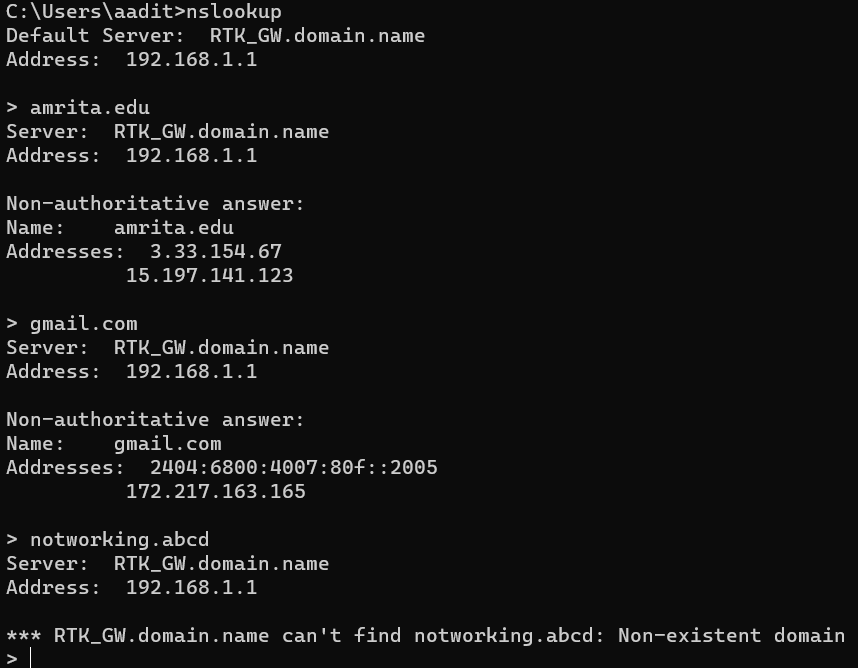
**1. nslookup**

In this lab, 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 (see the textbook for definitions of these terms).
* 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.



**2. ipconfig**

ipconfig (for Windows) and ifconfig (for Linux/Unix) are among the most useful little utilities in your host, especially for debugging network issues.

Here we’ll only describe ipconfig, although the Linux/Unix ifconfig is very similar.

**ipconfig** can be used to show your current TCP/IP information, including your address, DNS server addresses, adapter type and so on. For example, if you want to see all this information about your host, simply enter:

**ipconfig /all**

into the Command Prompt, as shown in the following screenshot.

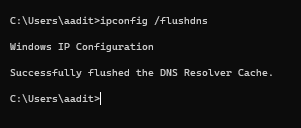


* *ipconfig* is also very useful for managing the DNS information stored in your host.
* We have learned that a host can cache DNS records it recently obtained.
* To see these cached records, after the prompt provide the following command: **ipconfig /displaydns**



Each entry shows the remaining Time to Live (TTL) in seconds. To clear the cache, enter

**ipconfig /flushdns**

****

Flushing the DNS cache clears all entries and reloads the entries from the hosts file.

**3. Tracing DNS with Wireshark**

Now that we are familiar with nslookup and ipconfig, we’re ready to get down to some serious business. Let’s first capture the DNS packets that are generated by ordinary Web surfing activity.

• Use ipconfig to empty the DNS cache in your host.

• Open your browser and empty your browser cache. (With Internet Explorer, go to Tools menu and select Internet Options; then in the General tab select Delete Files.)

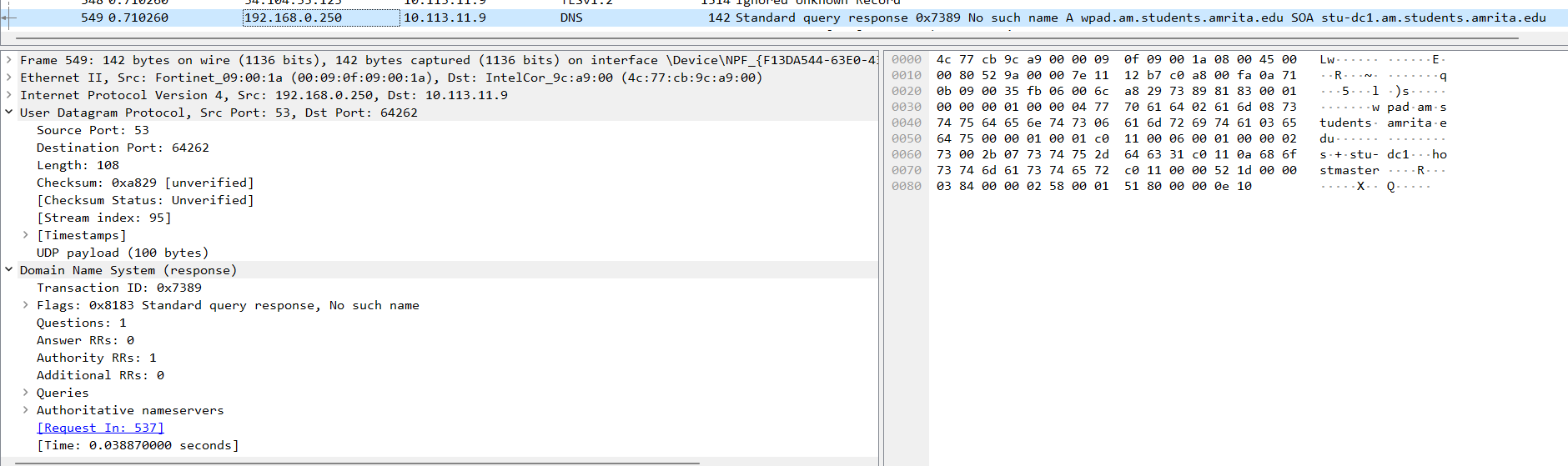
• Open Wireshark and enter “ip.addr == your\_IP\_address” into the filter, where you obtain your\_IP\_address (the IP address for the computer on which you are running Wireshark) with ipconfig. This filter removes all packets that neither originate nor are destined to your host.

• With your browser, visit Web pages in internet.

• Stop packet capture.

1. Explain the working of the DNS protocol[DNS Request Query and DNS Response message] briefly with typed answers and answer highlighted screenshots for the above capture
   1. Locate the DNS query and response messages. Are they sent over UDP or TCP?

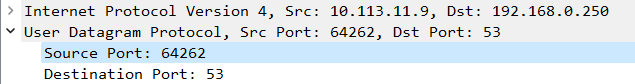
Ans: They are sent over UDP



* 1. What is the destination port for the DNS query message? What is the source port of DNS response message?

Ans: Source Port: 53, Destination Port: 64262

Query Message:

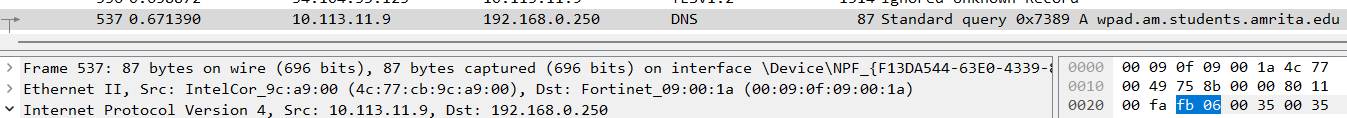


Response Message:

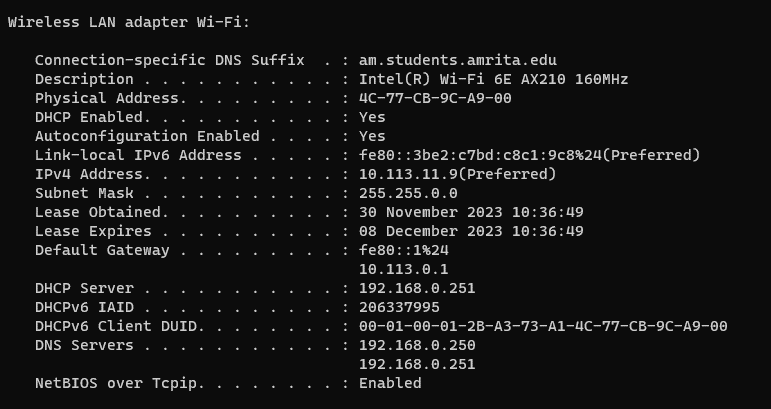


* 1. To what IP address is the DNS query message sent? Use ipconfig to determine the IP address of your local DNS server. Are these two IP addresses the same?

Ans: DNS Server : 192.168.0.250

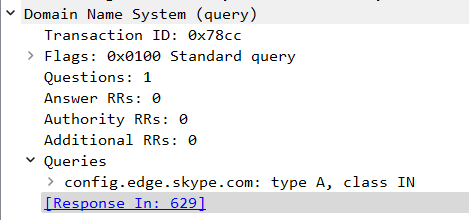


Yes, the DNS query is being to sent to the same IP address



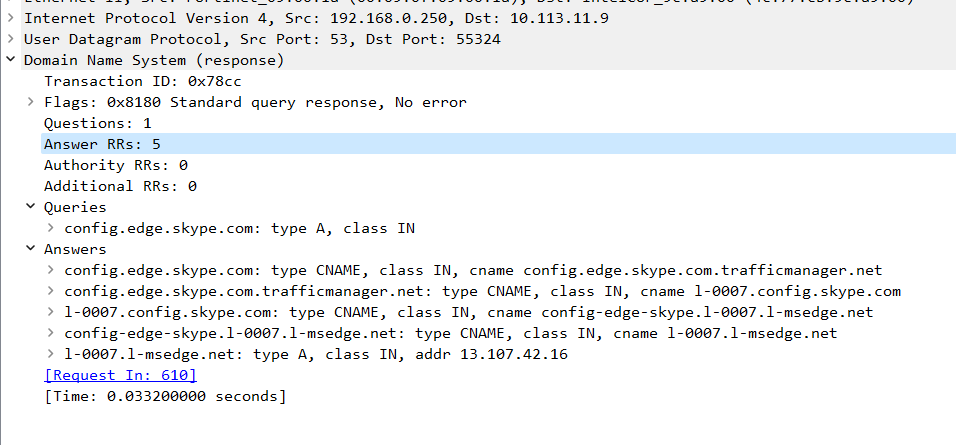
* 1. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?

Ans: The DNS query is a Standard Query. It does not contain any answer



* 1. Examine the DNS response message. How many “answers” are provided? What does each of these answers contain?

Ans: There are 5 answers in the response message



* 1. Before retrieving each image/object in your web page, does your host issue new DNS queries?

Ans: No, the host doesn’t issue new DNS queries for each image or object

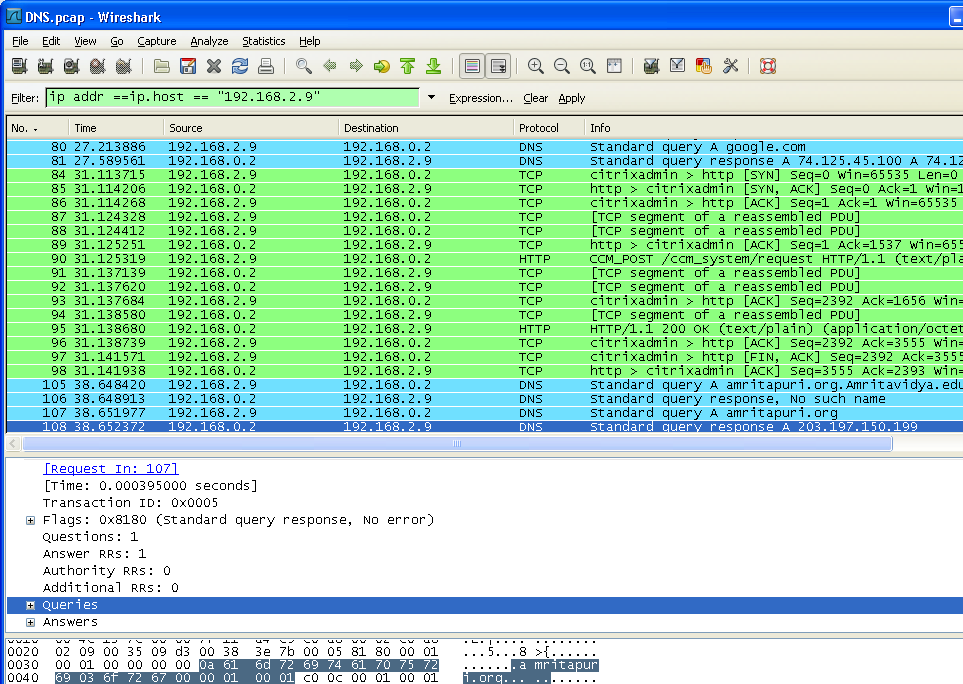
1. Now let’s play with nslookup.

• Start packet capture.

• Do an nslookup on [amritapuri.org](http://amritapuri.org), [google.com](http://www.google.com) etc.

• Stop packet capture.

You should get a trace that looks something like the following:



We see from the above screenshot that *nslookup* actually sent two/three DNS queries and received two/three DNS responses.

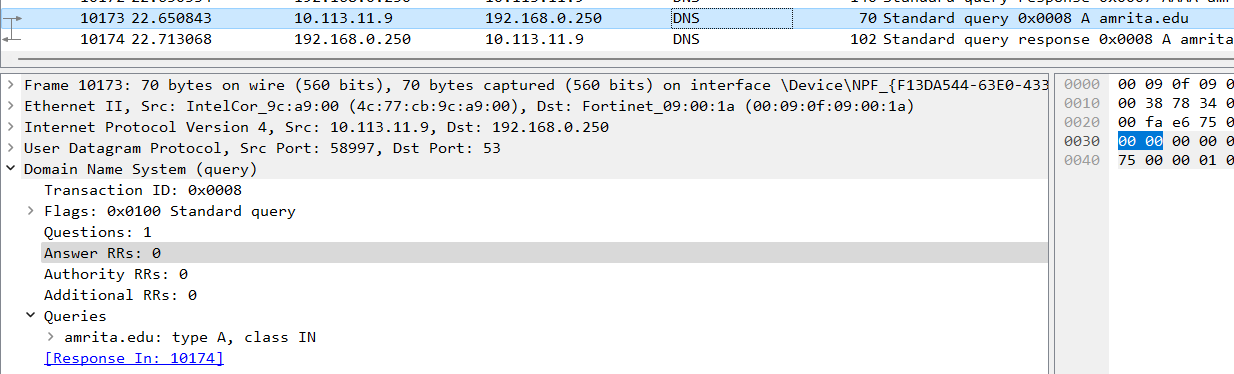
For the purpose of this assignment, in answering the following questions ignore the first one/two sets of queries/responses, as they are specific to *nslookup* and are not normally generated by standard Internet applications.

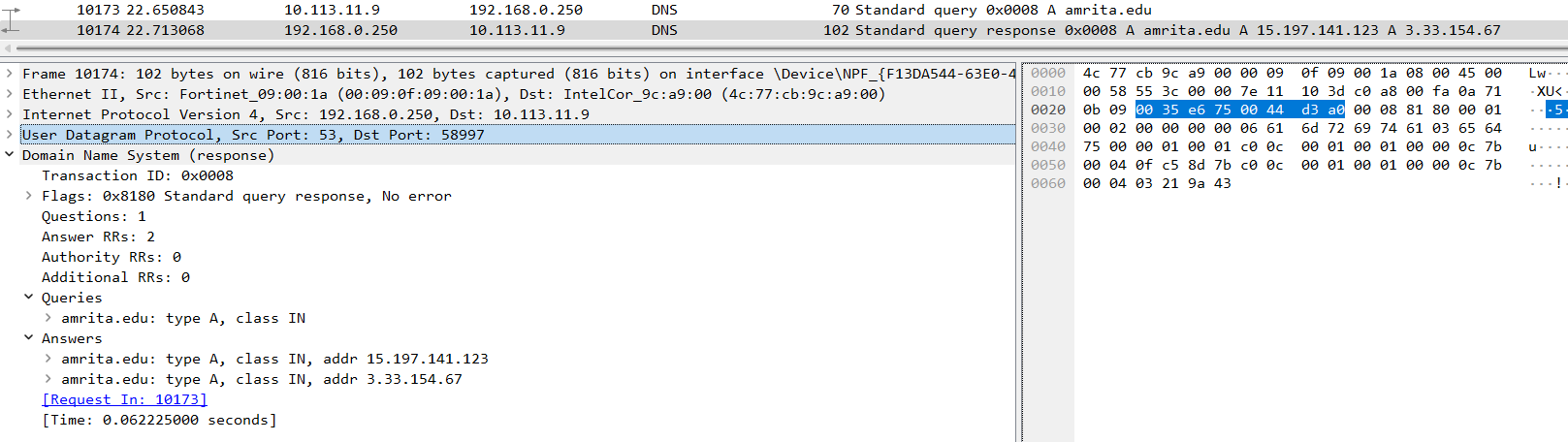
You should instead focus on the last query and response messages. Again, answer the following questions for this capture of frames.

1. What is the destination port for the DNS query message? What is the source port of DNS response message?

Ans: Destination Port of query message = 53

Source Port of DNS response message = 53



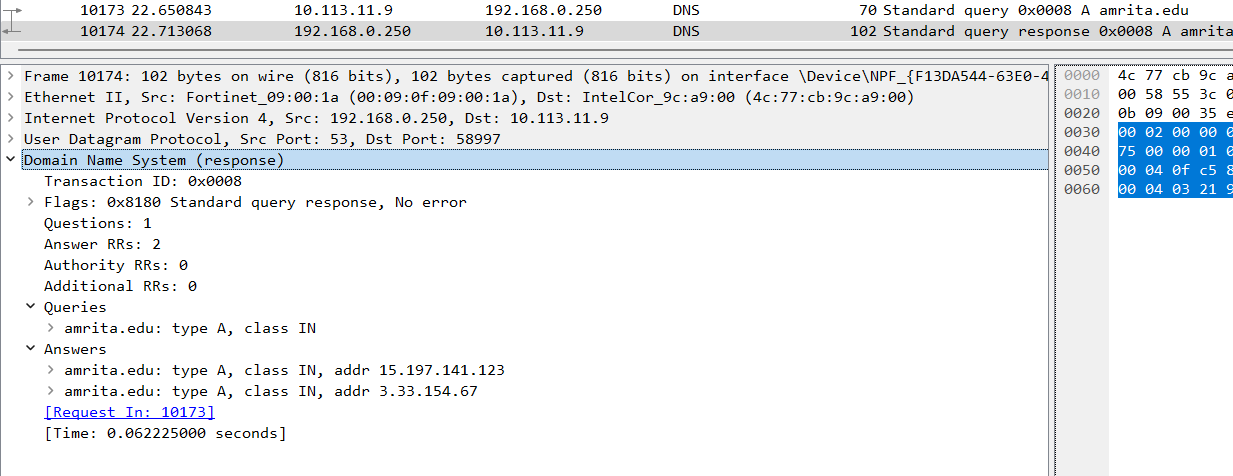


1. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?

Ans: IP address = 192.168.0.250, Yes its the same address of my default local DNS server

1. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?

Ans: It’s a standard query response (Type A), Yes it has 2 answers



1. Examine the DNS response message. How many “answers” are provided? What does each of these answers contain?

Ans: The DNS response message contains 2 answers

