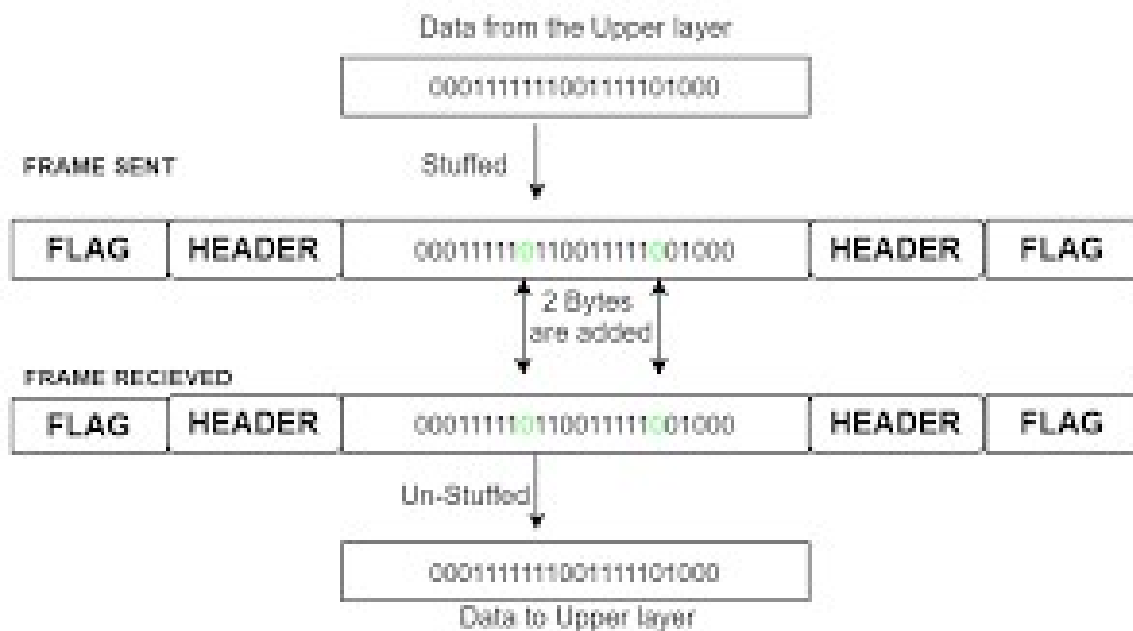


PRACTICAL - 5

DATE: , Wednesday

AIM: Write a program which demonstrates the concept of bit stuffing.

- What is Bit-Stuffing?
 - Bit stuffing is a technique used in data communication to prevent the occurrence of specific bit patterns, like frame delimiters, from appearing in the data itself. By inserting additional bits into the data stream, it ensures that these control patterns are not accidentally interpreted as part of the data, maintaining the integrity of the transmission.



- Code for Bit-Stuffing

```
#include <iostream>
#include <string>
#define PATTERN "11111"
using namespace std;
void bitStuffing(string user_bits) {
    cout << "Passed bits are: " << user_bits << endl;
    size_t pos = 0;
    int indexes[10] = {-1};
    for (int i = 0; i < 10; i++) {
        pos = user_bits.find(PATTERN, pos);
        if (pos != std::string::npos) {
            indexes[i] = pos;
            pos += 1;
        } else {
            break;
        }
    }
    for (int i = 0; i < 10 && indexes[i] != -1; i++) {
        cout << "Position of pattern: " << indexes[i] << endl;
    }
}
int main() {
    cout << "Enter the bits: ";
    string user_bits;
    cin >> user_bits;
    bitStuffing(user_bits);
    return 0;
}
```

- Output

```
PS C:\personal_documents\CSE\CSE_github\SEM 5\CN\Codes\
odes\" ; if ($?) { g++ BitStuffing.cpp -o BitStuffing
Enter the bits: 1011111010111011011111
Passed bits are: 1011111010111011011111
Position of pattern: 2
Position of pattern: 17
```

Date of Submission:

Sign: