Bit Stuffing / Destuffing

This bit stuffing is analogous to byte stuffing, in which an escape byte is stuffed into the outgoing character stream before a flag byte in the data. It also ensures a minimum density of transitions that help the physical layer maintain synchronization. USB (Universal Serial Bus) uses bit stuffing for this reason.

When the receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically destuffs (i.e. , deletes) the 0 bit. Just as byte stuffing is completely
transparent to the network layer in both computers, so is bit stuffing. If the user data contain the flag pattern, 01111110, this flag is transmitted as 011111010 but stored in the receiver's memory as 01111110. Figure 3-5 gives an example of bit stuffing.

With bit stuffing, the boundary between two frames can be unambiguously
recognized by the flag pattern. Thus, if the receiver loses track of where it is, all it has to do is scan the input for flag sequences, since they can only occur at frame boundaries and never within the data.

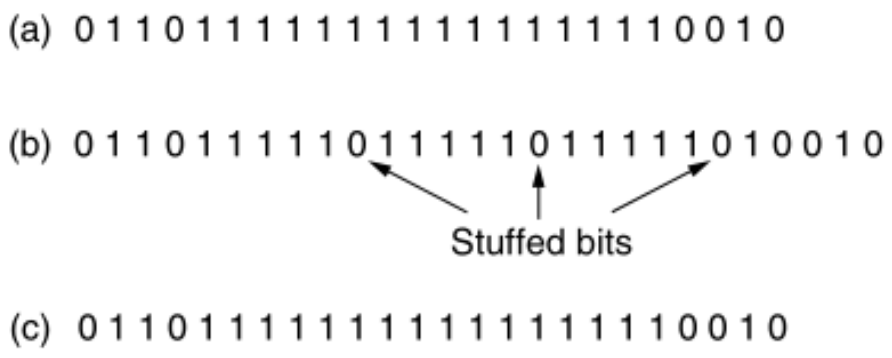


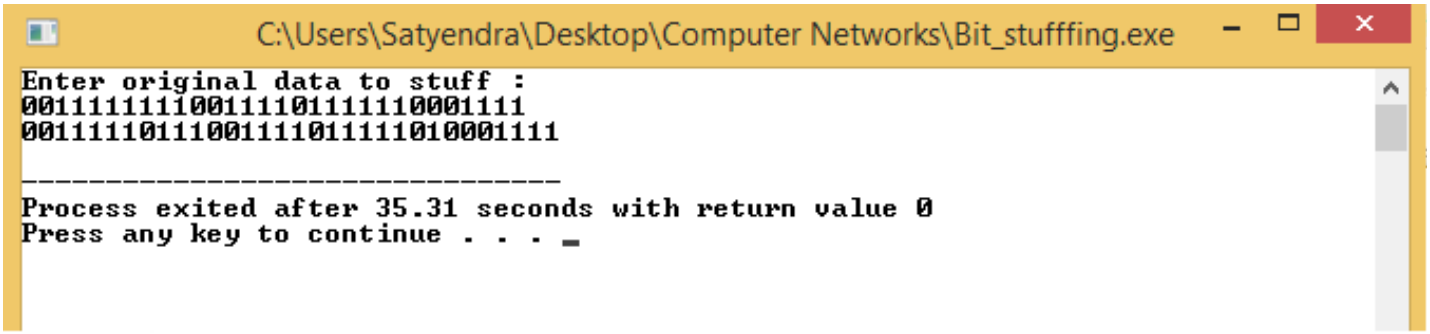
Figure 3-5. Bit stuffing. (a) The original data. (b) The data as they appear on the line. (c) The data as they are stored in the receiver's memory after destuffing.

Code of Bit Stuffing

```
#include<iostream>
#include<cstdio>
#define MAX_SIZE 100
using namespace std;

int main()
{
        bool Source[MAX_SIZE];
        int size=0,ptr=0;
        char c;
        cout<<"Enter original data to stuff :\n";
        c=getchar();
        while(c!='\n')
   {
      if(ptr==5)
         ptr=0,Source[size++]=0;
      if(c=='1')
         ptr++,Source[size++]=1;
      else
         ptr=0,Source[size++]=0;
      c=getchar();
   }
        for(int i=0;i<size;)
      cout<<Source[i++];
   cout<<endl;
}
```

RESULT : -



Code of Bit Destuffing

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#define MAX_SIZE 100
using namespace std;

int main()
{
        bool Source[MAX_SIZE];
        int size=0,ptr=0;
        char c;
        cout<<"Enter the data to destuff :\n";

        c=getchar();
        while(c!='\n')
   {
      if(ptr==5)
      {
         if(c=='1')
         {
            cout<<"\nError Message : This is not correct stuffed data.";
            cout<<"As stuffed data should have 0 after five 1's\n";
            exit(-1);
         }
         ptr=0;
         c=getchar();
      }

      if(c=='1')
         ptr++,Source[size++]=1;
      else
         ptr=0,Source[size++]=0;
      c=getchar();
   }

        for(int i=0;i<size;)
      cout<<Source[i++];
   cout<<endl;
}
```

RESULT : -

```
C:\Users\Satyendra\Desktop\Computer Networks\Bit_destufffing.exe
Enter the data to destuff :
00111110110011111011110001
00111111100111111110001

--------------------------------
Process exited after 28.4 seconds with return value 0
Press any key to continue . . .
```

```
C:\Users\Satyendra\Desktop\Computer Networks\Bit_destufffing.exe
Enter the data to destuff :
000111110111110111111011

Error Message : This is not correct stuffed data.As stuffed data should have 0 a
fter five 1's

--------------------------------
Process exited after 11.36 seconds with return value 4294967295
Press any key to continue . . .
```