ECE 358 - Project 3
Encapsulation and Network Utilities
(Date of last revision: Nov. 12, 2014)

Table of Contents

A. Objective
B. Overview
C. Background Material
(1) Ethernet Frame
(2) IP/UDP/TCP Header
(3) Sample Frame
(4) Use of *Wireshark* (a software network traffic analyzer, which was known as Ethereal before)
(5) Network Utilities
D. Questions
E. Final Report

A. Objective

After this project, students are expected to have:

- understood the format of standard frames and packet headers;
- learnt how to use basic network utilities to monitor network traffic; and
- learnt how to use *Wireshark* to analyze frames.

B. Overview

A fundamental concept in networking is encapsulation. In this project, you will be asked to interpret the headers of TCP segments encapsulated in IP datagrams which in turn are encapsulated in Ethernet frames. You will also get an opportunity to use some network utilities to get an idea about the performances of the network.

C. Background Materials

I. Ethernet Frame

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

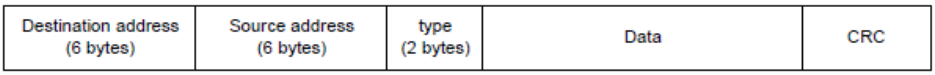

Figure 2: A sample of Ethernet frame

II. 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 four bytes 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 bytes (32-bits) long. You will also need to refer to the ICMP protocol (RFC 792). Figure 3 has been explained in class, and Figures 4-5 will be explained in due course.

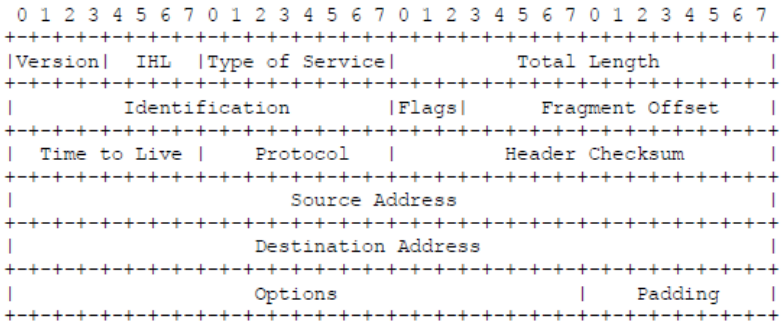

Figure 3: Example Internet Datagram Header

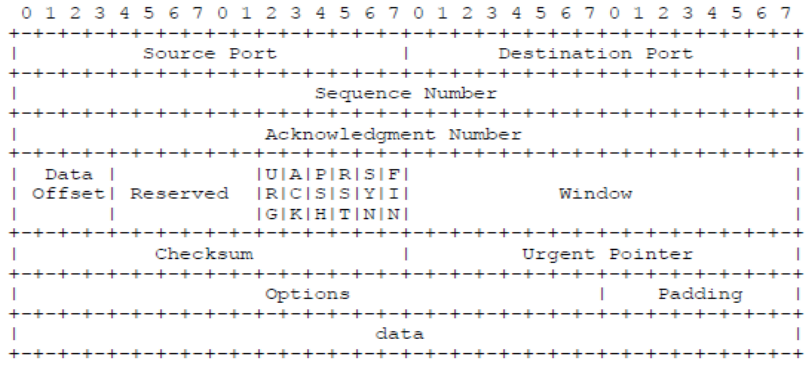

Figure 4: TCP Header Format

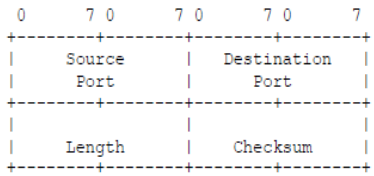

Figure 5: UDP Header Format

III. Protocol Header Analysis

In this project, refer to the following frame/packet formats for analysis:

1) Ethernet frame header (14 bytes): Figure 2
2) IP header: Figure 3
3) Transport layer headers (TCP or UDP): Figures 4-5

Analysis of a sample MAC frame has been shown in the remains of this section.

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 0 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).

Data: none

Overall comment on the given frame: The given example frame contained a pure TCP ACK (no data). We observe a lot of those when we monitor the Internet. There may be data in a frame, so you just need to highlight the data portion without analyzing it. You should try to include as much information about the frame as possible. Do not try to analyze the TCP options (see Figure 4).

IV. Use of Wireshark

Wireshark (formerly known as *Ethereal*) is a free network protocol analyzer for Unix and Windows systems. It enables you to analyze data from a live network or from a stored file on a disk. You can interactively browse the captured data and view the summary and detail information for each packet. For details, visit http://www.wireshark.com. Since capturing frames requires super-user privilege, the capturing function of *Wireshark* is, in general, not available to ordinary users. However, it can still be used as a reader to read pre-captured traces in a user-friendly manner.

Visit this site for useful information: http://wiki.Ethreal.com/Ethrealwiki/SampleCaptures

V. Network Utilities

In this project, you will use the following network utilities:

- `arp`
- `ifconfig`
- `nslookup`
- `netstat`
- `ping`
- `traceroute (tracert)`

Detailed information about each utility can be obtained from the Internet. Also, you can find information about the utilities by using the ***man*** command on Unix/Linux machines. You can also access the Linux servers: ecelinux.uwaterloo.ca and ecelinux2.uwaterloo.ca remotely via SSH.

==========

D. Questions

I. Protocol Header Analysis

Question 1: Obtain two frames in bytes (They actually appear as bits, though). Parse the frames in a human readable format. For example, write an IP address in the dotted decimal notation and header length as a positive integer. Any hexadecimal values should be preceded by 0x. Also, color (or, underline) the different parts of the frames to indicate their layers: 2, 3, 4, or app data. An example has been given in the Background section.

II. Use of Wireshark

Question 2: Obtain two Wireshark traces. You will be able to see all the header information of the trace under Wireshark. Describe what is happening in the trace by means of a diagram. What is the round-trip time between the IP source and the IP destination?

III. Network Utilities

Question 3: (arp)

(a) Explain the functions of the utility.

(b) Use the command `/sbin/arp -a` to see the ARP table of the machine. Include the output of the command in your report and explain it. What is the MAC address of exsw02-circuitnet.uwaterloo.ca (129.97.56.1)?

Question 4: (ifconfig)

(a) Explain the functions of the utility.

(b) Use the command `/sbin/ifconfig -a`. Include the output in your report and explain it.

Question 5: (netstat)

(a) Explain the functions of the utility.

(b) Use the command `netstat -in`. How many packets are sent from and received by interface `eth0`?
(c) Use the command `netstat -r`. Include the output in your report and explain it.

Question 6: (nslookup)
(a) Explain, in your own words, what the utility does.
(b) Use the command to obtain the IP addresses of the following hosts and explain what you get.
1. `www.uwaterloo.ca`
2. `www.youtube.com`
3. `www.gmail.com`
4. `www.facebook.com`
5. `www.brasil.gov.br`

Question 7: (ping)
(a) Explain the functions of the utility.
(b) Use `ping -c5 hostname` to estimate the average round-trip-time from ecelinux1.uwaterloo.ca to the following hosts. Include the output in your report and explain what you get.
1. `www.uwaterloo.ca`
2. `www.youtube.com`
3. `www.gmail.com`
4. `www.facebook.com`
5. `www.brasil.gov.br`

Check if each host above is up by using a web browser to connect to the hosts.

Question 8: (traceroute)
(a) Explain the functions of the utility.
(b) Use `/usr/sbin/traceroute hostname` to find out how many hops are there between the host and the following hosts. Include the outputs in your report and explain what you get. If the full path name does not work, just type in the command name.
1) `www.uwaterloo.ca`
2) `www.youtube.com`

E. Final Report

Submit a print copy of your report.
Provide a cover page.
Give all the details as specified.
If you use color coding, print the pages in color. If you do not want to print it in color, use other techniques to identify the different blocks of bytes in MAC frames.