Name: _____ Section: _____

HW Partner (if any):_____

HW #11 – Packet Parsing

IC322 Computer Networks

Due: 8 Nov

This homework may be completed alone or in pairs. Each student must hand in their own sheet, filled out themselves to get credit.

In this assignment you will examine a raw Ethernet frame in hex format. You will identify the parts of the frame that make up the Ethernet header, the IP header, the TCP header, and the payload. Additionally, you will locate elements of each header/payload, and convert them to a human-readable format. The following hex data is a complete Ethernet packet sent from one computer to another. You will mark the pieces of this hex throughout the homework:

```
AA AA AA AA AA AA AA AB
00 0D ED A6 EB BF 00 26
B9 E8 FF C3 08 00 45 00
00 3D 00 01 00 00 40 06
55 87 0A 35 21 D9 AD C2
4B 63 04 72 00 50 00 00
00 00 00 00 00 00 50 02
20 00 A6 C1 00 00 47 45
54 20 2F 6D 61 70 73 20
48 54 54 50 2F 31 2E 31
20 0A 0A D0 C3 77 26
```

Required tools:

- Yellow highlighter
- Pink highlighter
- Green highlighter

Remember the following conversions:

`'F'` (single hex character) == 4 bits

`'FF'` (octet) == 8 bits

`'FF FF FF FF'` (word) == 32 bits

1. Identifying the Layer-2 portions of the message

This is the format for an Ethernet message. Each number indicates a single octet.

Preamble								Destination MAC						Source MAC						EtherType/ Size		PayLoad					CRC			
1	2	3	4	5	6	7	8	1	2	3	4	5	6	1	2	3	4	5	6	1	2						1	2	3	4

Do the following:

(a) Circle the Preamble in the hexdump on page 1

(b) Enter the first octet of the preamble in hex: _____ Convert it to binary: _____

(c) Enter the last octet of the preamble in hex: _____ Convert it to binary: _____

(d) Circle the destination MAC address

(e) Underline the source MAC address

(f) The first three octets of a MAC address indicate the manufacturer. Go to this web page: http://standards.ieee.org/develop/regauth/oui/oui.txt and find the manufacturer for the source computer's network card _____ and the destination computer's network card _____.

(g) Circle the Ethertype.

(h) Go to this page: http://en.wikipedia.org/wiki/Ethertype and find out what type of message is stored in the Ethernet frame: _____.

(i) Circle the CRC.

(j) Highlight the four Ethernet header elements and the footer (CRC) in Yellow. Do not highlight the payload.

2. Identifying the Layer-3 portion of the message

The IP header begins immediately where the Ethernet header ends. Use this page as a reference: http://en.wikipedia.org/wiki/IPv4#Header

(a) Enter the IP version number: _____

(b) Enter the length of the IP header in words: _____

(c) Use the result from (b) above to figure out where the IP header ends. Highlight the entire IP header in Pink.

(d) Find the value in hex in the Protocol field: _____

(e) Go to this page http://en.wikipedia.org/wiki/List_of_IP_protocol_numbers to find out what type of message is stored in the IP payload: _____

(f) Find the source IP address. Convert it into "dotted quad" format (e.g. 10.10.1.22) and enter it here: _____

(g) Find the destination IP address and convert it to dotted quad: _____

(h) Go to http://www.networksolutions.com/whois/index.jsp and run a "Search by IP Address" on this IP. What company owns it? _____

3. **Identifying the Layer-4 portion of the message**
   The TCP header begins immediately where the IP header ends. Use this page as a reference: http://en.wikipedia.org/wiki/Transmission_Control_Protocol

   (a) Circle the source port.
   (b) Circle the destination port.
   (c) Convert the destination port to decimal (base-10) and enter it here: _____
   (d) Use this page http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers to figure out what Layer-7 protocol is usually requested from this port: _____
   (e) Circle the Data Offset value in the TCP header.
   (f) Convert the Data Offset value to decimal and enter it here: _____
   (g) Use the Data Offset value to find the extent of the TCP header. Highlight the entire TCP header in Green.

4. **Identifying the Layer-7 portion of the message**
   The remaining non-highlighted portion of the message is the Layer-7 payload. This data is interpreted by the application based on the protocol we determined in 3(d) above.

   (a) The protocol in this particular message is stored in hexadecimal ascii format. Convert it to human readable text and write out the message below. (You can use this page to convert by hand: http://www.asciitable.com/ or you can find a faster way to automate this conversion if you prefer.)

   (b) This Layer-7 protocol is requesting a webpage from the remote server. See page 103 (section 2.2.3) in your textbook if you need help understanding the request. Use a browser to surf to this same page on the server. Describe the webpage you see here: _____

Once you have made it to this webpage, go back and take a look at the original hex dump of the Ethernet frame. It probably still looks like an eye-chart, but you have now seen how to pull information out of it.

Let's do some quick statistical analysis:

Count the total number of hex characters in the original message: _____

Count the number of hex characters in the Layer-7 payload: _____

What percentage of hex characters in the message is Layer-7 payload? _____

This should give you some understanding of how much administrative overhead is required to move information across a network.

When you surfed to the webpage yourself in 4(b), your browser generated an Ethernet frame that looked very similar to the one you have been analyzing here. Most of the information the browser needed to generate that request was already on your computer. (e.g. it knew the local IP, it knew the local MAC, it knew that HTTP uses TCP port 80, it could request a local port number from the OS, etc.) There were two pieces of information that it had to send messages to find from remote machines.

1. **Remote IP address.** Most people prefer to type human-readable URLs like www.google.com rather than memorizing the dotted-quad for every website they visit. DNS servers maintain lookup tables that map URLs to IP addresses. Open a terminal shell and type "`nslookup www.google.com`". You should see a list of several IP addresses come back. Re-run the same command several times. (Use the up-arrow.) The order of the addresses should change. All of these IP addresses go to google.com, but the DNS server re-orders them each request to spread the load. Your browser grabs the first IP from the list and uses it as the Destination IP address. Enter the top IP address in the list here: _____

2. **MAC address of next routing hop.** The Ethernet frame MAC addresses change for each leg in the route. Since the Google IP address is on a different network than the local machine, our first stop must be at the default gateway router. We need to find its MAC address to fill in the Layer-2 header. Our local machine has its gateway's IP address stored locally. You can find it by running "`ipconfig`" at the command-line. Enter your gateway IP address here: _____

   The local network uses Address Resolution Protocol (ARP) to keep an updated map of IP to MAC addresses. Type "`arp –a`" and look up the MAC address of the default gateway router here: _____.

   The gateway address should already be in your local ARP table since you were recently online. Let's force your machine to send an ARP request to add a new one. Find an IP address that is not in your ARP table. (Try one of the lab computers or just guess a new one based on the IPs you can see.) Ping that new IP address. If you get a valid response, then run "`arp –a`" and you should see the IP/MAC pair added to the list. Your computer quietly finds these MAC addresses in the background every time you need to access a new IP address.