

Lab 4: Bit Stuffing

Introduction:

This lab you will write a program which duplicates the work of the data link layer. When data is sent from the network layer, the data link layer checks to ensure that the value 7E does not exist in the string of bits. It runs a program called bit stuffing when it checks each bit and after 5 consecutive '1' bits, it adds a '0'. After bit stuffing, the data link layer then attaches the flags to delimit synchronous communication '01111110' to the beginning and end of the frame. The frame is then sent to the physical layer for encoding and transport.

Program Specifications:

You can write your program in any scripting or programming language you are comfortable with.

1. Create an array of the following characters '011111101111110'
2. Assume this is the data sent to the data link layer from the network layer. Notice that the combination of 0s and 1s conflicts with the synchronous flag.
3. Set up a loop to count the number of 1s and if there are 5 1s in sequence, insert a 0 into the data stream. Notice the data stream will require 2 'stuffed' bits.
4. After bit stuffing append the flag to the front and end of the data stream.
5. Print your output (colour is optional) but the your program output should be similar to the screen shot below.
 - a. Show the data stream before stuffing with the number of characters in the array
 - b. Show the data stream after stuffing with the number of characters in the array
 - c. Show the synchronous frame with the flag delimiters appended.

```
PS C:\Users\dhr> C:\Users\dhr\OneDrive\Documents\DCforProgrammers\bitstuffing4.ps1
Before Stuffing: 0 1 1 1 1 1 1 0 1 1 1 1 1 1 0 -- 15 characters
After Stuffing:  0 1 1 1 1 1 0 1 0 1 1 1 1 1 0 1 0 -- 17 characters
After framing: 0 1 1 1 1 1 1 0 0 1 1 1 1 1 0 1 0 1 1 1 1 1 0 1 0 0 1 1 1 1 1 1 0
PS C:\Users\dhr>
```

Grading:

- Send a copy of your screen shot. Name the file learnname_L4_BitStuffing.jpeg (replace **learnname** with **your name** for submission.) [1.875 Marks]
- Send your source code as a text file. Both code and screenshot of an output is mandatory to get a grade.
- Upload the files to the link on My Seneca\Graded Work.