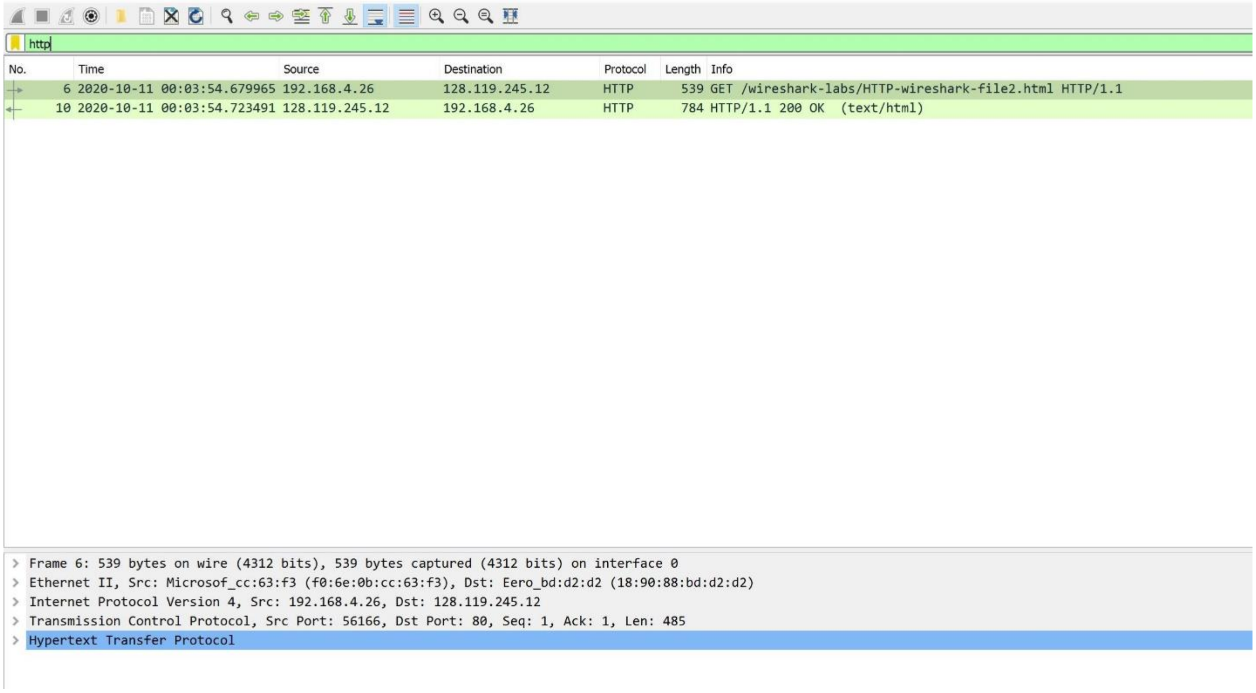
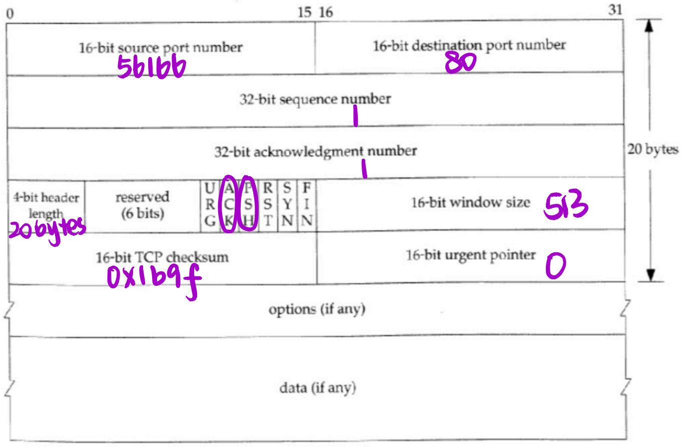
1.



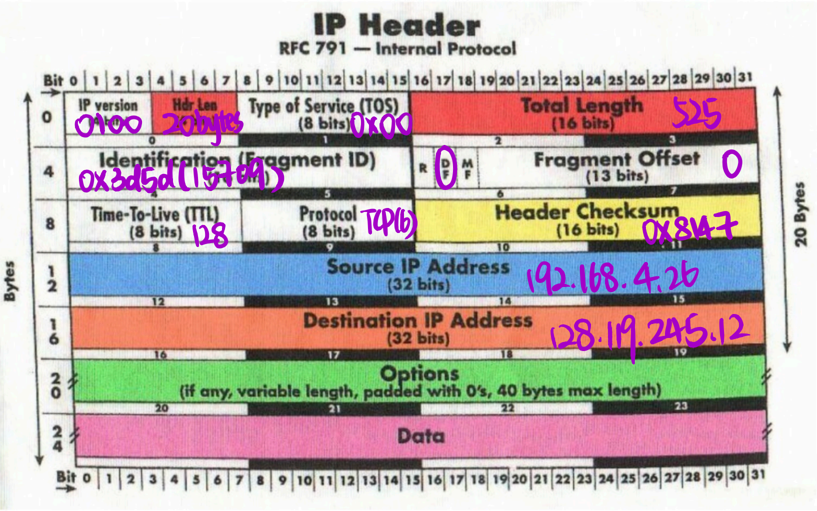
```
http
No.   Time                        Source          Destination     Protocol  Length  Info
    6 2020-10-11 00:03:54.679965 192.168.4.26     128.119.245.12  HTTP       539 GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1
   10 2020-10-11 00:03:54.723491 128.119.245.12   192.168.4.26    HTTP       784 HTTP/1.1 200 OK  (text/html)

> Frame 6: 539 bytes on wire (4312 bits), 539 bytes captured (4312 bits) on interface 0
> Ethernet II, Src: Microsof_cc:63:f3 (f0:6e:0b:cc:63:f3), Dst: Eero_bd:d2:d2 (18:90:88:bd:d2:d2)
> Internet Protocol Version 4, Src: 192.168.4.26, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 56166, Dst Port: 80, Seq: 1, Ack: 1, Len: 485
> Hypertext Transfer Protocol
```

1. TCP Header

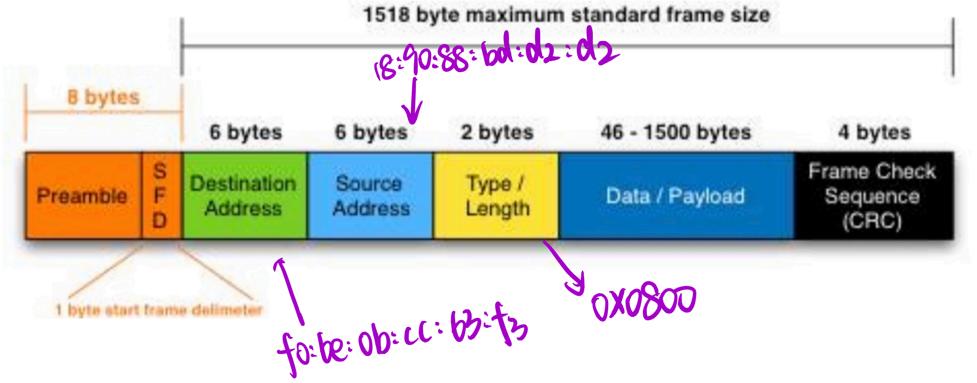


3. IP Header


IP Header
RFC 791 — Internal Protocol
Bit 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
IP version
0100
Hdr Len
20bytes
Type of Service (TOS) (8 bits)
0x00
Total Length (16 bits)
525
Identification (Fragment ID)
0x3d5d(15709)
R
DF
MF
Fragment Offset (13 bits)
0
Time-To-Live (TTL) (8 bits)
128
Protocol (8 bits)
TCP(6)
Header Checksum (16 bits)
0x8147
Source IP Address (32 bits)
192.168.4.26
Destination IP Address (32 bits)
128.119.245.12
Options (if any, variable length, padded with 0's, 40 bytes max length)
Data
Bytes
0
4
8
12
16
20
24
20 Bytes

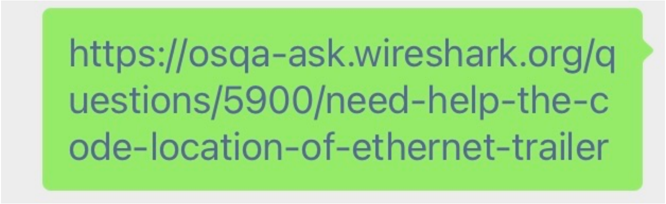
4. Ethernet Frame Format

IEEE 802.3 Standard Ethernet Frame



I cannot find the Data Link Trailer in the Ethernet frame capture in Wireshark. Wireshark assume the Ethernet trailer as the padding which is to make the packet length up to the minimum. If the packet does not require a trailer, or if we are unable to determine whether it is necessary to add a trailer which we cannot guarantee all extra data at the end of the packet will be recognized or treated as a trailer.

Resource from



https://osqa-ask.wireshark.org/questions/5900/need-help-the-code-location-of-ethernet-trailer

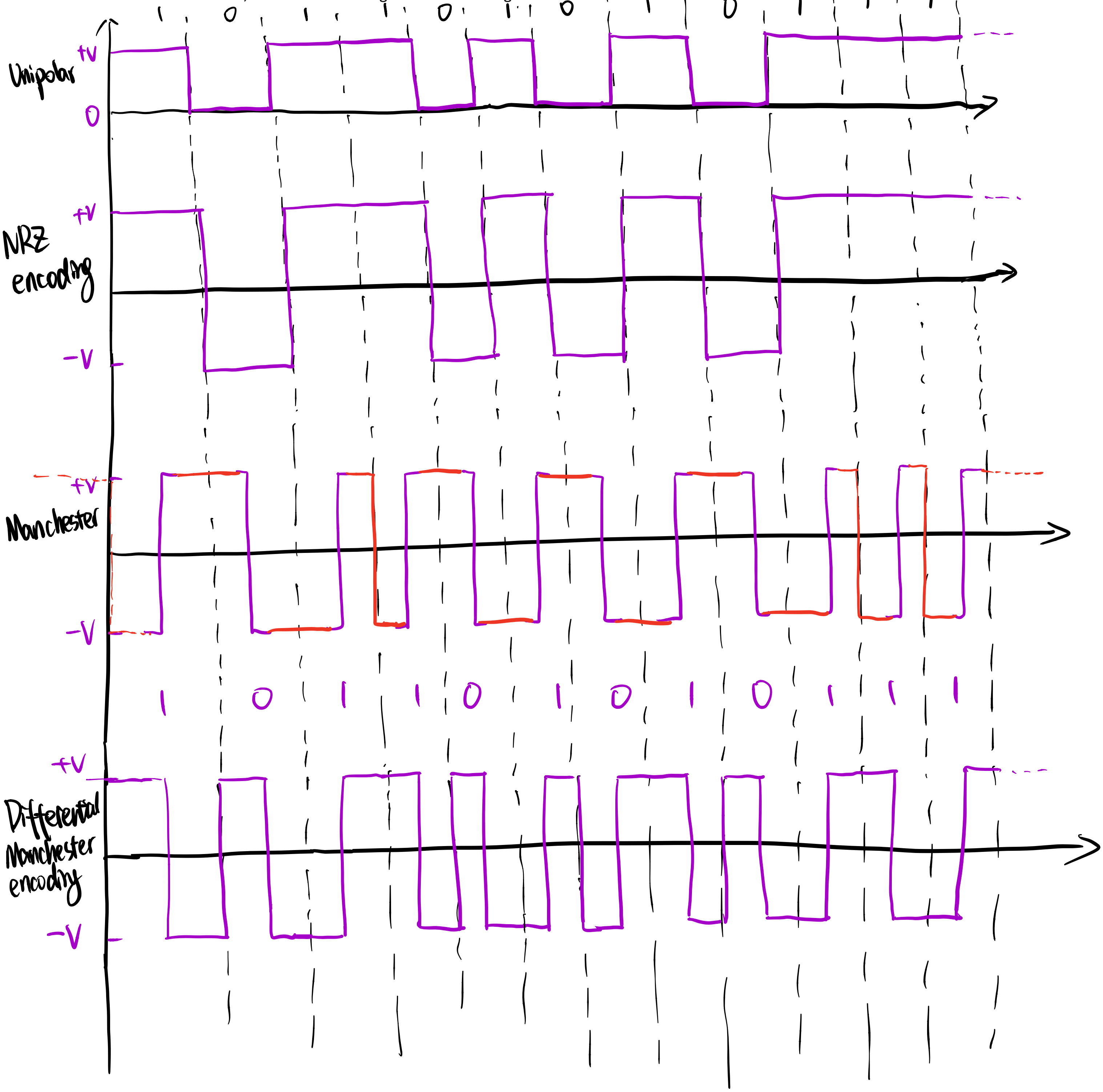
2. <Digital Encoding Question> The following bit stream is to be digitally encoded:

   1 0 1 1 0 1 0 1 0 1 1 1

   Draw the waveforms if the bit stream were to be encoded using

   a) Unipolar
   b) NRZ
   c) Manchester ✓ 0: high → low 1: low → high
   d) Differential Manchester encoding: In this scheme, logic 0 is represented by a transition at the beginning AND at the middle of the clock interval. The transition can be from low to high or high to low, that is, if it was low, it goes to high and if it was high, it goes to low. Logic 1 is represented by a transition ONLY at the middle of the interval. Again the transition can be either low to high or high to low.

   In all the above cases, assume that the signal is HIGH to begin with.



3. <Bit Stuffing Question>

a) The following message is to be sent by a host running a protocol with starting and ending flags and bit stuffing. The starting and ending flags are both 01111110 and they have ***not yet been added***.

01111110111111011110011111100111111000000111110101111110

What is the message actually sent (after bit stuffing and after adding the starting and ending flags)?

b) Suppose the bit pattern shown above is received by a host running the bit stuffing protocol. That is, this is the actual message that has been received after bit stuffing and after adding the starting and ending flags. How many frames are being received? What is the actual content of each frame before the flags are added and the bits are stuffed?

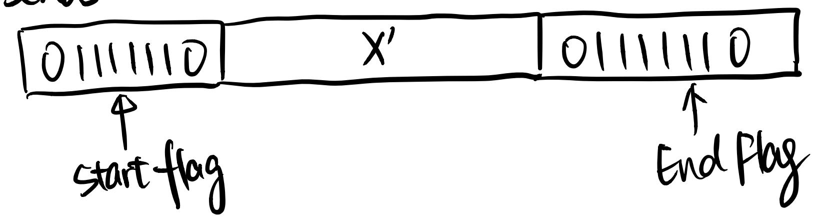
suppose X =

a) 0 111111 0 111111 0 1111 00 111111 00 111111 000000 11111 010 111111 0

0 11111 0 10 111111 0 0111100 11111 0 100 11111 0 1000000 11111 0 010 11111 0 10

X'

Sender sends



b) 0111111 0 11111 0 1111 00 111111 00 111111 000000 11111010 111111 0

Start flag

received frames

End Flag

11111 0 1111 00 111111 00111111 000000 11110 1

actual content 11111 1111 00 11111 0 111111 00000 11111