Instructor: Dr. Shaji Khan (shajikhan@umsl.edu or 314-489-9733)
Student: Omar Salih

INFSYS 3842/6836

Assignment 2 (Lab): Understanding layering and Ethernet MAC Frames Using Wireshark

Points Possible: 100

Due Date: March, 6 2016 by 11:59pm Central Time

IMPORTANT NOTE: THIS LAB MUST BE CARRIED OUT ON YOUR OWN COMPUTER *AND* OWN NETWORK. PLEASE DO NOT CAPTURE PACKETS ON A NETWORK THAT YOU DO NOT OWN. YOU'VE BEEN WARNED!

Lab Overview: It is important we are able to understand the idea behind "layers of functionality" provided by different protocols that work together at different levels to accomplish data networking. Capturing some network traffic using Wireshark and carefully analyzing the packets is a great way to learn about layered functionality as well as the syntax of some key protocols such as Transmission Control Protocol (TCP), Internet Protocol (IP), and Ethernet as well as some common application layer protocols such as Hyper-Text Transfer Protocol (HTTP)

Lab Purpose:

1) To download and install Wireshark on students' personal computers
2) To learn the basics of how to use Wireshark to capture network traffic (from students own computers and own networks)
3) To learn about basic "Capture Filters" available in Wireshark
4) **Understand the syntax of Layer 2 Frames**.

Lab Tasks: There are two tasks for this lab.

TASK 1

Download and install Wireshark on **OWN** Computer. Visit http://www.wireshark.org and download the latest version. Installation may vary a bit depending on your operating system. The best source of help is the Wireshark wiki (https://wiki.wireshark.org) but it can be a bit cryptic for beginners. There are plenty of other tutorials/videos online in case you need help.

TASK 2: (This task has five questions)

In this task you will use Wireshark to capture basic HTTP traffic and complete the activities and questions as described below.

I recommend you watch some videos on YouTube on capturing HTTP traffic using Wireshark. The process is fairly simply as demonstrated in class but feel free to learn more.

Also familiarize yourself with the basic interface of Wireshark (the menus, options, **filters** etc.). Again, plenty of videos are available online and the Wireshark wiki is best if you really want to learn.

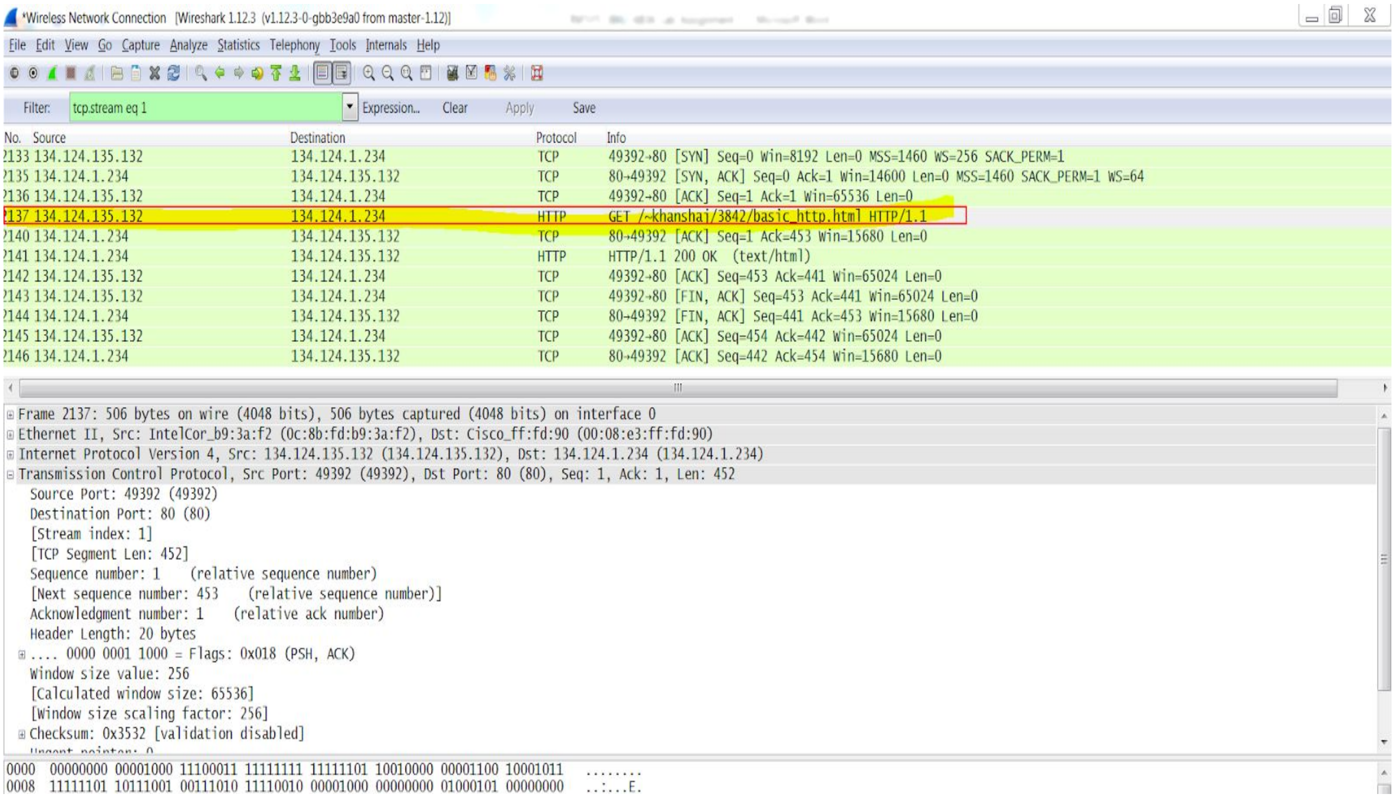
STEPS:

1) Open Wireshark and identify the interface you will capture traffic from (Wireless or Ethernet LAN)

LAB Question 1: How many interfaces does Wireshark recognize your computer has? What types of interfaces are they (Wired Ethernet LAN/ Wireless / Virtual?)? **Ethernet 2, ip, tcp, http**

CONTINUE:

2) Close all browser windows and other applications. Also, clear your browser's cache/history/temporary files. In Internet Explorer hit (Ctrl + Shift + Delete).
3) Open a browser window and type in http://www.umsl.edu/~khanshaj/3842/basic_http.html. **DO NOT Press Enter yet**.
4) Go back to Wireshark and **Start** a capture.
5) Go back to the browser window and now hit enter to visit the page.
6) Once the page loads, return to Wireshark and **Stop** the capture. **Save the capture on your computer. Call it "BasicHTTPCapture"**.
7) Examine the packets captured and scroll to find "green colored rows" that denote "TCP" based traffic. Notice the Three-step handshake and the HTTP requests and data responses from the server, acknowledgements, and four step closing.
8) FIND THE PACKET that belongs to the **HTTP Get** request asking for the basic_http.html file. See picture below. Once located, right click on that packet and choose "Follow TCP Stream". This should remove all other packets allowing you to focus on just this TCP session. Another window showing the HTTP requests and responses should also open. You can minimize it.


*Wireless Network Connection [Wireshark 1.12.3 (v1.12.3-0-gbb3e9a0 from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save

Filter: tcp.stream eq 1

```
No.  Source            Destination       Protocol  Info
2133 134.124.135.132   134.124.1.234     TCP       49392→80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
2135 134.124.1.234     134.124.135.132   TCP       80→49392 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1 WS=64
2136 134.124.135.132   134.124.1.234     TCP       49392→80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
2137 134.124.135.132   134.124.1.234     HTTP      GET /~khanshaj/3842/basic_http.html HTTP/1.1
2140 134.124.1.234     134.124.135.132   TCP       80→49392 [ACK] Seq=1 Ack=453 Win=15680 Len=0
2141 134.124.1.234     134.124.135.132   HTTP      HTTP/1.1 200 OK  (text/html)
2142 134.124.135.132   134.124.1.234     TCP       49392→80 [ACK] Seq=453 Ack=441 Win=65024 Len=0
2143 134.124.135.132   134.124.1.234     TCP       49392→80 [FIN, ACK] Seq=453 Ack=441 Win=65024 Len=0
2144 134.124.1.234     134.124.135.132   TCP       80→49392 [FIN, ACK] Seq=441 Ack=453 Win=15680 Len=0
2145 134.124.135.132   134.124.1.234     TCP       49392→80 [ACK] Seq=454 Ack=442 Win=65024 Len=0
2146 134.124.1.234     134.124.135.132   TCP       80→49392 [ACK] Seq=442 Ack=454 Win=15680 Len=0
```

```
⊞ Frame 2137: 506 bytes on wire (4048 bits), 506 bytes captured (4048 bits) on interface 0
⊞ Ethernet II, Src: IntelCor_b9:3a:f2 (0c:8b:fd:b9:3a:f2), Dst: Cisco_ff:fd:90 (00:08:e3:ff:fd:90)
⊞ Internet Protocol Version 4, Src: 134.124.135.132 (134.124.135.132), Dst: 134.124.1.234 (134.124.1.234)
⊟ Transmission Control Protocol, Src Port: 49392 (49392), Dst Port: 80 (80), Seq: 1, Ack: 1, Len: 452
    Source Port: 49392 (49392)
    Destination Port: 80 (80)
    [Stream index: 1]
    [TCP Segment Len: 452]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 453    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    Header Length: 20 bytes
  ⊞ .... 0000 0001 1000 = Flags: 0x018 (PSH, ACK)
    Window size value: 256
    [Calculated window size: 65536]
    [Window size scaling factor: 256]
  ⊞ Checksum: 0x3532 [validation disabled]
    Urgent pointer: 0
```

```
0000  00000000 00001000 11100011 11111111 11111101 10010000 00001100 10001011   ........
0008  11111101 10111001 00111010 11110010 00001000 00000000 01000101 00000000   ..:...E.
```

9) **With the above GET request packet selected (it should remain highlighted), please complete the rest of this lab**

LAB Question 2:

Using the packet associated with the GET request as mentioned above, please complete the following fields in the Frame Header below (SEE NEXT PAGE FOR FIGURE).

The idea here is to learn about the different fields and what they accomplish.

NOTE: Although the field values are always in binary (as seen in the bottom most section of your capture) please feel free to provide the values as they appear in the MIDDLE part (i.e. either decimal or hex as the case maybe) of Wireshark window.

Type in your values in the “Light Gray” shaded areas immediately below the each field. **Only fields with light gray areas below them are required to be completed**.

[Hint: please visit https://en.wikipedia.org/wiki/Ethernet_frame to learn more about Ethernet and its Frame Structure and to better understand what each of the fields below mean]

<table>
<tr><td rowspan="9">Layer 1 Ethernet packet (the actual bits sent at the phy</td><td colspan="3">Preamble - 7 Octets (Not displayed in Wireshark)</td></tr>
<tr><td colspan="3">Start of Frame Delimiter (SFD) - 1 Octet (Not displayed in Wireshark)</td></tr>
<tr><td rowspan="7">Layer 2 Ethernet Frame</td><td rowspan="4">Layer 2 Ethernet <b>Frame Header</b></td><td>Destination MAC Address - 6 Octets
<b><i>1c:1b:68:61:45:e0</i></b></td></tr>
<tr><td>Source MAC Address - 6 Octets
<b><i>34 : de : 1a :27 : ae : 0c</i></b></td></tr>
<tr><td>802.1Q tag - 4 Octets (<b>Optional</b>. Present only in 802.1Q tagged frames. Can have more than one
Tag Protocol ID (2 Octets)
Tag Control Information (2 Octets)
PCP (3 bits) | DEI (1 bit) | VLAN ID (12 bits)</td></tr>
<tr><td>Ethertype (present in Ethernet II frames) or length (present in ‘pure’ IEEE 802.3 frames) 2 O
<b><i>IPv4 (0x0800)</i></b></td></tr>
<tr><td rowspan="2"><b>Frame Data</b></td><td><b>Payload</b> (Either 42 octets or 46 octets minimum to 1500 octets maximum)
<b>Layer 3 Header</b>
(IP Datagram/Packet Header)
<b>Layer 4 Header</b>
(e.g., TCP Segment Header)
<b>Application Data</b>
(e.g., HTTP)</td></tr>
<tr><td>PADDING (if needed)</td></tr>
<tr><td><b>Frame Trailer</b></td><td>Frame Check Sequence (a 32-bit Cyclic Redundancy Check number, 4 Octets) (Not displayed in</td></tr>
</table>

sical layer)			
			Inter-packet Gap (12 Octets)

LAB Question 3: Using the **https://en.wikipedia.org/wiki/Ethernet_frame link or any other sources you find, please briefly explain what each of the following fields mean (i.e. their purpose). Please do not simply copy-paste but try to understand and explain. [Copy-paste answers will not receive any credit]**

[Hint: A good reading of the above link and understanding the frame structure will help with this task. The answers are given on the link above but try to actually understand what each field does]

1. Preamble - 7 Octets: 64 bit preamble is 7 bytes for synchronization and 1 byte for SFD Preamble syncs the frames. On physical layer.
2. Start of Frame Delimiter (SFD) - 1 Octet: Basically says that a new frame is coming basically. Physical Layer.
3. Destination MAC Address - 6 Octets: It identifies the Manufacturer in the first 3 bytes. Then once wireshark gives you a Destination MAC, that means there was a frame sync at this point. Then the second least significant bit is always set to 0 identifying that this MAC is factory. Otherwise it will be a 1.
4. Source MAC Address - 6 Octets: A MAC address is used to uniquely identify a node (or group of nodes) on an Ethernet or WiFi local network. They're used on Ethernet and WiFi networks and one is 'burned into' every Ethernet and WiFi interface by its manufacturer. The source MAC identifies the source node sending the packets.
5. Ethertype (present in Ethernet II frames) or length (present in 'pure' IEEE 802.3 frames) 2 Octets [explain both purposes clearly and clarify how is it known what purpose these 2 octets are serving]: EtherType: 2 Octets, it is used to indicate the size of the payload, and the type of protocol is being used in the payload frame. If there is a Ethertype number then it should use Ether II, if the higher layer protocol doesn't have an Ethertype number, then that upper layer protocol would have to use an 802.3/802.2 to designate the upper-layer to send the data to. They both designate the protocol encapsulated in the data field.
6. Frame Check Sequence - 4 Octets: (clearly identify what part of frame is this present in, what is the purpose, and how does it work) repeatedly looks for corrupt data throughout the entire frame on the receiving side. Corruption anywhere but the Frame Check Sequence. A healthy frame will always return a 0 for being healthy, using the CRC algorithm.

LAB Question 4: In general, what is the minimum size of the Layer 2 Ethernet Frame in bytes? What is/are the maximum size(s)? **MINIMUM IPv4 Packet size: Ethernet Header is 18 bytes + IPv4 Header is 20 bytes + UDP header 8 bytes + 18 byte payload = 64 bytes frame**

MINIMUM IPv6 Packet size: Ethernet Header is 18 bytes + IPv4 Header is 40 bytes + UDP header 8 bytes + 18 byte payload = 84 byte frame

MAXIMUM: Ether Maximum Transmission Unit (**MTU**) is 1500 bytes + header of 26 bytes = 1526 bit frame

LAB Question 5: Considering the *payload* of the layer 2 Ethernet frame:

1) What is the maximum *payload size* of a regular Ethernet frame?
   a. TCP Max JUMBO packet size is 9000 bytes subsequently could be 64k (65535 bytes)
   b. Ethernet IEEE 802.3 1518 bytes now 1522 bytes to allow VLAN tagging
   c. ICMP max is MTU 1500 bytes by default
2) Briefly but fully describe the contents of the payload.
   a. Payload does not include information such as metadata or headers. Payload is basically the cargo of the transmission. The payload performs the actions such as messages, texts, email, the actual message or point of the transmission is the payload.

LAB DELIVERABLES (to be uploaded to MyGateway):

1) **Name Word document as "FirstName_LastName_Assignment2". Upload completed Word document**
2) **Upload the Wireshark capture file**

GETTING HELP:

1) Call (314-489-9733) / email (shajikhan@umsl.edu) me anytime. Feel free to walk-in to my office if you see me there or setup appointment
2) Tutors will not be able to help you capture packets in Wireshark (capturing must be done at home). Tutors will be able to help with lab answers if you bring your captured file with you.
3) Of course try to help each other out. If some students are already familiar with the tasks listed above, I encourage you to help others. "Teaching" and helping others is by far the best way to learn!