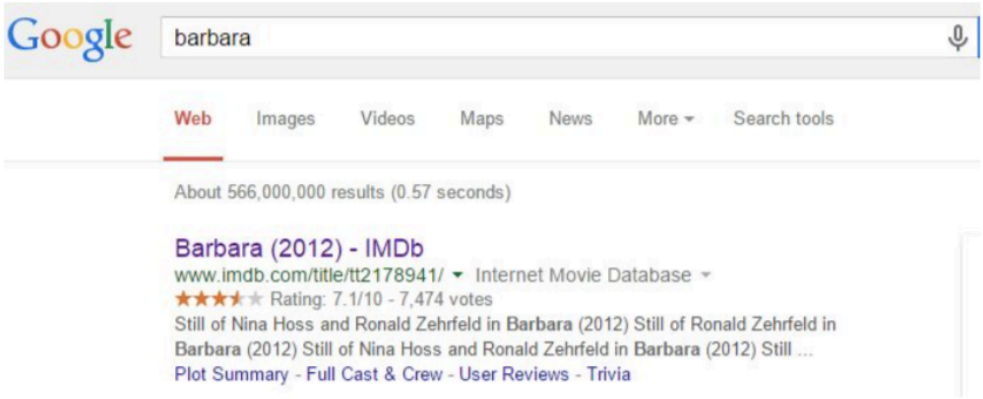
Lab 2

This lab consists of three parts. The first part is an exercise to learn more about DNS while the second and third parts help to learn about Wireshark.

Part 1: DNS

In this part we are going to get familiar with the DNS protocol and how we can use it. DNS is mainly used for getting the IP address of a host representing a domain name, but it also provides some other services. We will see some of its other services in this part. Without the DNS protocol, we would have to use IP addresses to access different websites which is hard to even imagine.

In this part, you are going to use a domain address as the target address we want to get information about. Each student has to use a domain name found by searching Google for your first name. For example if your name is "Barbara", you search Google for Barbara and the first returned result is your target domain name. In this example www.imdb.com is the target domain for a person named Barbara.



Now open a terminal and type the following command:

```
$ nslookup www.<your target>.com
```

```
amoha006@sierra-15 $ nslookup www.imdb.com
Server:         138.23.169.10
Address:        138.23.169.10#53

Non-authoritative answer:
www.imdb.com    canonical name = us.dd.imdb.com.
Name:   us.dd.imdb.com
Address: 72.21.203.211
```

Nslookup is a DNS resolver so it works by connecting to a DNS server to get information about the mapping between a domain name and its IP address. Normally, it is available in many different operating systems. Other tools like **host** and **dig** can be used for the same purpose too.

Based on the received response:
Q1.1) What are the server and address in first two lines?
Q1.2) What is the canonical name?
Q1.3) What is the IP address in the last line?

Now, by using the query flag, change the type of the query to MX and then to NS.

```
amoha006@sierra-15 $ nslookup -query=MX imdb.com
amoha006@sierra-15 $ nslookup -query=NS www.imdb.com
```

Q1.4) Can you guess what their functionality based on the output? If not, please try to find it by carrying out a search.

Now change the query type to SOA.
Q1.5) Be ready to explain what the following fields represent.

origin
mail
addr
serial
refresh
retry
expire

minimum

```
amoha006@sierra-15 $ nslookup -query=SOA www.imdb.com
Server:         138.23.169.10
Address:        138.23.169.10#53

Non-authoritative answer:
www.imdb.com    canonical name = us.dd.imdb.com.
us.dd.imdb.com
        origin = ns-924.amazon.com
        mail addr = root.amazon.com
        serial = 1428375037
        refresh = 3600
        retry = 900
        expire = 7776000
        minimum = 60
```

Q1.6) Is it possible to get the name for an IP by using the DNS protocol? If so, find the name representing 8.8.8.8 IP address. If not, explain why.

Demo: The TA may ask you to provide all or some of the answers to the questions above.

Part 2: Working with Wireshark

In this section, you will work with the wireshark interface and filter to get familiar with the tool.

1) Wireshark reads files with a .pcap extension so first we will copy some sample files from the lab computer to the mininet VM. We shall work with 1 pcap file dns.pcap. This file has been downloaded to /extra but can also be downloaded from https://wiki.wireshark.org/SampleCaptures (dns.cap is under 4. General/Unsorted). Note that .cap is the older extension used for .pcap files and Wireshark can open both types of files.)

2) Open a terminal on the lab workstation and move to folder /extra i.e. use command
   cd /extra/

3) Copy the pcap file that is stored under /extra/ to your folder under extra i.e.
   cp dns.pcap <directory name>
4) Get the workstation name by using
   uname –n

```
eliri001@wch129-04 $ uname -n
wch129-04.cs.ucr.edu
```

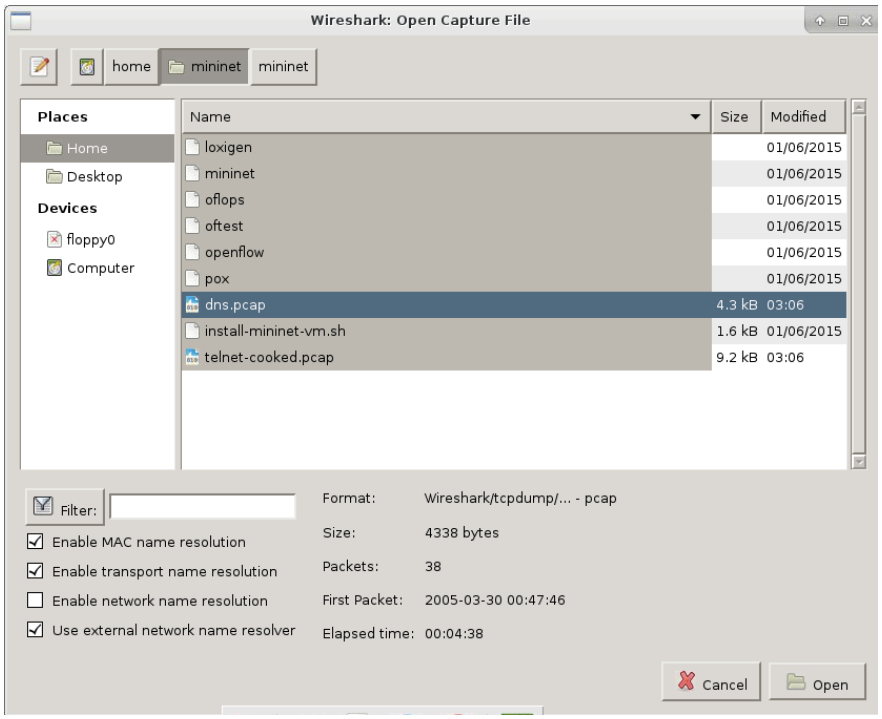
5) Go to the terminal that is connected to the mininet VM and make sure you are under /home/mininet then copy the file from the workstation to the mininet VM using scp

cd /home/mininet

scp –r <username>@<workstationname>.cs.ucr.edu: /extra/<directory name>/*.pcap ./

```
mininet@mininet-vm:~$
mininet@mininet-vm:~$ pwd
/home/mininet
mininet@mininet-vm:~$
mininet@mininet-vm:~$
mininet@mininet-vm:~$ scp -r eliri001@wch129-04.cs.ucr.edu:/extra/eliri001/*.pcap ./
eliri001@wch129-04.cs.ucr.edu's password:
dns.pcap                                      100% 4338     4.2KB/s   00:00
telnet-cooked.pcap                            100% 9244     9.0KB/s   00:00
mininet@mininet-vm:~$
```

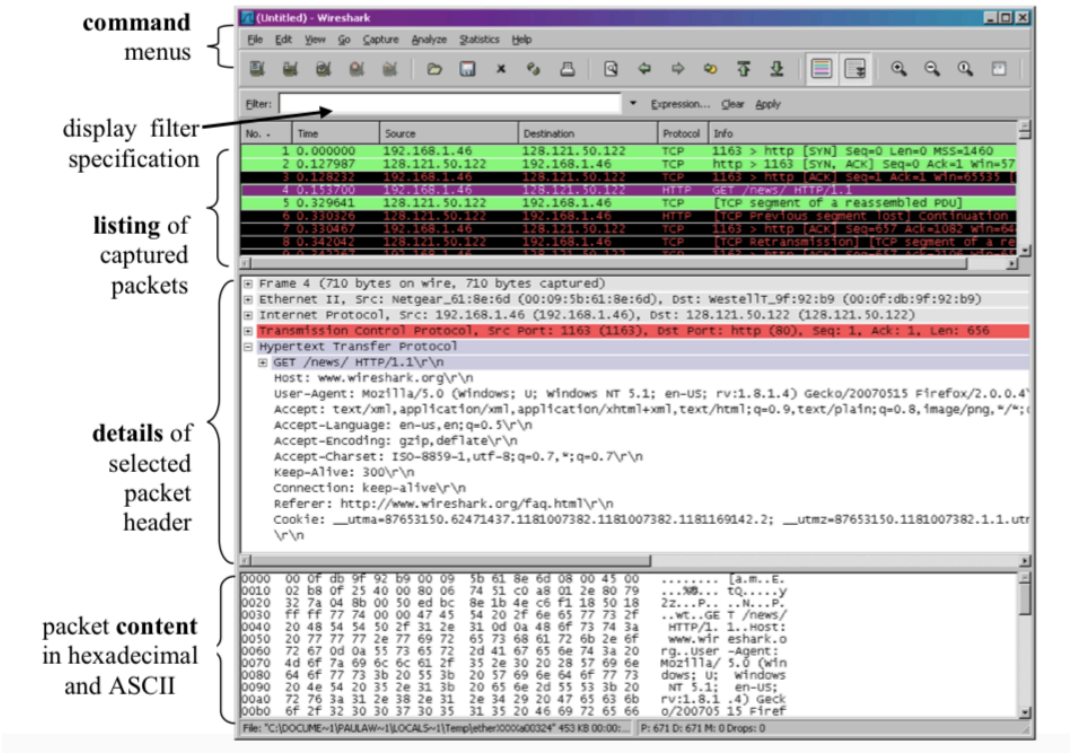
6) We can now open these files in Wireshark. In Wireshark go to File, Open. In the screen that opens, under Places select Home and then select dns.pcap under Name. Click Open to open the file.



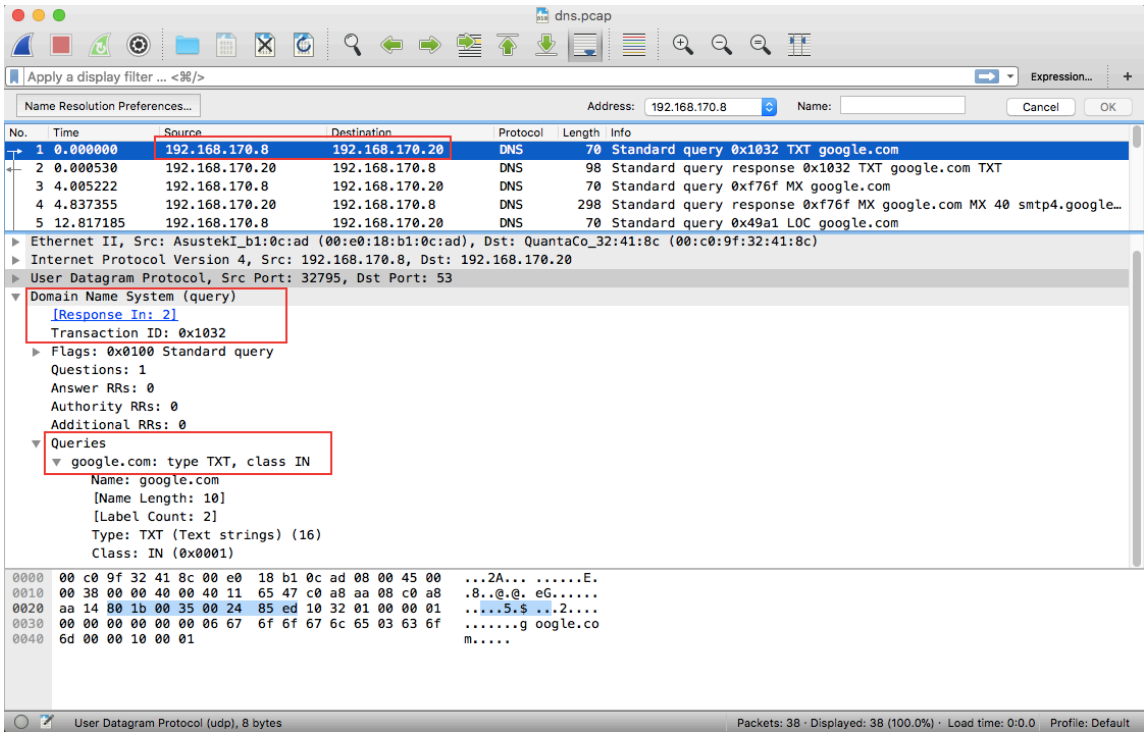
7) The next window that opens shows the file content and has different sections. The sections are labelled as shown in the next figure. (Note this figure is not showing the

contents of dns.pcap). Also note that below the packet content section, there is a small status description section indicating the file location, packets and the profile.
Also note that in the packet details section the data is separated into four layers: Link, Network, Transport and Application layers.

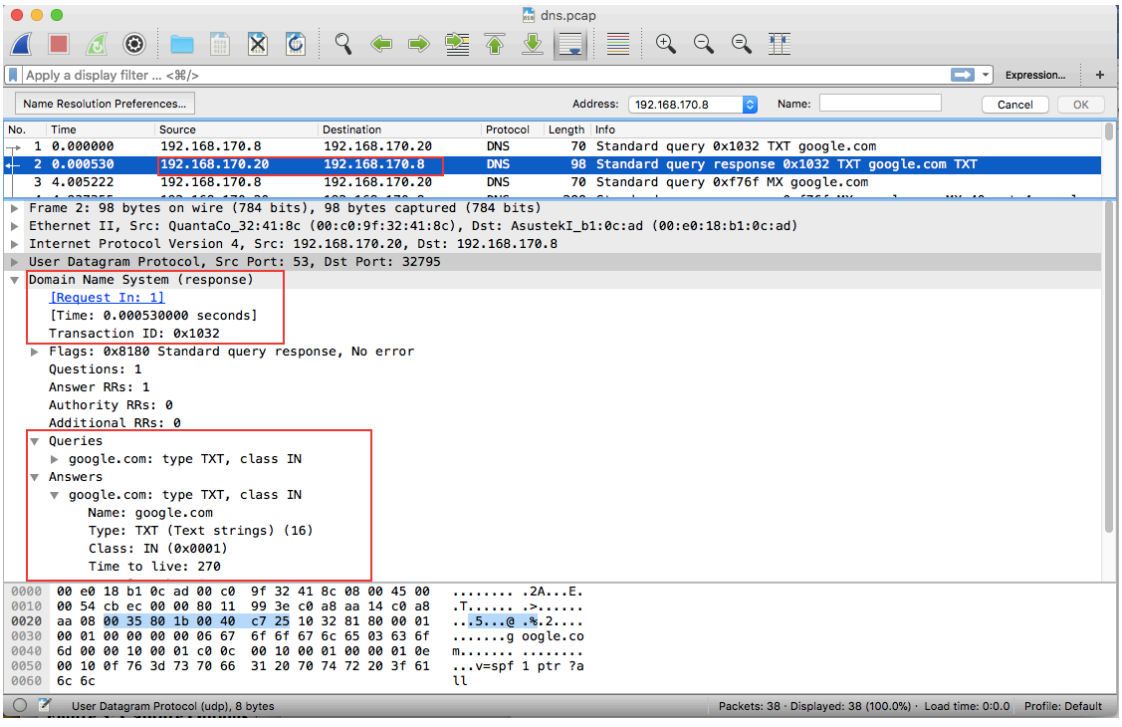
You can navigate through the different menu options in wireshark to get familiar with the different menu options. There are also many online tutorials available if you search via Google.



8) From the open dns.pcap file frame 1 shows a DNS request from 192.168.170.8 to 192.168.170.20. If you expand the Domain Name System field in the packet details section you can see that it tells you the Response is in frame 2. You can also see that the Transaction ID is 0x1032. The last part of the details section shows the payload which includes the website that the dns request is being sent out for **google.com** and the required parameters.



9) In frame 2, the DNS response, you can see the same Transaction ID and the payload includes both the query and the answer.

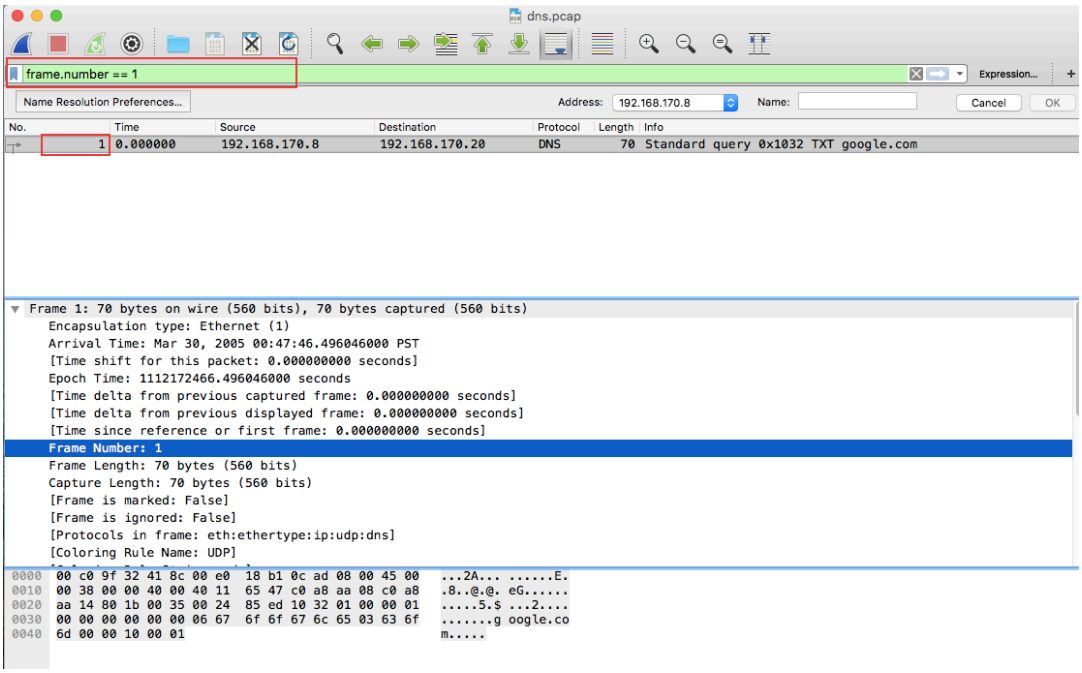


10) Another key feature in Wireshark is the use of filtering to identify packets of interest. In order to display particular packets, you need to enter the filter in the filter specification display box. There are three ways to do this which we present below.

11) Suppose we wanted to only display frame number 1 in the list.

a. Method 1: Manually enter filter

If you know the format of the filter parameters to use, you can manually enter this into the filter and click Apply. In this case we would enter **frame.number == 1** into the filter.

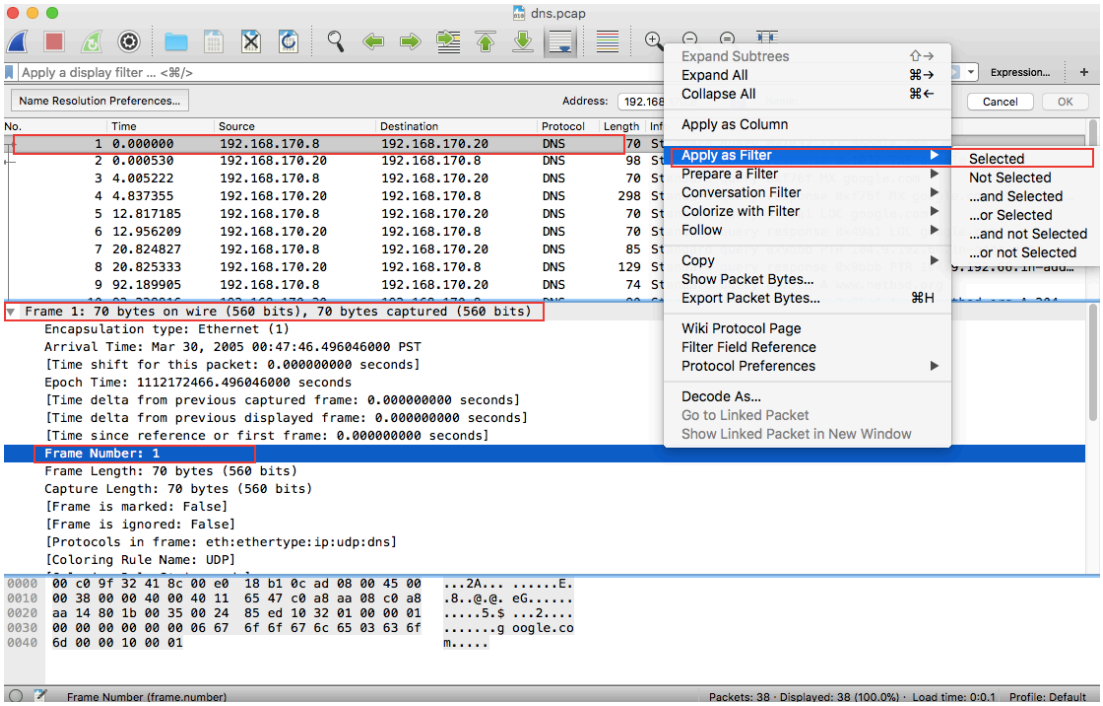


Clear the filter by deleting the text in the display filter and pressing Enter or Apply.

b. Method 2: Use Apply as Filter menu

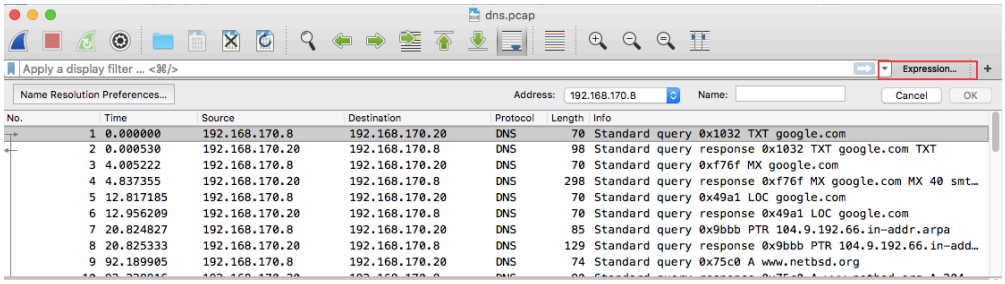
In case you are not sure about the format, you first select the packet you are interested in from the packet list i.e. select frame 1.

- In the packet details window, right click the parameter you are interested in. Note that you may need to expand some fields. In this case we expand the Frame section and right click Frame Number:1 . This presents several menu options and you select Apply as Filter, Selected. This then uses the field Frame Number: 1 (highlighted in blue) as the filter value.

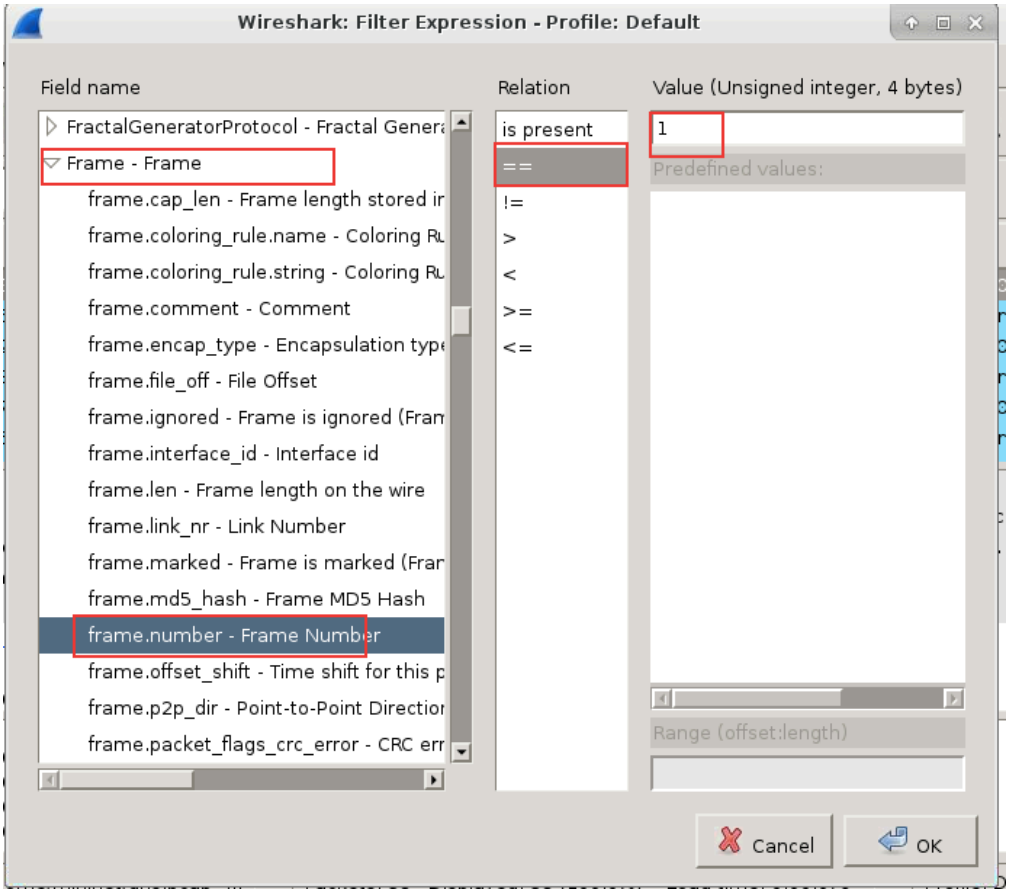


c. Method 3: Use the Expression menu

In this option, you navigate through the wireshark display expressions. In the main Wireshark window click expressions



- The next window that opens shows the list of protocols Wireshark supports together with their various parameters and any predefined values.
- Scroll through the Field Name to Frame and expand the Frame entry by clicking the down arrow.
- Navigate down to and select Frame Number.
- Select the appropriate relation which in this case is ==
- Enter the Value of 1 and since this parameter has no predefined values you can click ok.
- Back in the main window click Apply.



Note that

- Delete the filter by clicking clear
- If the filter parameters are all correct the display filter background will be green

12) Using what you have learned about filtering and by performing a search where necessary, please answer the following questions using the dns.pcap file:

d. Frame 3 is another dns request. How many mail exchange servers were sent back from 192.168.170.20 in response to this request? Give the mail server names and their IP addresses.

e. What are the frame numbers for the packets with the following Transaction IDs

i. 0xdca2
ii. 0x208a
iii. 0xd060

f. How many packets have Transaction IDs between 0x7777 and 0x8888? What are their frame numbers?

g. Using method 2 to get the filter parameters and method 3 to view the predefined values, what is last predefined value stored for the dns query type? You can give the string or integer value.

h. How long was this trace run for and on what day was this trace run? (Hint: You can use View, Time Display Format to change the displayed time)

You can close the file using File, Close but leave Wireshark running for the next section

Demo: The TA may ask to see all or some of the answers to the questions above.

Part 3: WGET and Wireshark Activity

For this part, you do not have to store the output your terminal commands in a file, but you do have to keep your wireshark window open.

As in the previous lab, we will be using wireshark to investigate the activity of a w get request. If you don't remember how to do this, please refer to the Part 1 instructions.

Open wireshark and start capturing packets. In the mininet terminal send a simple wget request to www.cnn.com.

$ wget www.cnn.com

In Wireshark use an appropriate filter to focus on packets between this wget session and the cnn website. You may want to use a filter similar to the following, which configures Wireshark to not show the packets from IP address 10.0.2.2.

!ip.addr == 10.0.2.2

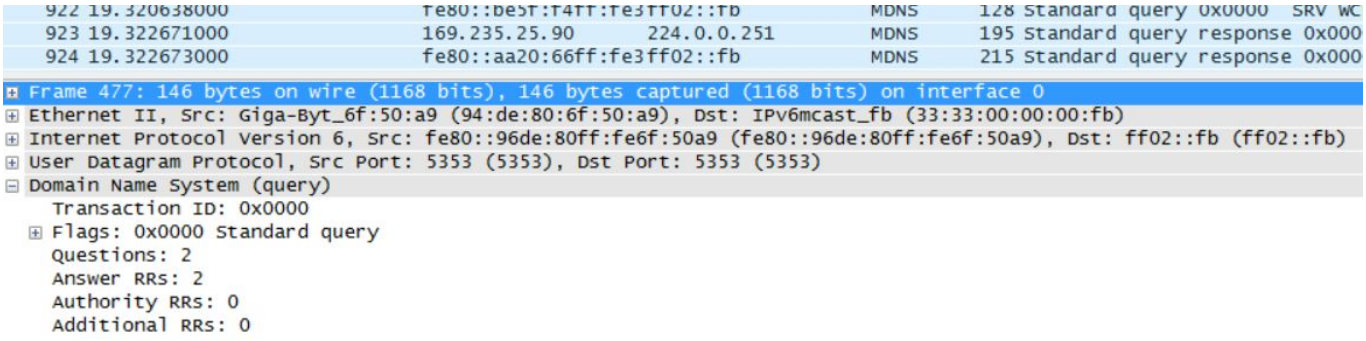
Q3.1) What are the first packets? (Normally Wireshark shows them in a different color.)

Q3.1.1) Are these packets over UDP or TCP? What is the destination port number?

Q3.1.2) What is their destination IP address?

Q3.1.3) How does your VM know this IP address?

Q3.1.4) What is the query? (By selecting a packet, you can find detailed information about its other layers like in the following image.)



Q3.1.5) What is the response?

Q3.1.6) Why are there two sets of queries and responses? What is the difference?

Q3.2) After the first query and response part, what are the next three packets?

Q3.3) After these three packets, what type of request is sent?

Q3.4) In the next packets, how are the ack numbers related to the sequence number of the packets before and the packets after? (If you're not familiar with the way TCP sequence/acknowledgement numbers work, check out http://packetlife.net/blog/2010/jun/7/understanding-tcp-sequence-acknowledgment-numbers/

Q3.5) Explain what the last 4 packets are and what their purpose is.

Demo: Please keep the wireshark open and be ready to answer the aforementioned questions. Good Luck!