# FRAMING METHODS

**Experiment No : 5 DATE:11-03-2024**

**Aim**: To study different Framing Techniques.

1. Bit Stuffing
2. Byte Stuffing

# Theory:

* 1. **Bit Stuffing:**
     + Bit stuffing involves adding an extra '0' bit after every five consecutive '1's in the data to avoid misinterpretation.
     + Flags are added at the beginning and end of the data without actual stuffing (i.e., without inserting extra bits).
     + Bit stuffing helps in maintaining clock synchronization between sender and receiver by avoiding long sequences of identical bits.

Example:

Original Data: 11111101010111111

Bit-Stuffed Data: 1111101010101111101

Explanation: Flags '01111110' are added at the beginning and end of the data without any additional stuffing. Extra '0' bits are inserted after every five consecutive '1's to avoid misinterpretation.

# Byte Stuffing:

* + - Byte stuffing uses an escape character to differentiate control characters from actual data bytes.
    - Byte stuffing is commonly used in protocols like PPP (Point-to-Point Protocol) and HDLC (High-Level Data Link Control) to frame data for transmission.
    - Byte stuffing ensures data integrity by avoiding conflicts with control characters such as flags or escape characters.

Example:

Original Data: FLAG A FLAG ESC

Byte-Stuffed Data: FLAG ESC FLAG A ESC FLAG ESC ESC FLAG

Explanation: Flags 'FLAG' is added at the start and end of the data. If the data contains a flag or an escape character ('ESC'), an escape character is added before it to distinguish it from the flag sequence.

**Code:**

1. Bit Stuffing

#include <iostream>

using namespace std;

void bit\_stuffing(string data)

{

int i = 0, j = 0, count = 0; string stuffed\_data;

while(i < data.length())

{

if(data[i] == '1')

{

count++;

}

else

{

count = 0;

}

stuffed\_data += data[i];

if(count == 5 && data[i+1] == '1')

{

stuffed\_data += '0';

count = 0;

}

i++;

}

cout << "Stuffed data: " << stuffed\_data << endl;

}

int main(void)

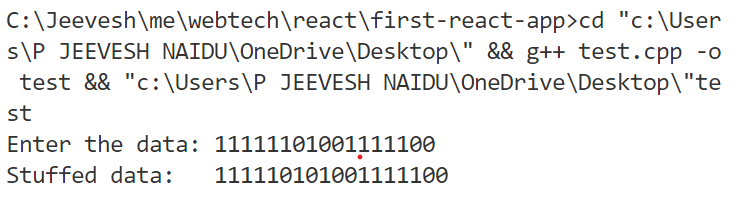
{

string data;

cout << "Enter the data: "; cin >> data; bit\_stuffing(data);

}

**Output:**

****

1. Byte Stuffing

#include <iostream>

using namespace std; int main(void)

{

string data; string d; int n;

cout << "Enter the number of bytes: "; cin >> n;

data += "FLAG ";

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

{

cout << "Enter the data: "; cin >> d;

if (d == "ESC" || d == "FLAG")

{

data += "ESC ";

}

data += d; data += " ";

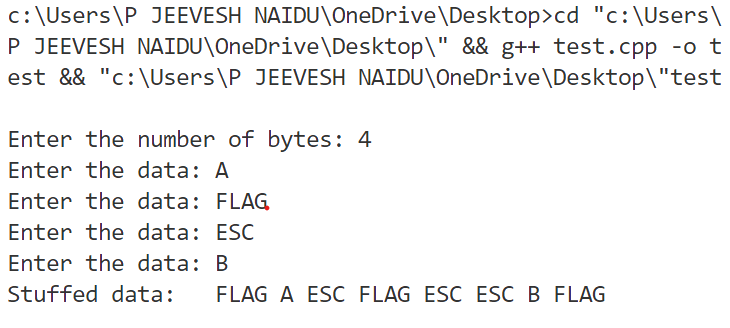
}

data += "FLAG ";

cout << "Stuffed data: " << data << endl; return 0;

}

**Output:**



**Conclusion:**

Studied different types of framing Methods with successful execution of programs.