



Green University of Bangladesh
Department of Computer Science and Engineering (CSE)
Faculty of Sciences and Engineering
Semester: (Spring, Year:2025), B.Sc. in CSE (DAY0)

Lab Report NO :01
Course Title: Data Communication Lab
Course Code: CSE 308/312 Section: 231_D1

**Lab Experiment Name: Implementation of Byte Stuffing and De-Stuffing &
Binary to IPv4 Conversion using C Programming.**

Student Details

Name		ID
1.	Promod Chandra Das	231002005

Lab Date : 23-02-2025
Submission Date : 02-03-2025
Course Teacher's Name : Ms. Rusmita Halim Chaity

Lab Report Status

Marks:
Comments:.....

Signature:.....
Date:.....

1. TITLE OF THE LAB REPORT EXPERIMENT

Implementation of Byte Stuffing and De-Stuffing & Binary to IPv4 Conversion using C Programming.

2. OBJECTIVES

The objectives of this experiment are to:

1. Implement Byte Stuffing and De-stuffing to ensure special characters are handled correctly in data transmission.
2. Convert a 32-bit Binary IP Address into IPv4 format, a crucial step in networking.
3. Understand the importance of byte stuffing in data communication protocols.
4. Learn how binary IP addresses are processed in networking.

3. PROCEDURE

The experiment consists of two parts: Byte Stuffing & De-stuffing and Binary to IPv4 Conversion. Below are the step-by-step procedures for both.

Byte Stuffing & De-stuffing:

1. Take a bitstream (a string of characters) as input from the user.
2. Check each character:
 - If it is a special character (F for FLAG or E for ESCAPE), add an extra ESCAPE character (E) before it.
 - Otherwise, copy the character as it is.
3. Display the stuffed bitstream (modified data).
4. To retrieve the original data, perform de-stuffing by scanning the stuffed bitstream:
 - If an ESCAPE character (E) is found before another character, remove it.
 - Copy the remaining characters to get back the original data.
5. Print the de-stuffed bitstream to confirm correctness.

Binary to IPv4 Conversion:

1. Take a 32-bit binary IP address (like 11000000.10101000.00000001.00000001) as input from the user.
2. Divide the input into four 8-bit sections (octets).
3. Convert each 8-bit binary value into a decimal number.
4. Display the IPv4 address in decimal format (192.168.1.1).

4. IMPLEMENTATION

Experiment 1: Byte Stuffing & De-stuffing

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

```

#define FLAG 'F'
#define ESCAflE 'E'

void byteStuffing(char *input, char *output)
{
    int i = 0, j = 0;
    while (input[i] != '\0') {
        if (input[i] == FLAG || input[i] == ESCAflE)
            output[j++] = ESCAflE;
        output[j++] = input[i];
        i++;
    }
    output[j] = '\0';
}

void byteDeStuffing(char *input, char *output)
{
    int i = 0, j = 0;
    while (input[i] != '\0') {
        if (input[i] == ESCAflE)
            i++;
        output[j++] = input[i];
        i++;
    }
    output[j] = '\0';
}

int main() {
    char input[100], stuffed[200], destuffed[100];

    printf("Enter the bitstream (use 'F' for flag and 'E' for escape character): ");
    fgets(input, 100, stdin);
    input[strlen(input)] = '\0';

    byteStuffing(input, stuffed);
    printf("\nStuffed Bitstream: %s\n", stuffed);

    byteDeStuffing(stuffed, destuffed);
    printf("\nDe-stuffed Bitstream: %s\n", destuffed);

    return 0;
}

```

Experiment 2: Binary to IPv4 Conversion

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

int binaryToDecimal(char *bin)
{
    int decimal = 0;
    for (int i = 0; i < 8; i++) {
        decimal = decimal * 2 + (bin[i] - '0');
    }
    return decimal;
}

int main() {
    char binary[36], octet[9];
    int ip[4], j = 0, k = 0;

    printf("Enter a 32 bit binary lfl:");
    scanf("%35s", binary);

    for (int i = 0; i < strlen(binary); i++)
    {
        if (binary[i] == '.') continue;
        octet[k++] = binary[i];
        if (k == 8) {
            octet[k] = '\0';
            ip[j++] = binaryToDecimal(octet);
            k = 0;
        }
    }

    printf("lflv4 Address: %d.%d.%d.%d\n", ip[0], ip[1], ip[2], ip[3]);

    return 0;
}
```

5. TEST RESULT

Output for Byte Stuffing & De-stuffing

```
"C:\Users\MURSHALIN\OneDri" x + v
Enter the bitstream (use 'F' for flag and 'E' for escape character): ABCDEFFEFDFEG
Stuffed Bitstream: ABCDEEEFEFEFEFDEFEEG
De-stuffed Bitstream: ABCDEFFEFDFEG
Process returned 0 (0x0)   execution time : 22.781 s
Press any key to continue.
|
```

Output for Binary to IPv4 Conversion

```
"C:\Users\MURSHALIN\OneDri" x + v
Enter a 32-bit binary IP : 11000000.10101000.00000001.00000001
IPv4 Address: 192.168.1.1
Process returned 0 (0x0)   execution time : 17.413 s
Press any key to continue.
|
```

6. ANALYSIS AND DISCUSSION

Byte Stuffing & De-stuffing Analysis:

- Byte stuffing helps in ensuring that special characters (FLAG and ESCAPE) do not interfere with data transmission.
- The byte-stuffed data contains extra ESCAPE characters, which the receiver removes during de-stuffing to get back the original data.
- This method is commonly used in frame-based network protocols like HDLC.

Binary to IPv4 Conversion Analysis:

- The program correctly converts 32-bit binary IP addresses into decimal IPv4 format.
- Each 8-bit segment of the binary address is converted to a decimal number, forming an IP address like 192.168.1.1.

- This conversion is necessary because computers use binary, but humans use decimal IP addresses.

7. SUMMARY

1. Byte stuffing adds an ESCAPE character to prevent misinterpretation of FLAG (F) and ESCAPE (E) characters in data transmission.
2. Byte de-stuffing removes extra ESCAPE characters, restoring the original data.
3. Binary to IPv4 conversion is crucial in networking as it helps in human-readable IP addressing.
4. The experiment successfully demonstrated both techniques through C programming.
5. These concepts are widely used in data link layer protocols and IP networking for efficient and reliable communication.