**FRAME FORMAT**

The **data link layer** splits long messages into a number of small segments which are small enough for the physical layer to handle. Each segment send is a packed framed consisting of various field which help in easier transmission and error detection.

The **frame format** used in the following implementation **doesn’t** represent the actually standard format used in the real world. The format used here is just for the convenience of the user and to understand the formatting step easily.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Fields | Frame delimiter  (start) | Sequence  No. | Field Divider  (-) | Frame type  (N/E) | Field Divider  (-) | Payload length  (decimal) | Field Divider  (-) | Payload  (Actual Message) | Field Divider  (-) | Error Detection  (Checksum) | Frame Delimiter  (End) |
| Example | < | 01 | - | N | - | 06 | - | nikhil | - |  | > |

**9 chars 11 chars 6 chars**

**Explanation**

|  |  |  |
| --- | --- | --- |
| Fields | Characters | Description |
| Frame Delimiter (Start) | 1 | Represent the start of the frame with symbol ‘<’ |
| Sequence Number | 2 | Tells about the frame number. (Always a 2-digit decimal) |
| Field Divider | 1 | A single ‘-‘ to divide the various fields in the frame. |
| Frame Type | 1 | Tells whether frame is end frame or not.  N = Normal Frame which is not a final segment of the message.  E = End frame which is the final segment of a message. |
| Field Divider | 1 | A single ‘-‘ to divide the various fields in the frame. |
| Payload Length | 2 | No. of characters in the payload (actual message) represented using 2-digit decimal number. |
| Field Divider | 1 | A single ‘-‘ to divide the various fields in the frame. |
| Payload (Actual Message) | 0-99 | Actual message, must not exceed the 99-char limit. |
| Field Divider | 1 | A single ‘-‘ to divide the various fields in the frame. |
| Checksum | 5 | Any error detection module implemented. |
| Frame Delimiter (End) | 1 | Represent the start of the frame with symbol ‘>’ |

**Example**

Let us assume the **MTU** (Maximum Transmission Unit) has size of **24** chars. Let the message to be sent is

**My name is Nikhil. I live in Jammu and Kashmir. I stay in Kolkata.**

Size of message = 66 chars

Since every frame has some prefix and suffix.

Prefix = <01-N-11- = 9 chars

Suffix = -abcd> = 6 chars

Maximum message length = 9

**FORMAT**

startOfFrame-SeqNo-TypeOfFrame-LengthOfMessage-Message-Checksum-endOfFrame

Hence the message will be divided in 8 frames.

Frame1=<00-N-09-My name i-cb21>

Frame2=<01-N-09-s Nikhil.-6973>

Frame3= <02-N-09- I live i-f04a>

Frame4= <03-N-09-n Jammu a-648f>

Frame5= <04-N-09-nd Kashmi-a706>

Frame6=<05-N-09-r. I stay-d83a>

Frame7=<06-N-09- in Kolka-b93a>

Frame8=<07-E-03-ta.-8b70>