Protocol Header Analysis

Objective:

- Understand the format of standard frames and packet headers.

1 Overview

Refer to the following layered architecture of the Internet protocol stack.

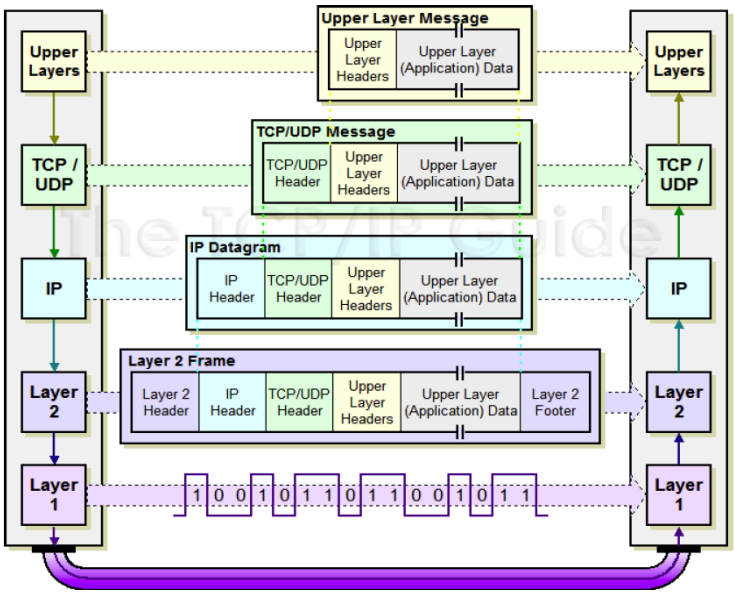

Figure 1: Layered architecture

In this lab assignment, you will be asked to interpret the encapsulated headers of captured Ethernet frames.

2 Background Material

2.1 Ethernet Frame

Figure 2 shows the format of Ethernet frames sent and received by the data link (MAC) layer. The preamble bits are not shown. If a frame is received without bit errors, the "Data" portion is passed on to the upper layer (network layer).

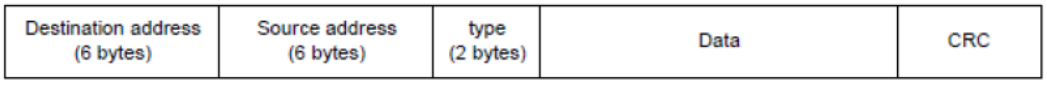

Figure 2: MAC layer frame format

2.2 IP/TCP/UDP Header

The IP protocol is defined in RFC 791 (RFC: Request for Comment), and a summary of the IP header is given in Figure 3. The number on the top is the bit number and each row is fourbyte long. Figures 4 and 5 show the format of the headers of TCP and UDP, respectively. They are defined in RFC 793 and RFC 768, respectively. All the RFCs can be found at http://www.ietf.org/rfc.html. The numbers on top again represent the bit number and each row is four-byte (32-bits) long. You will also need to refer to the ICMP protocol (RFC 792) and tell us what is the highest protocol (e.g., FTP, HTTP, etc.).

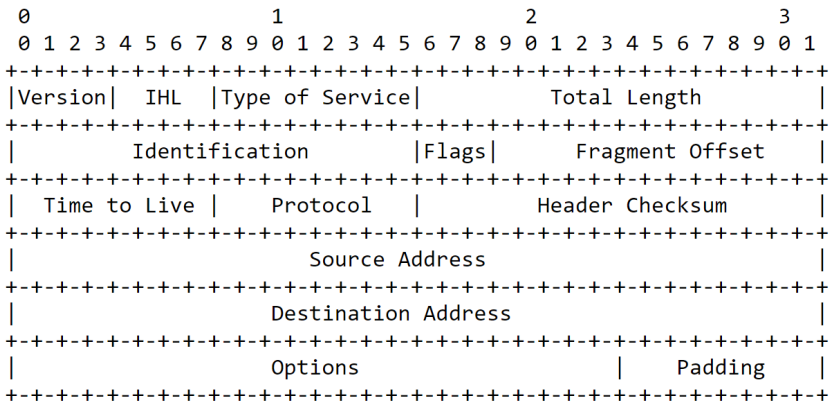

Figure 3: Example Internet Datagram Header

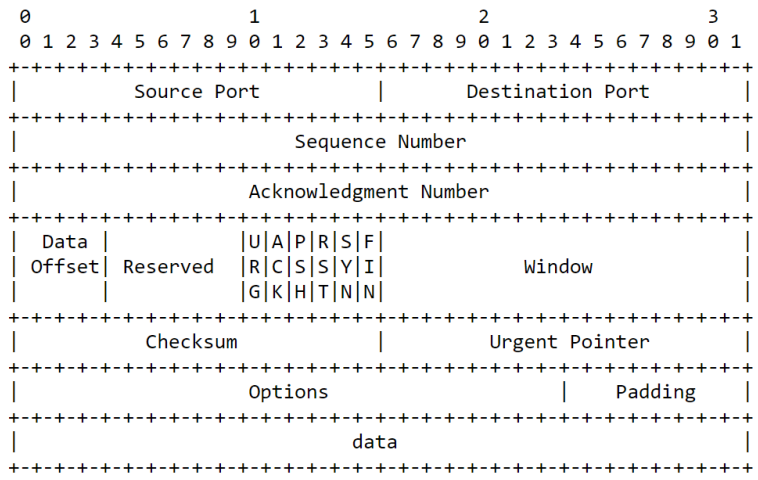

Figure 4: TCP Header Format

2.3 Protocol Header Analysis

The analysis of a sample MAC frame is being shown below.

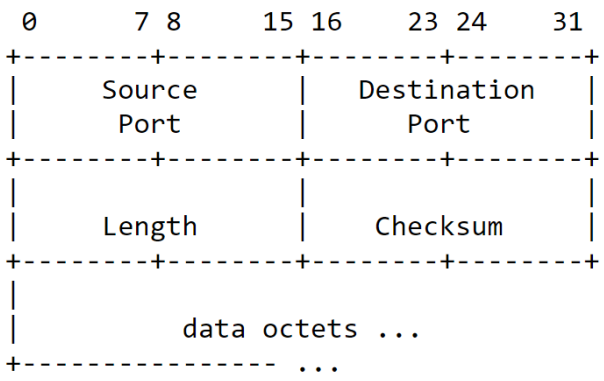

Figure 5: UDP Header Format

Sample frame:

```
00 00 0c d9 fa 88 00 00 b4 a0 15 c1 08 00 45 00
00 28 04 04 40 00 80 06 42 a0 80 d3 a0 3c 80 0a
13 14 04 3a 00 15 54 f1 f2 09 d6 7d df 9d 50 10
40 5a b9 e8 00 00
```

Ethernet header:

`00 00 0c d9 fa 88`: Ethernet destination address is 00 00 0c d9 fa 88 (unicast).
`00 00 b4 a0 15 c1`: Ethernet source address: 00 00 b4 a0 15 c1 (unicast).
`08 00`: The payload type is IP (0x0800). (Note: 0x0806 is ARP.)

IP header:

`45`: This is an IP version 4 datagram,
`45`: The header length is 5x4 = 20 bytes. (There is no *options* field in the given IP header).

`00`
(`0 0 0 0 0 0 0 0` in binary): This datagram has routine precedence (the lowest). The IP Precedence field is used by some routers to determine which datagram to drop, therefore datagrams with the lowest precedence will be dropped first.

(0 0 0 0 0 0 0 0 in binary): the 3 type of service (ToS) bits
0 0 0 *Normal delay*
0 0 0 *Normal throughput*
0 0 0 *Normal Reliability*
(0 0 0 0 0 0 0 0 in binary): The last two bits must be zero (for future use).

00 28: Total length of the IP datagram is 40 (0x0028) bytes.
04 04: The identification of this datagram is 0x0404 (for fragmentation purpose).
40 00: (0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0):
1 Don't Fragment flag set
0 More Fragment flag unset
The Fragment offset is 0.
This means that the datagram cannot be fragmented, and there are no fragments after this datagram. With a fragment offset equals to zero, we know that this is the only fragment of a datagram.
80: Time to live = 128 (0x80), meaning the datagram may exist for *at most* 128 more hops.
06: The Protocol on top is TCP (0x06) (Note: 0x01 is ICMP and 0x11 is UDP).
42 a0: This is the checksum of the datagram.
80 d3 a0 3c: Source IP address is 128.211.160.60.
80 0a 13 14: Destination IP address is 128.10.19.20.

TCP header:

04 3a: The Source port is 1082, which is an arbitrarily port number assigned by the operating system.
00 15: The Destination port is 21, which is the well-known port for FTP (File Transfer Protocol).
54 f1 f2 09: The Seq. no. is 1425142281.
d6 7d df 9d: The Ack no. is 3598573469.
50: Data offset is 20 (5 x 4) bytes. This is the length of the TCP header.
10 (0 0 0 1 0 0 0 0):
Flags: URG 0 ACK 1 PSH 0
RST 0 SYN 0 FIN 0
Only the ACK flag is set, meaning that the value carried in the acknowledgement field is valid. **(You should comment on all the flags that are set, i.e., equal to 1)**
40 5a: the receiver window size is 16474 (0x405a) bytes.
b9 e8: Checksum of the whole TCP segment.
00 00: Urgent pointer (Not used in this segment).

3 Capture and Analysis

In this assignment, you will use Wireshark to analyze one TCP frame. The following steps show you how to capture a TCP frame.

(1) First, install the Wireshark on your own computer.

(2) Use Wireshark to capture TCP frames: Turn off any network-intensive programs you may already have running. Then, tell Wireshark to start capturing frames on the active network interface. Then, you should do **something** that will cause your host (computer) to send and receive several TCP frames. After you see a few TCP packets in your capture window, stop capture before you have a massive file!

(3) Set your packet filter so that Wireshark only displays TCP frames sent and received at your host.

Pick one of these TCP frames. Using the example given in section 2.3 as a template, parse the frame in a human readable format and comment. For example, write an IP address in the dotted decimal notation and header length as a positive integer. Also, color (or, underline) the different parts of the frames to indicate their layers: 2, 3, 4, or application data and indicate the name of the highest layer protocol. Submit your report (one single PDF file) on Moodle.