Experiment No.3

Title- Implementation of Framing Techniques

1. Character count
2. Bit Stuffing

Theory: Data link layer translates the physical layers raw bit stream into discrete messages called frames. Frames are the units of digital transmission, particularly in computer networks and telecommunications.

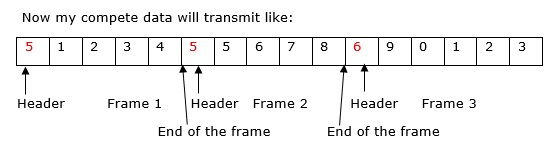
Framing techniques:

1. Character Count

This method is rarely used and is generally required to count total number of characters that are present in frame. This is be done by using field in header. Character count method ensures data link layer at the receiver or destination about total number of characters that follow, and about where the frame ends.

There is disadvantage also of using this method i.e., if anyhow character count is disturbed or distorted by an error occurring during transmission, then destination or receiver might lose synchronization. The destination or receiver might also be not able to locate or identify beginning of next frame

Explanation



1. Without Errors

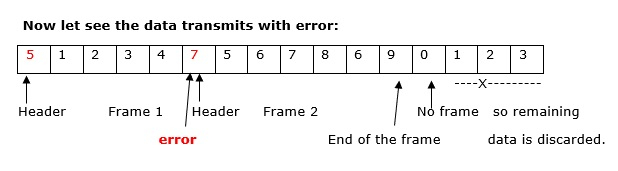
**Step 1** − Starting header in the frame indicate the character count, so first frame consists of 5 units of data including that number,

**Step 2** − Second frame header consists of 5 units of data including that number, so second frame consists of data 5,6,7,8. 8 indicate the end of the frame here.

**Step 3** − Third frame header consists of character count 6 that means the frame consists of 6 characters including 6. So the data in the third frame is 9,0,1,2,3.

**Step 4** − My data transfer to the receiver side without any errors.

1. With Error



**Step 1** − Starting header in the frame indicates the character count, so the first frame consists of 5 units of data including that number.

**Step 2** − Second frame header consists of 7 character count including that number actually it is an error, even though error is there the data will be transmitted, so second frame consists of data 5,6,7,8,6,9. Here, 9 indicate the end of the frame here.

**Step 3** − Third frame header consists of character count 0 that means the frame consists of 0 characters. The last frame data is discarded.

**Step 4** − My data transfer to the receiver side with errors.

1. Bit Stuffing

Definition

Bit stuffing is the mechanism of inserting one or more non-information bits into a message to be transmitted, to break up the message sequence, for synchronization purpose.

Purpose of Bit Stuffing

In Data Link layer, the stream of bits from the physical layer is divided into data frames. The data frames can be of fixed length or variable length. In variable - length framing, the size of each frame to be transmitted may be different. So, a pattern of bits is used as a delimiter to mark the end of one frame and the beginning of the next frame. However, if the pattern occurs in the message, then mechanisms needs to be incorporated so that this situation is avoided.

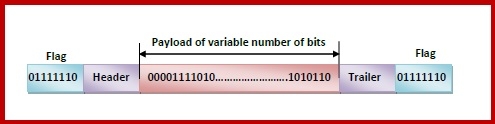
The two common approaches are −

* **Byte - Stuffing** − A byte is stuffed in the message to differentiate from the delimiter. This is also called character-oriented framing.
* **Bit - Stuffing** − A pattern of bits of arbitrary length is stuffed in the message to differentiate from the delimiter. This is also called bit - oriented framing.

Frame in a Bit - Oriented Protocol

In bit-oriented protocols, the message is coded as a sequence of bits, which are interpreted in the upper layers as text, graphics, audio, video etc. A frame has the following parts −

* **Frame Header** − It contains the source and the destination addresses of the frame.
* **Payload field** − It contains the message to be delivered.
* **Trailer** − It contains the error detection and error correction bits.
* **Flags** − A bit pattern that defines the beginning and end bits in a frame. It is generally of 8-bits. Most protocols use the 8-bit pattern 01111110 as flag.



Bit Stuffing Mechanism

In a data link frame, the delimiting flag sequence generally contains six or more consecutive 1s. In order to differentiate the message from the flag in case of the same sequence, a single bit is stuffed in the message. Whenever a 0 bit is followed by five consecutive 1bits in the message, an extra 0 bit is stuffed at the end of the five 1s.

When the receiver receives the message, it removes the stuffed 0s after each sequence of five 1s. The un-stuffed message is then sent to the upper layers.

