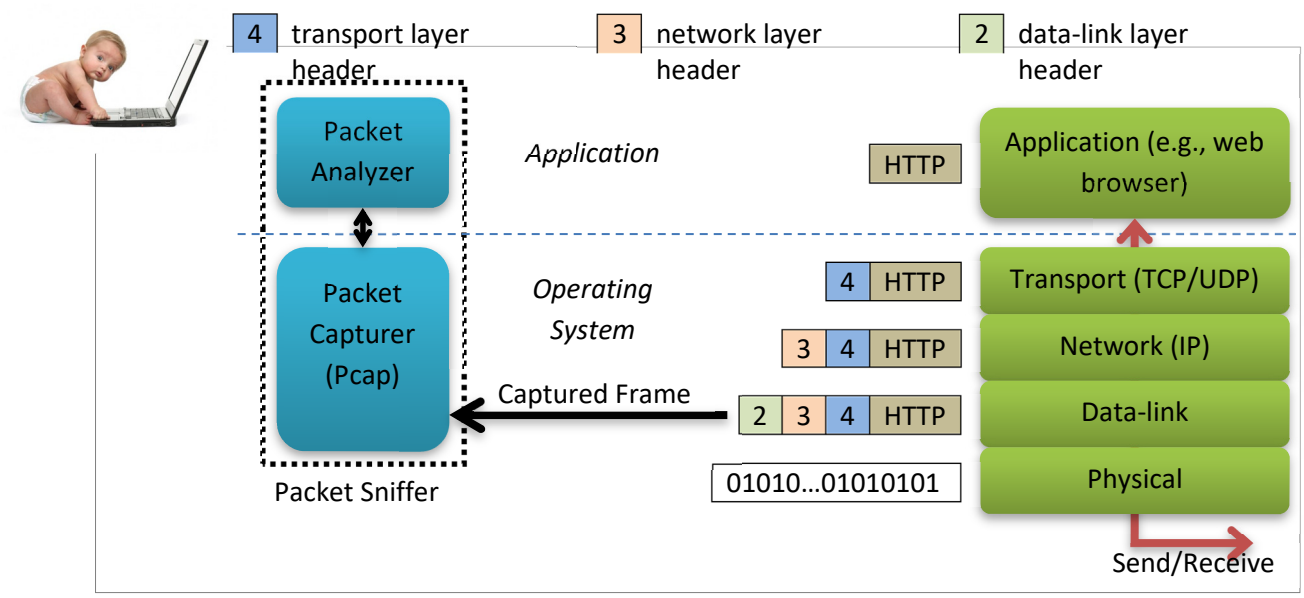
Submission Notes:

Please indicate your name and student ID and upload your solution to Canvas. No late submissions will be accepted.

Wireshark is a free *packet sniffer* available for major operating systems, such as Windows (64 bit/32 bit), MacOS, and Linux. A *packet sniffer* is a piece of software that captures packets being sent from and received by your computer. The captured packets allow you to understand the interaction and message exchanges of Internet protocols.

A packet sniffer software like Wireshark makes up of two components – a packet capturer and a packet analyzer. The packet capturer captures a copy of all outgoing and incoming frames (at the data-link layer) and passes them to the packet analyzer. The packet analyzer can then extract different headers and the ultimate message for analysis.

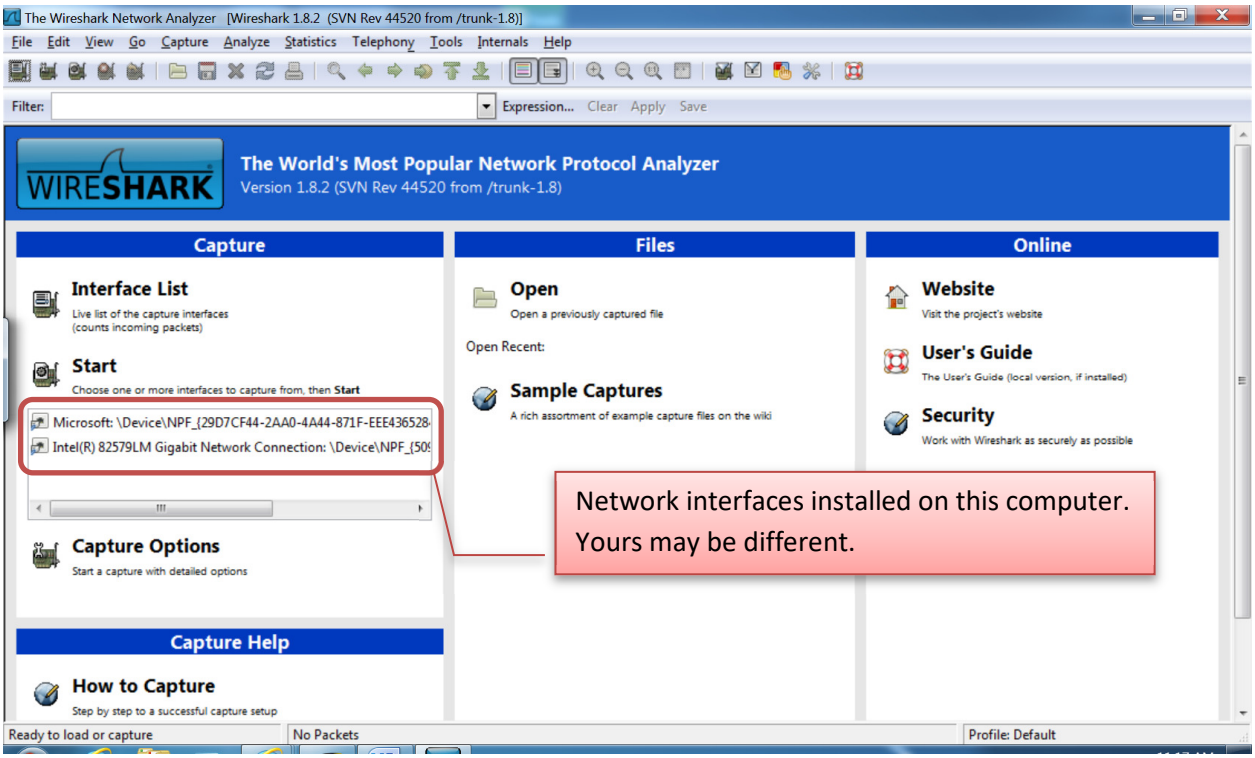


Getting Ready

Wireshark can be downloaded from http://www.wireshark.org/download.html. Detailed installation instructions can be found on http://www.wireshark.org/docs/wsug_html_chunked/ChapterBuildInstall.html. During install, Wireshark also installs a packet capturer Pcap(The capturer is Npcap in latest version of Wireshark). Some operating systems may require you to reboot your computer to use Wireshark.

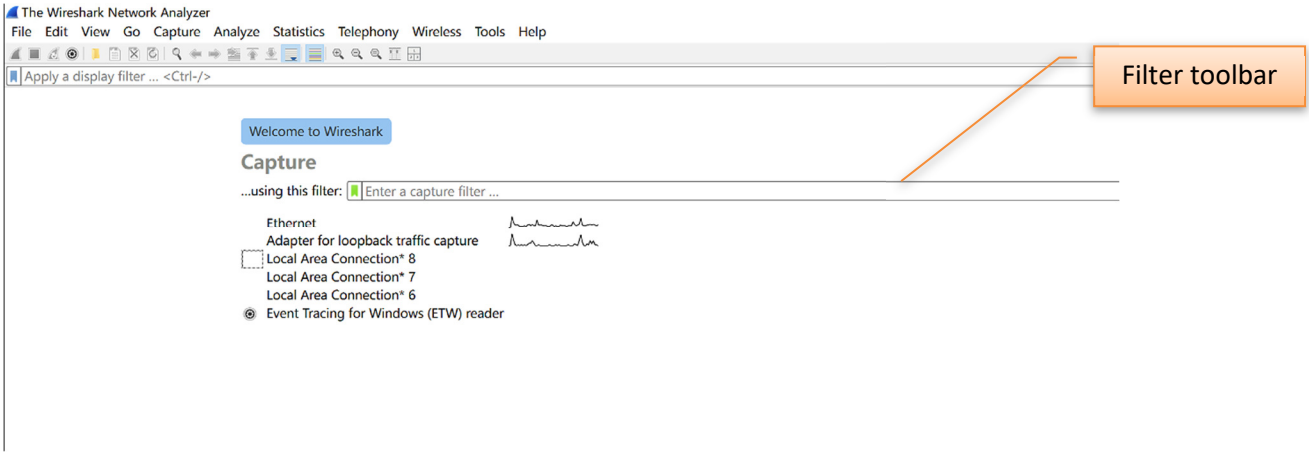
Familiar with Wireshark User Interface

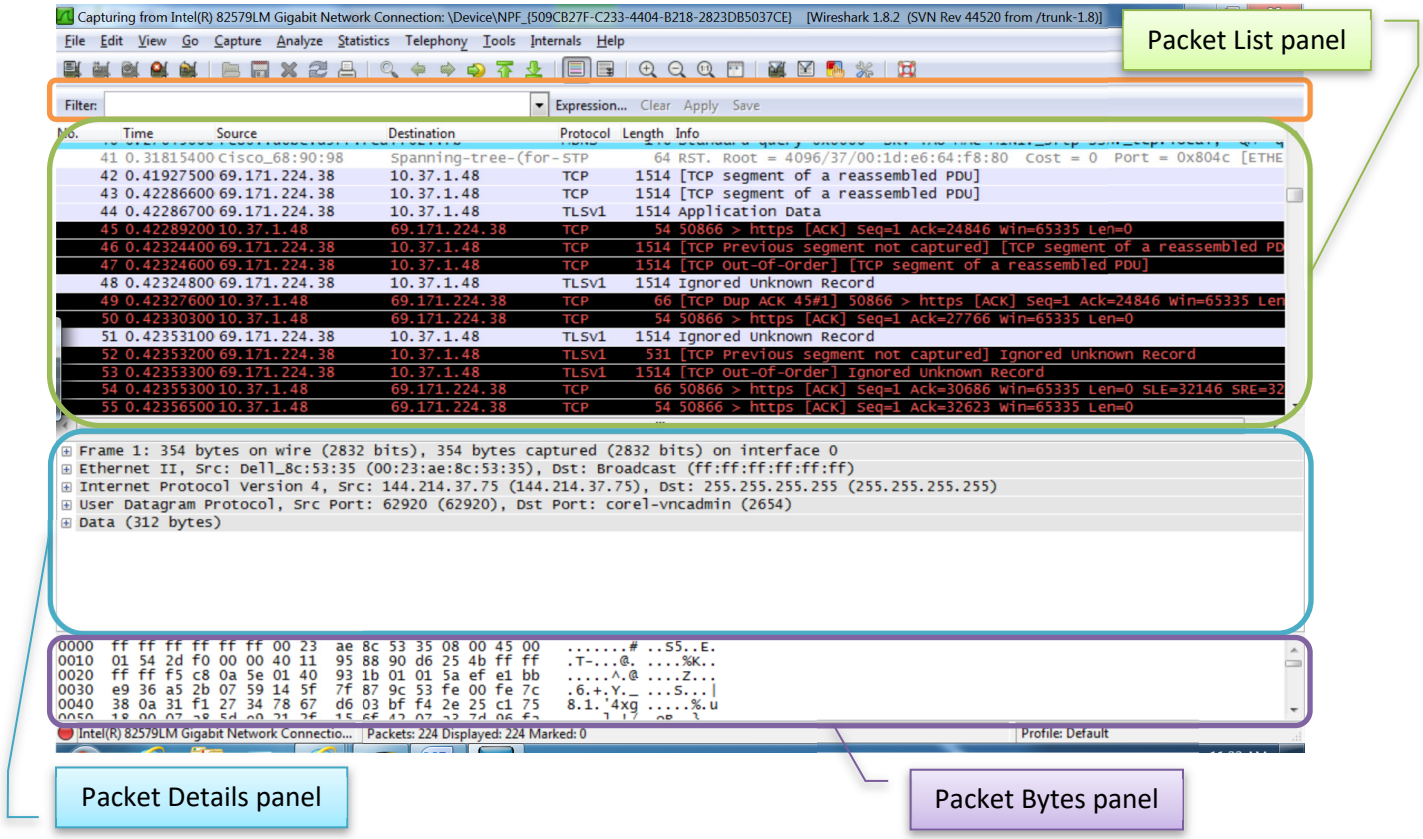
The startup screen of Wireshark 1.8.2 looks like the following.



The startup screen of Wireshark 4.4.0 looks like the following.

When Wireshark begins capturing packets, the screen looks like the following.

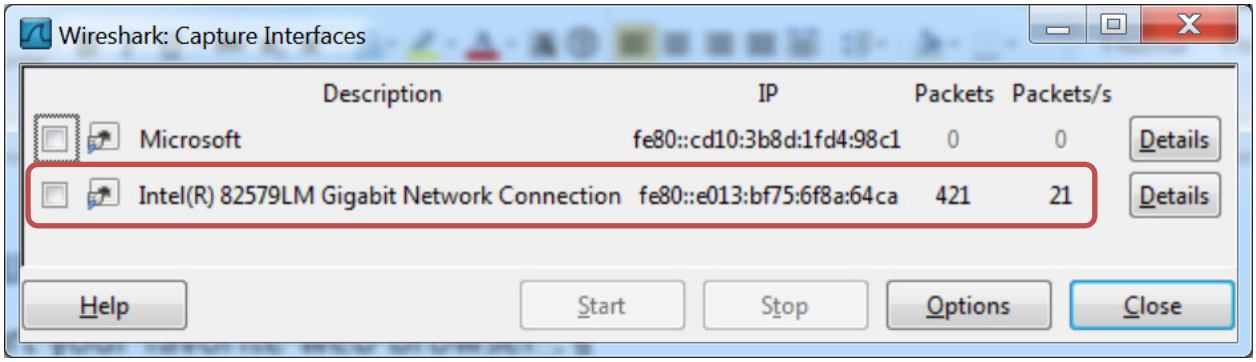




The detailed functionality of the filter toolbar and each panel can be found on http://www.wireshark.org/docs/wsug_html_chunked/ChapterUsing.html.

Problem 1: Capturing Web Browser HTTP Traffic (30 marks)

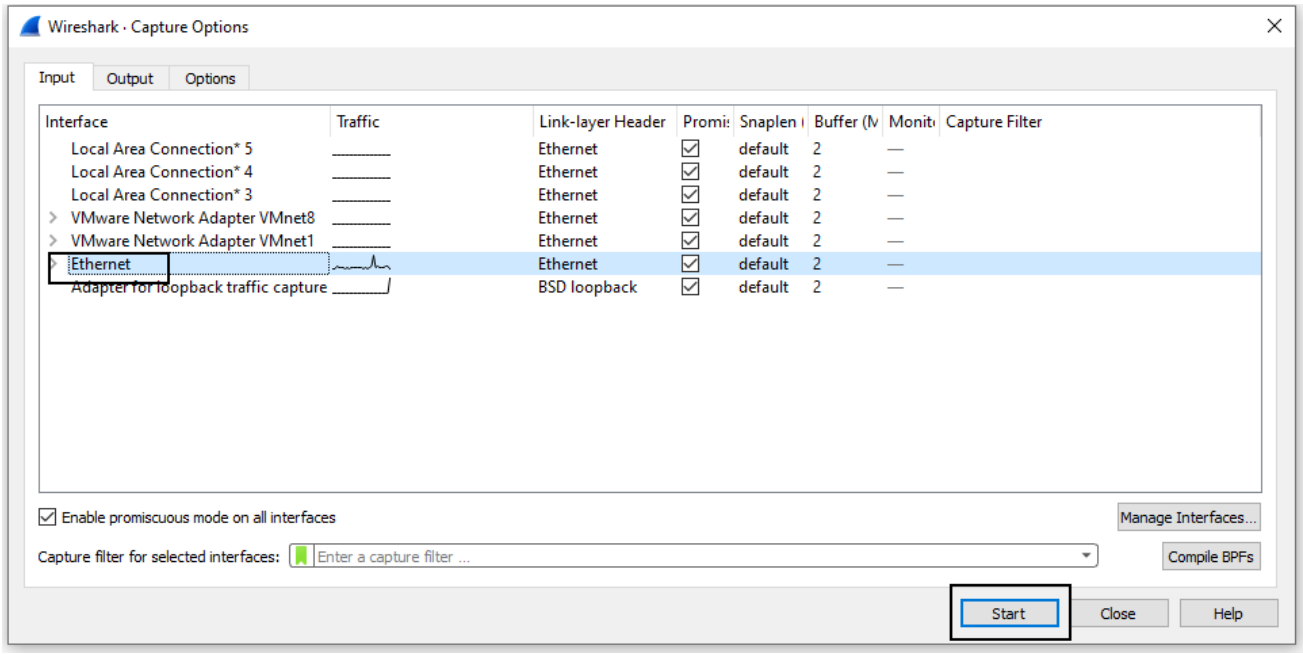
1. Start your favorite web browser[1].
2. (Wireshark 1.8.2) From the Wireshark menu, select Capture ➔ Interfaces…. A Wireshark: Capture Interfaces window appears.



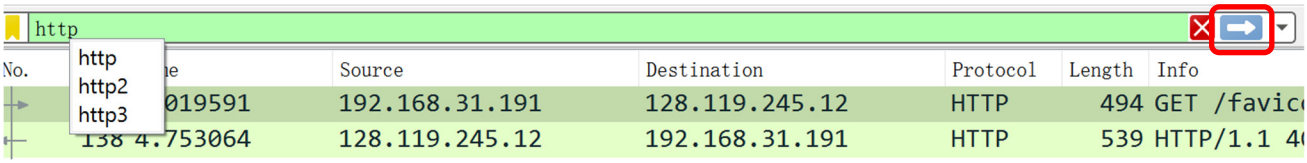
Select a network interface. If you have more than one network interfaces, select the one with non-zero number of packets. Then click the [Start] button to start capturing packets. A packet window likes the one shown in Familiar with Wireshark User Interface above appears.

[1] Clear your web browser's cache to ensure you are loading the fresh version of web page from the Internet. Different web browsers have different steps. Refer to http://www.wikihow.com/Clear-Your-Browser%27s-Cache if necessary.

(Wireshark 4.4.0) In the latest version, you can click the interface in the startup page. You can also select the interface by clicking Capture -> Options in the menu, select the capture interface and click 'start' (The name of your interface may be different from what is shown in the picture).

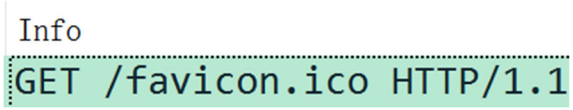


3. In your favorite web browser, open the following URL.
   http://gaia.cs.umass.edu/networks/resources/index.html
4. After your web browser displays the content of the above URL, stop Wireshark packet capture: from the Wireshark menu, select Capture ➔ Stop.
5. In Wireshark Filter toolbar, enter `http`. Then click the Rightarrow button or Enter on your keyboard.



Only HTTP messages are displayed in the Packet List panel.

6. In the Packet List panel, look for the HTTP GET message[2] from the Info column.



Select this message, the protocols and protocol fields of the packet selected are displayed in Packet Details panel.
In the Packet Details panel, click the > sign next to Transmission Control Protocol to show the details of the TCP packet.

Questions

Based on the above results, answer the following questions.

[2] Some web browser may use HTTP/1.0 instead of HTTP/1.1.

1. What languages (if any) does your browser indicate that it can accept to the server in HTTP Get message? (3 marks)
2. What application layer protocol does your web browser access above website? (3 marks)
3. What transport layer protocol does this HTTP GET packet use? (3 marks)
4. What is the destination IP address and port of the HTTP GET message? (3 marks)
5. For the HTTP GET message, what is the ***next sequence number*** in the TCP header? (3 marks)
6. Find the response packet corresponding to the HTTP GET message. Take a screenshot. Briefly describe how you are able to tell this packet corresponds to the HTTP GET message. (3 marks)

Do the following steps:

- Start up your web browser, and make sure your browser's cache is cleared.
- Start the Wireshark sniffer.
- Enter the following URL into your browser http://gaia.cs.umass.edu/networks/cnrg_wireless-10-02.pdf Your browser should display a one-page pdf file.
- Quickly enter the same URL into your browser again (or simply select the refresh button on your browser)
- Stop Wireshark packet capture, and enter "http" in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing window.

(Download problem 1.pcapng if you failed to capture the trace.)

Answer the following questions:

7. Inspect the contents of the first HTTP GET request from your browser to the server. Do you see an "IF-MODIFIED-SINCE" line in the HTTP GET? (1 mark)
8. Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell? (4 marks)
9. Now inspect the contents of the second HTTP GET request that requests the pdf file. Do you see an "IF-MODIFIED-SINCE:" line in the HTTP GET? If so, what information follows the "IF-MODIFIED-SINCE:" header? (3 marks)
10. What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain. (4 marks)

Problem 2: Capturing DNS Traffic (20 marks)

Background

We'll make extensive use of the *nslookup* tool, which is available in most Linux/Unix and Microsoft platforms today. To run *nslookup* in Linux/Unix, you just type the *nslookup* command on the command line. To run it in Windows, open the Command Prompt and run *nslookup* on the command line.

In it is most basic operation, *nslookup* tool allows the host running the tool to query any specified DNS server for a DNS record. The queried DNS server can be a root DNS server, a top-level-domain DNS server, an authoritative DNS server, or an intermediate DNS server. To accomplish this task, *nslookup* sends a DNS query to the specified DNS server, receives a DNS reply from that same DNS server, and displays the result.

Consider the first command:

```
nslookup www.mit.edu
```

In words, this command is saying "please send me the IP address for the host www.mit.edu". Try it yourself (You may use a different host name that you like). The response from this command provides two pieces of information: (1) the name and IP address of the DNS server that provides the answer; and (2) the answer itself, which is the host name and IP address of www.mit.edu. Although the response came from the default local DNS server, it is quite possible that this local DNS server iteratively contacted several other DNS servers to get the answer.

Now consider the second command:

```
nslookup –type=NS mit.edu
```

In this example, we have provided the option "-type=NS" and the domain "mit.edu". This causes *nslookup* to send a query for a type-NS record to the default local DNS server. In words, the query is saying, "please send me the host names of the authoritative DNS for mit.edu". (When the –type option is not used, *nslookup* uses the default, which is to query for type A records.) The answer first indicates the DNS server that is providing the answer (which is the default local DNS server) along with several MIT nameservers. Each of these servers is indeed an authoritative DNS server for the hosts on the MIT campus. However, *nslookup* also indicates that the answer is "non-authoritative," meaning that this answer came from the cache of some server rather than from an authoritative MIT DNS server. Finally, the answer also includes the IP addresses of the authoritative DNS servers at MIT. (Even though the type-NS query generated by *nslookup* did not explicitly ask for the IP addresses, the local DNS server returned these "for free" and *nslookup* displays the result.)

Now finally consider the third command:

```
nslookup www.google.com asia1.akam.net
```

In this example, we indicate that we want to the query sent to the DNS server asia1.akam.net rather than to the default DNS server. Thus, the query and reply transaction takes place directly between our querying host and asia1.akam.net. In this example, the DNS server asia1.akam.net provides the IP address of the host www.google.com, which is a web server at the Google Inc..

Do the following steps:

1. Open Windows Command Prompt[3]: type 'cmd' in the search bar.
2. From the Wireshark menu, select Capture ➔ Options. A Wireshark: Capture Interfaces window appears. Select a network interface then click the [Start] button to start capturing packets.
3. In Windows Command Prompt, enter the following command[4] to clear all cached DNS entries.

   `ipconfig /flushdns`

4. In Windows Command Prompt, enter the following command[5].

   `nslookup www.openrice.com`
   (`-timeout=60`: set the timeout to 60 seconds.)
   The output should contain the IP address of https://`www.openrice.com`

5. Stop Wireshark packet capture: from the Wireshark menu, select Capture ➔ Stop.
6. In Wireshark Filter toolbar, enter `dns`. Then click the 'Apply' symbol or press 'Enter' / 'Return' on your keyborad.

   Only DNS packets are displayed in the Packet List panel.

Questions

Based on the above results, answer the following questions.

1. Are they sent over UDP or TCP? (2 marks) How can you tell? (2 marks)
2. In the Packet List panel, look for the DNS query packet (Standard query *0x????* A `www.openrice.com`) from the Info column. What "Type" of DNS query is it? Does the query message contain any "answers"? (6 marks)
3. Look for the DNS response packet (Standard query response *0x????* A `www.openrice.com`). Examine the DNS response message. What is the source port of DNS response message? (2 mark) What is the type of address in the answer of the response message? (2 marks)
4. List the IP address of `www.openrice.com`. (3 marks)
5. To what IP address is the DNS query message sent? Use ipconfig to determine the IP address of your local DNS server (If you are using OSX, you may check it in "System preferences/ Network/ Advanced/ DNS"). Are these two IP addresses the same? (3 marks)

[3] If you use other operating systems, look up your operating system's user guide for steps on opening the console/terminal.
[4] If you use other operating systems, check out http://www.whatsmydns.net/flush-dns.html for steps on clearing all cached DNS entries on your operating system.
[5] If your operating system does not have `nslookup`, use `dig` from Internet Systems Consortium's BIND https://www.isc.org/downloads/current

Problem 3: Capturing SSL Traffic (35 marks)

The first step is to capture the packets in an SSL session. To do this, you should go to an e-commerce site and begin the process of purchasing an item (terminate before you actually pay!). After capturing the packets with Wireshark, you should set the filter "ssl" so that it displays only the Ethernet frames that contain SSL records sent from and received by your host.

Questions

1. Find a pair of client and server. Find the first 8 Ethernet frames in their session. For each frame, specify the source address and destination address, determine the number of SSL records that are included in the frame, and list the SSL record types that are included in the frame. Draw a table to show the packet exchange between client and server. An example of the table is shown as follows. (There might be segments missed by Wireshark, e.g., "**TCP Previous segment not captured, Ignored Unknown Record**". Find a session that has no missing segment.) (12 marks)

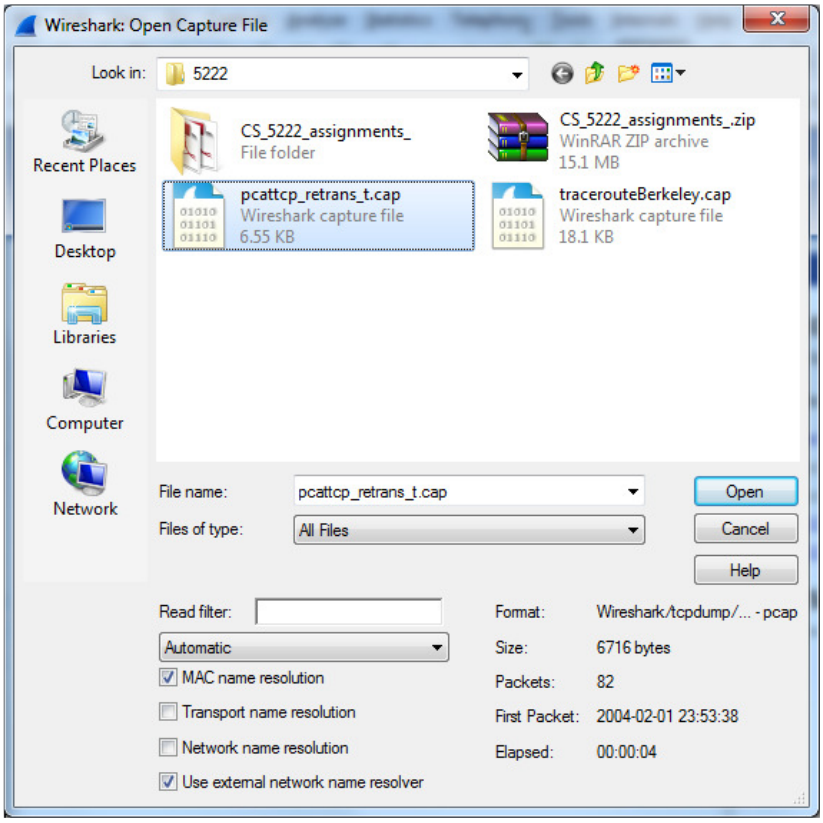
Frame	Source address	Destination address	SSL count	SSL Type

   Specify the IP addresses of client and server respectively. How can you tell? (3 marks)

2. Each of the SSL records begins with the same three fields (with possibly different values). One of these fields is "content type" and has length of one byte. List all three fields and their lengths. (3 marks)
3. Expand a ClientHello record. What is the value of the content type? What is it used for? (3 marks)
4. Does the ClientHello record advertise the cipher suites it supports? Show the first cipher suite. (2 mark)
5. Find the corresponding ServerHello SSL record. Does this record specify a chosen cipher suite? Show the chosen cipher suite. (3 mark)
6. Does the ServerHello SSL record include a nonce? If so, how long is it? What is the purpose of the client and server nonces in SSL? (3 mark)
7. Does the ServerHello SSL record include a session ID? What is the purpose of the session ID? (3 mark)
8. Does the ServerHello SSL record contain a certificate, or is the certificate included in a separate record. Does the certificate fit into a single Ethernet frame? (3 mark)

Problem 4: Analyzing TCP Traffic (15 marks)

Download `pcattcp_retrans_t.cap` Capture File from the course web. The Capture File contains Wireshark packet capture of TCP packets sending from 192.168.0.100:4480 to 192.168.0.102:5001. From the Wireshark menu, select File ➔ Open…. A Wireshark: Open Capture File window appears. Select `pcattcp_retrans_t.cap` you downloaded from step 1 then click the [Open] button to open the downloaded Capture File.



Wireshark displays the captured packets from `pcattcp_retrans_t.cap`.

Questions

1. Draw a diagram to illustrate the handshake between 192.168.0.100:4480 and 192.168.0.102:5001. List the length of the all packets used to complete TCP handshake. (6 marks)
2. Finding the first TCP segment sending from 192.168.0.100:4480 after the handshake of TCP. What's the length of this TCP segment? (3 marks)
3. Calculating the RTT value for the TCP segment you found in question 2. (Hint: the value of the Time column in the packet listing window is the amount of the time, in seconds, since Wireshark tracing began.) (6 marks)