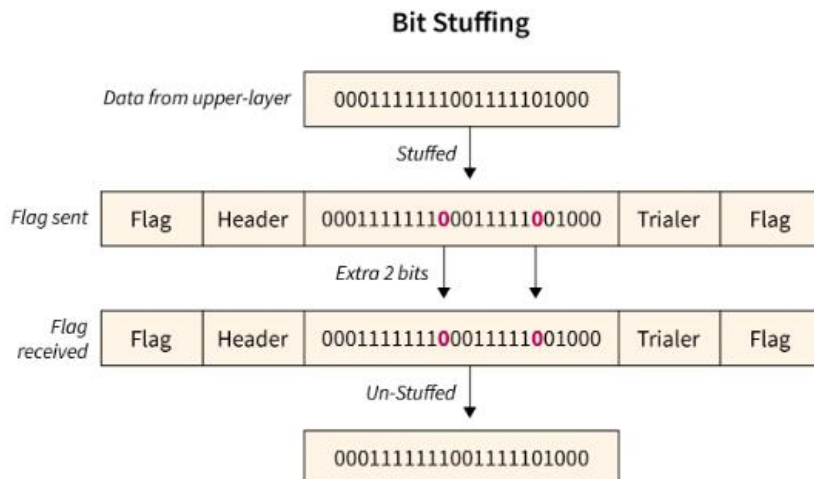


Output:

```
PS C:\Users\Jadhav\Desktop\BTech\4th sem\CN\Code> & 'c:\Users\Jadhav\.vscode\ms-vscode.cpptools-1.14.4-win32-x64\debugAdapters\bin\WindowsDebugLauncher\bin\Microsoft-MIEngine-In-dtg4vblq.pjs' '--stdout=Microsoft-MIEngine-Out-dsmi' '--stderr=Microsoft-MIEngine-Error-dwkzp4ny.lcx' '--pid=Microsoft-MIEngine-Pid' '.nf0' '--dbgExe=C:\msys64\mingw64\bin\gdb.exe' '--interpreter=mi'
Enter the number of data units: 4
Enter the Data Unit 1: This
Enter the Data Unit 2: is
Enter the Data Unit 3: Prerna
Enter the Data Unit 4: Jadhav
Data String to be Transmitted: 5This3is7Prerna7Jadhav
The Data Units transferred were:
Data Unit 1: This
Data Unit 2: is
Data Unit 3: Prerna
Data Unit 4: Jadhav
PS C:\Users\Jadhav\Desktop\BTech\4th sem\CN\Code> 
```

## 2. Bit-Oriented Framing

- Most protocols use a special 8-bit pattern flag 01111110 as a result of the delimiter to stipulate the beginning and so the end of the frame.



- Bit stuffing is completed at the sender end and bit removal at the receiver end.

Program:

```
#include <stdio.h>
#include <string.h>
int main()
{
    int a[20], b[30], i, j, k, count, n;
    printf("Enter frame size:");
    scanf("%d", &n);
    printf("Enter the frame in the form of 0 and 1 :");
```



```
for (i = 0; i < n; i++)
    scanf("%d", &a[i]);

i = 0;
count = 1;
j = 0;
while (i < n) {
    if (a[i] == 1)
    {
        b[j] = a[i];
        for (k = i + 1; a[k] == 1 && k < n && count < 5; k++) {
            j++;
            b[j] = a[k];
            count++;
            if (count == 5) {
                j++;
                b[j] = 0;
            }
            i = k;
        }
    }
    else
    {
        b[j] = a[i];
    }
    i++;
    j++;
}
printf("After Bit Stuffing :");
for (i = 0; i < j; i++)
    printf("%d", b[i]);

return 0;
}
```

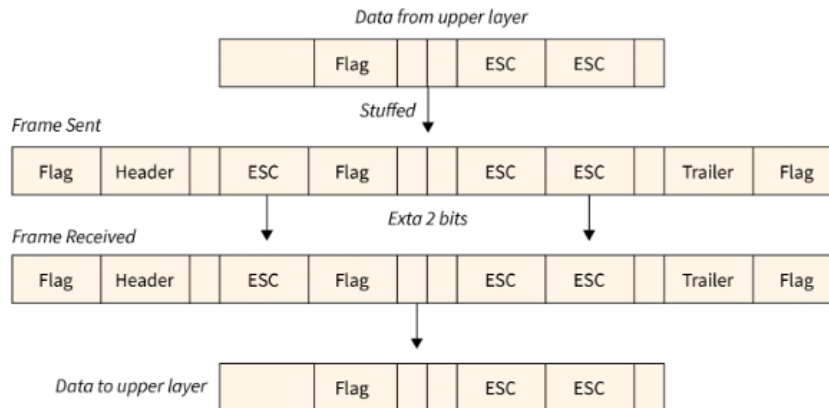
 Output:

```
PS C:\Users\Jadhav\Desktop\BTech\4th sem\CN\Code> & 'c:\Users\Jadhav\.vscode
s\ms-vscode.cpptools-1.14.4-win32-x64\debugAdapters\bin\WindowsDebugLauncher.
din=Microsoft-MIEngine-In-ebxxnml1.db2' '--stdout=Microsoft-MIEngine-Out-w5se
'--stderr=Microsoft-MIEngine-Error-et5cuchp.cjc' '--pid=Microsoft-MIEngine-Pi
.kwf' '--dbgExe=C:\msys64\mingw64\bin\gdb.exe' '--interpreter=mi'
Enter frame size:10
Enter the frame in the form of 0 and 1 : 0 1 1 1 1 1 1 0 1 0
After Bit Stuffing :011111101010
```



### 3. Byte-Oriented Framing

- Byte stuffing is one of the methods of adding an additional byte once there is a flag or escape character within the text.



- Take an illustration of byte stuffing as appeared in the given diagram.
- The sender sends the frame by adding three additional ESC bits and therefore the destination machine receives the frame, and it removes the extra bits to convert the frame into an identical message.
- Program:

```
#include <stdio.h>
#include <string.h>

int main()
{
    char frame[50][50], str[50][50];
    char flag[10];
    strcpy(flag, "flag");
    char esc[10];
    strcpy(esc, "esc");
    int i, j, k = 0, n;
    strcpy(frame[k++], "flag");
    printf("Enter length of String : ");
    scanf("%d", &n);
    printf("Enter the String: \n");
    for (i = 0; i <= n; i++)
    {
        gets(str[i]);
    }
    printf("You entered :\n");
    for (i = 0; i <= n; i++)
    {
        puts(str[i]);
    }
}
```



```
printf("\n");
for (i = 1; i <= n; i++)
{
    if (strcmp(str[i], flag) != 0 && strcmp(str[i], esc) != 0)
    {
        strcpy(frame[k++], str[i]);
    }
    else
    {
        strcpy(frame[k++], "esc");
        strcpy(frame[k++], str[i]);
    }
}
strcpy(frame[k++], "flag");
printf("-----\n");
printf("Byte stuffing at sender side:\n");
printf("-----\n");
for (i = 0; i < k; i++)
{
    printf("%s\t", frame[i]);
}
return 0;
}
```

Output:

```
Adapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-ilmaoa4f.qo4' '--stdout=Microso
'--stderr=Microsoft-MIEngine-Error-irnjnes4.4m1' '--pid=Microsoft-MIEngine-Pid-amwqinbj.myp' '--dbg
exe' '--interpreter=mi'
Enter length of String : 6
Enter the String:
1
1
0
1
0
0
0
You entered :

1
1
0
1
0
0

-----
Byte stuffing at sender side:
-----
flag  1      1      0      1      0      0      flag
PS C:\Users\Jadhav\Desktop\BTech\4th sem\CN\Code> |
```



**Conclusion:**

- ✚ Framing in the Data link layer additionally contains headers that embody information like error-checking codes.
- ✚ Framing in data link layer relay, token ring, ethernet, and alternative sorts of data link layer ways have their frame structures.
- ✚ Framing in the Data link layer enables the information to be divided into multiple recoverable elements that may be inspected for corruption.
- ✚ Framing in the Data link layer provides a flow management mechanism that manages the frame flow such that the information congestion does not occur on slow receivers thanks to quick senders.
- ✚ Framing in the Data link layer provides valid information transfer services within the layers of the peer network.