Wireshark Lab: DNS

Submit online as a pdf in Canvas

**General Instructions:** **What to hand in: Please answer the questions posed in this lab, please make it clear what questions you are answering, and please use screenshots to support your answers. Marks will be awarded for correctness, completeness, and professionalism. You’ll also be using this document to study for your tests and exams.**

Whenever possible, when answering a question below, create a screenshot of the packet(s) within the trace that you used to answer the question asked. Annotate the printout to explain your answer. To print a packet, use *File->Print*, choose *selected packet only*, choose *Packet summary line,* and select the minimum amount of packet detail that you need to answer the question.

**Learning Outcome:** At the end of this lab you should

* Be familiar with the different DNS message types.
* Understand the role of the DNS server in any network.
* Finally, be able to explain the operation of DNS.

## Exercise 1

**Objective for Exercise 1:**

Understand the role of nslookup tool. The *nslookup* tool allows the host running the tool to query any specified DNS server for a DNS record.



The above screenshot shows the results of three independent *nslookup* commands (displayed in the Windows Command Prompt). In this example, the client host is located on the campus of University of Auckland, where the default local DNS server is dns-virtual1.auckland.ac.nz. When running *nslookup*, if no DNS server is specified, then *nslookup* sends the query to the default DNS server, which in this case is dns-virtual1.auckland.ac.nz. Consider the first command:

nslookup www.auckland.ac.nz

In words, this command is saying “please send me the IP address for the host www.auckland.ac.nz”. As shown in the screenshot, 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.auckland.ac.nz. Although the response came from the local DNS server at Auckland University, 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 auckland.ac.nz

In this example, we have provided the option “-type=NS” and the domain “auckland.ac.nz”. 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 auckland.ac.nz”. (When the –type option is not used, *nslookup* uses the default, which is to query for type A records.) The answer, displayed in the above screenshot, first indicates the DNS server that is providing the answer (which is the default local DNS server) along with two UoA name servers. Each of these servers is indeed an authoritative DNS server for the hosts on the UoA campus. If *nslookup* indicates that the answer is “non-authoritative,” it means that this answer came from the cache of some server rather than from an authoritative UoA DNS server.

Now finally consider the third command:

nslookup www.assta.org dns1.auckland.ac.nz.

In this example, we indicate that we want to the query sent to the DNS server dns1.auckland.ac.nz rather than to the default DNS server (dns-virtual1.auckland.ac.nz). Thus, the query and reply transaction takes place directly between our querying host and dns1.auckland.ac.nz. In this example, the DNS server dns1.auckland.ac.nz provides the IP address of the host www.assta.org, which is a web server at the Australasian Speech Science and Technology Association.

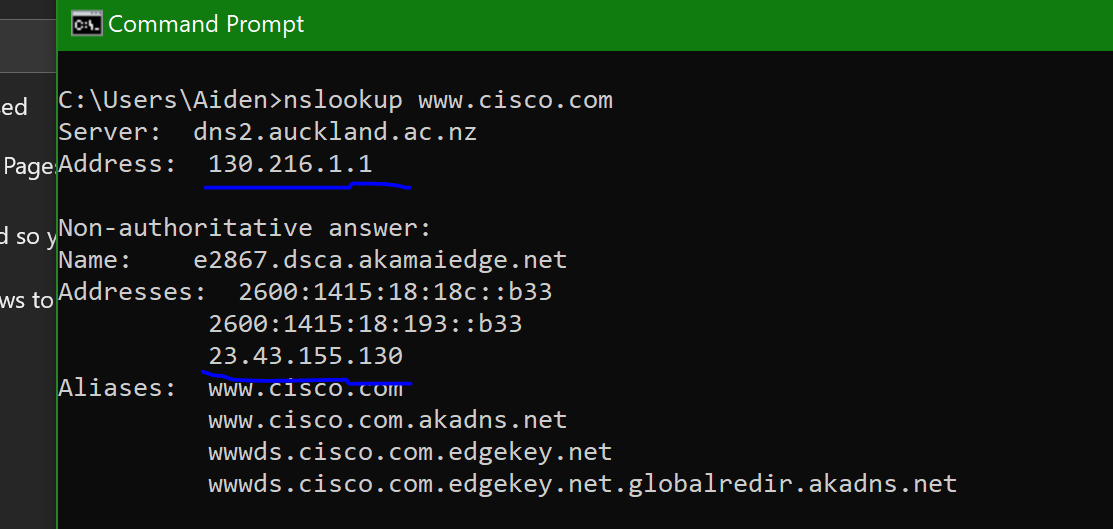
Now that we have gone through a few illustrative examples, you are perhaps wondering about the general syntax of *nslookup* commands. The syntax is:

nslookup –option1 –option2 host-to-find dns-server

In general, *nslookup* can be run with zero, one, two or more options. And as we have seen in the above examples, the dns-server is optional as well; if it is not supplied, the query is sent to the default local DNS server.

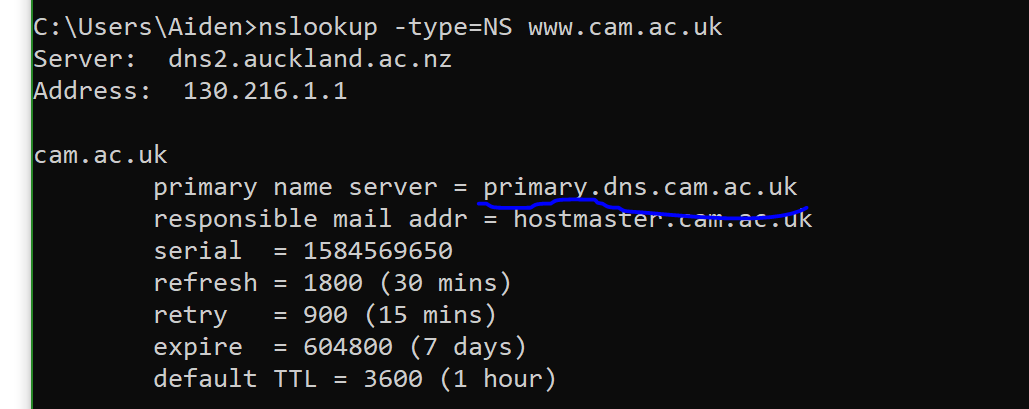
## Activity and Questions

Now that we have provided an overview of *nslookup*, it is time for you to test drive it yourself. Do the following (and write down the results):

* 1. Run *nslookup* to obtain the IP address of “www.cisco.com”. What is the IP address of the DNS server providing the answer and what is the IP address of www.cisco.com? (just the IPv4 address)

The IP address of the DNS server is 130.216.1.1

The IP address of [www.cisco.com](http://www.cisco.com) is 23.45.155.130

* 1. Run *nslookup* to determine the authoritative DNS servers for University of Cambridge in the UK. What is the address of this server?

The authorative DNS servers for University of Cambridge is primary.dns.cam.ac.uk

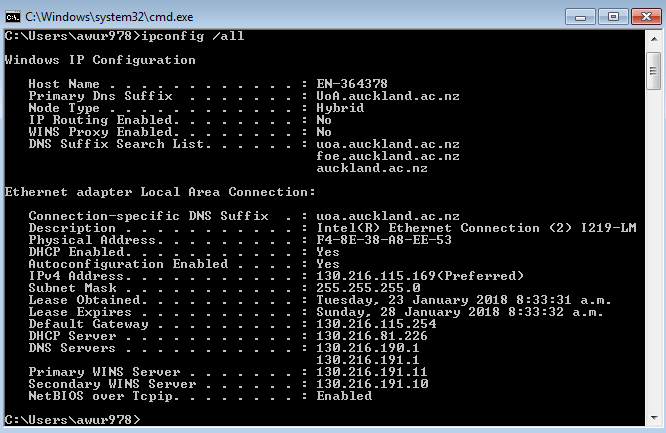
## Exercise 2

# **Objective for Exercise 2: Understanding the role of 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 enter all this information about your host simply by entering

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. Both your web browser and your operating system can cache DNS records. To see these cached records on your host computer, after the prompt C:\> 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.

## Exercise 3

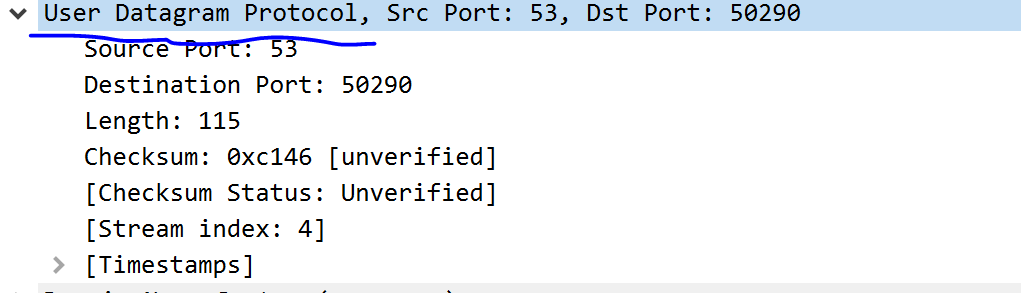
# **Objective for Exercise 3: Tracing DNS with Wireshark**

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 with ipconfig. This filter removes all packets that neither originate nor are destined to your host.
* Start packet capture in Wireshark.
* With your browser, visit the Web page: [http://www.ietf.org](http://www.ietf.org/)
* Stop packet capture.

## Activities and Questions

Answer the following questions. Whenever possible, when answering a question below, you should hand in a printout of the packet(s) within the trace that you used to answer the question asked. Annotate the printout to explain your answer.

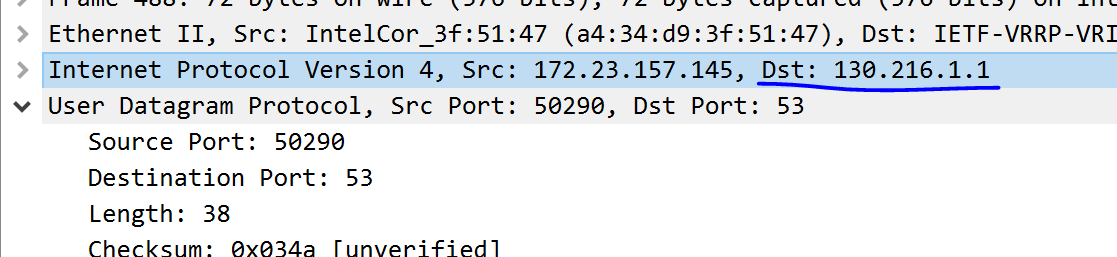
* 1. Locate the DNS query and response messages. By exploring the Packet detail pane in Wireshark, Are DNS messages being sent over UDP or TCP?

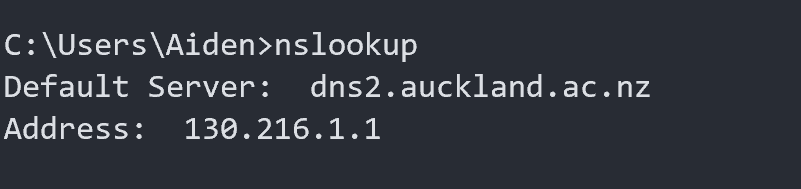
DNS messages are being sent over UDP

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

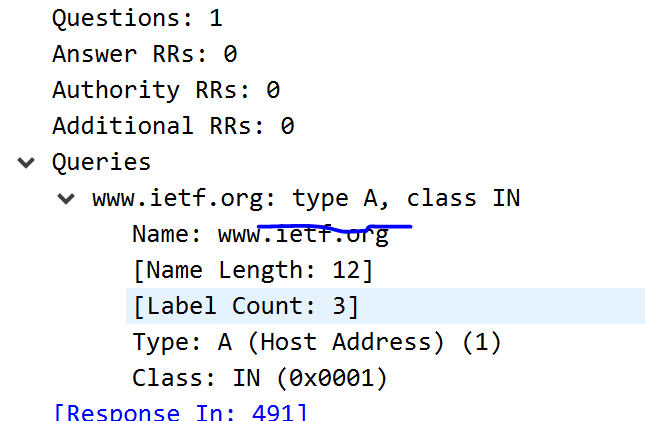
50290 is the destination port

53 is the source port

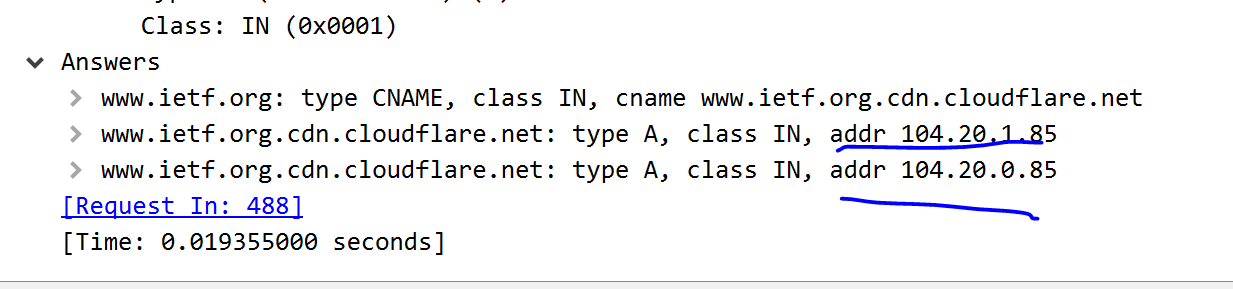
* 1. What IP address is the DNS query message sent? Use nslookup to determine the IP address of your local DNS server. Are these two IP addresses the same?

The DNS query message is sent to IP 130.216.1.1

The local DNS server IP is also 130.216.1.1

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

The DNS query is of type A. There are no answers in the query message.

* 1. Examine the DNS response message. How many “answers” are provided? What does each of these answers contain? List these *(hint: the first is Name)*

There are three answers. The first answer gives us the canonical name for [www.ietf.org](http://www.ietf.org). The second answer contains an IP address for [www.ietf.org](http://www.ietf.org). And the third answer contains an IP address for www.ietf.org.

* 1. [www.ietf.org](http://www.ietf.org) is a Canonical name, what does that mean?

[www.ietf.org](http://www.ietf.org) is not a canonical name, but rather an alias. This is a more human readable form of the root/actual name. The canonical name is [www.ietf.org.cdn.cloudflare.net](http://www.ietf.org.cdn.cloudflare.net)

## Exercise 4

**Objectives of Exercise 4: Real world scenarios**

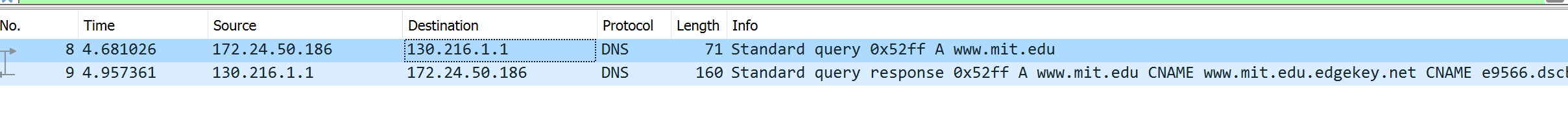
When client report poor internet response times, you should verify that DNS is operating efficiently. If the name of a webpage takes too long to resolve and for security investigation to determine abnormal DNS behaviour, the following are some of the things to look out for.

1. How many packets are exchanged in the data transfer? How many packets are transmitted for each UDP datagram? What is the size of the UDP payload of these packets?
2. The DNS response time

You are provided with 2 DNS capture files, using these files, you are required to determine whether the DNS operation was successful or not using the information provided and other Wireshark tools.

## Exercise 4.1 plus questions

Open “captureDNS1.pcap” and set the filter to DNS

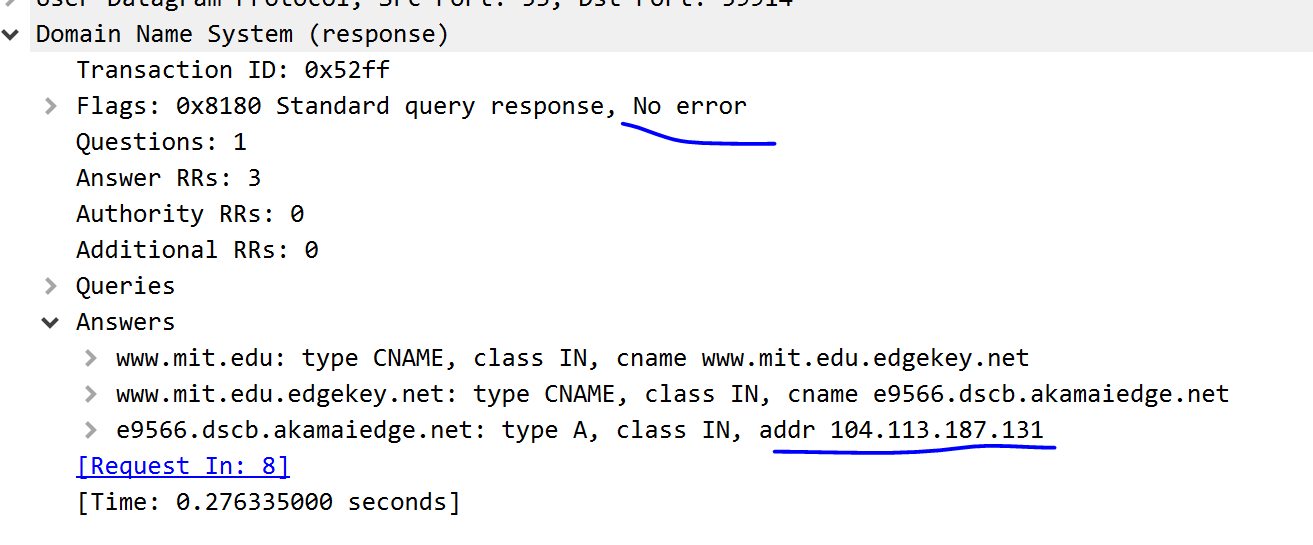
* + 1. Analyse the 2 packets by identifying the DNS message type. Which packet is the response packet? Which packet is sending out the request?

The packet sending out the request is the top one (8).

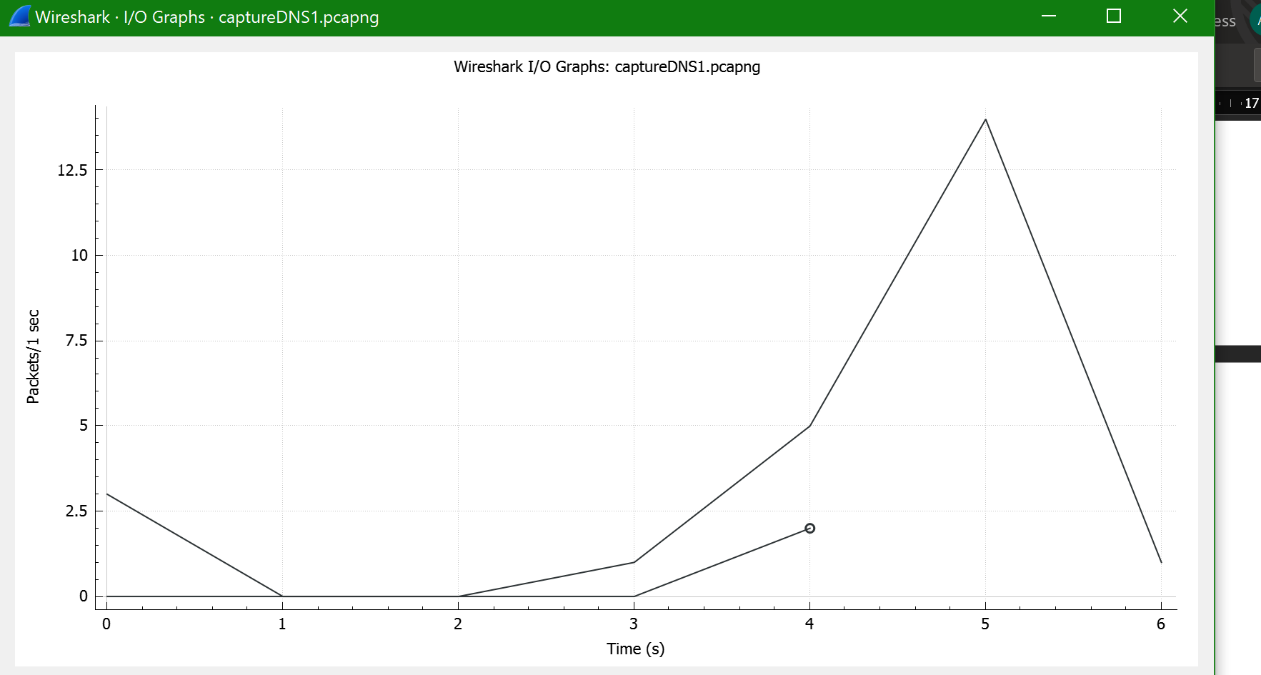
The response packet is the bottom one (9).

* + 1. What is the DNS response time? Here, longer response time correspond to DNS request time out (in windows the default time out is 2 seconds)

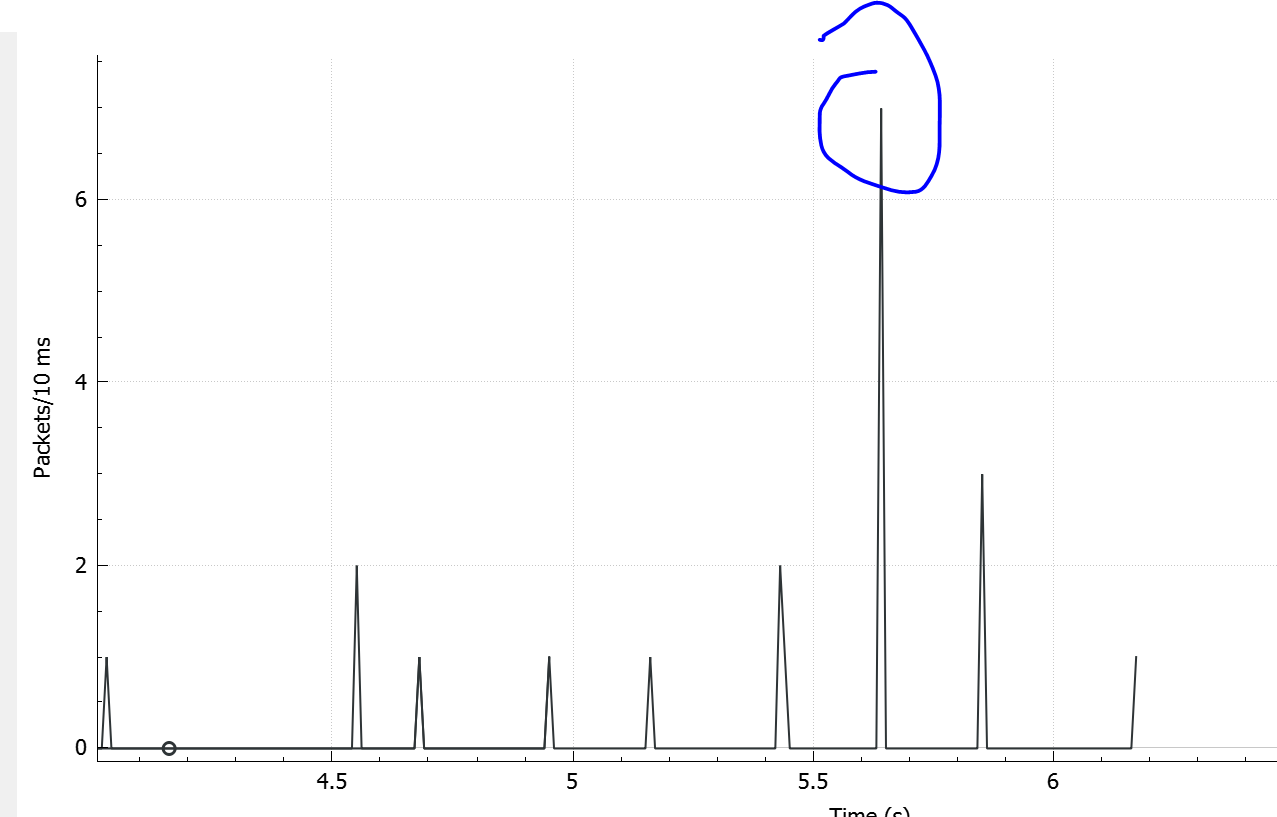
The DNS response time is 4.957361 – 4.681026 = 0.276s or 276ms

* + 1. By looking at this two packets, did the server successfully resolve the domain name requested from it? Give reasons for your answers

The server did successfully resolve the domain name. There were no error flags raised and there is an ip resolved in the answers.

* + 1. Check the IO Graphs, what is the peak packets/second of this communication and at what time? What is the peak packets/10msec of the communication and at what time? What information do these two plots show?, and do the two plots contradict each other?

The peak packets/second is 14 packets/second at time 5s.



The peak packets/10msec is 7/10msec at around time 5.6s

The plots show the amount of traffic on the client. The plots do not directly contradict each other, but the 10msec plot is more relevant and shows in more detail what traffic is like on the client. If we were looking at a server which has more consistent traffic, the 1sec plot would be more relevant.

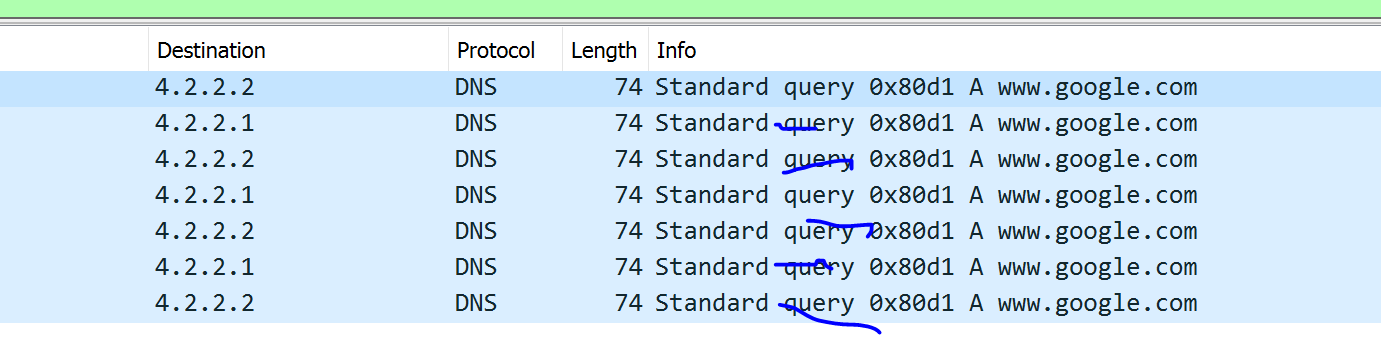
* + 1. Open the flow graph, (set show to “Displayed packets”), attach a picture of this and list the source and destination port being used

Source port: 59914

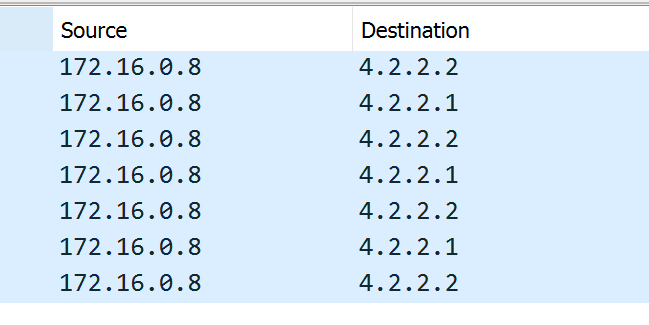
Destination port: 53

## Exercise 4.2 plus questions

Open “captureDNS2.pcap” and set the filter to DNS. The IP address 4.2.2.2 and 4.2.2.1 is the public DNS which anyone can use (note: there are other public types too)

* + 1. Name the DNS message type available in this capture *(hint: look in info in the Packet List Pane)*

There are only DNS queries available in the capture.

* + 1. What is the source and destination IP address in this Query

Source IP: 172.16.0.8

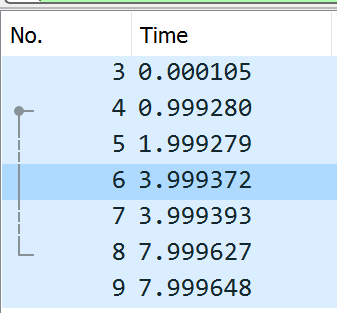
Destination IP: 4.2.2.1 or 4.2.2.2

* + 1. What is the web address of the server being queried?

[www.google.com](http://www.google.com) is being queried

* + 1. In this capture, did the DNS server ever respond to the source’s request? Give reasons for your answer

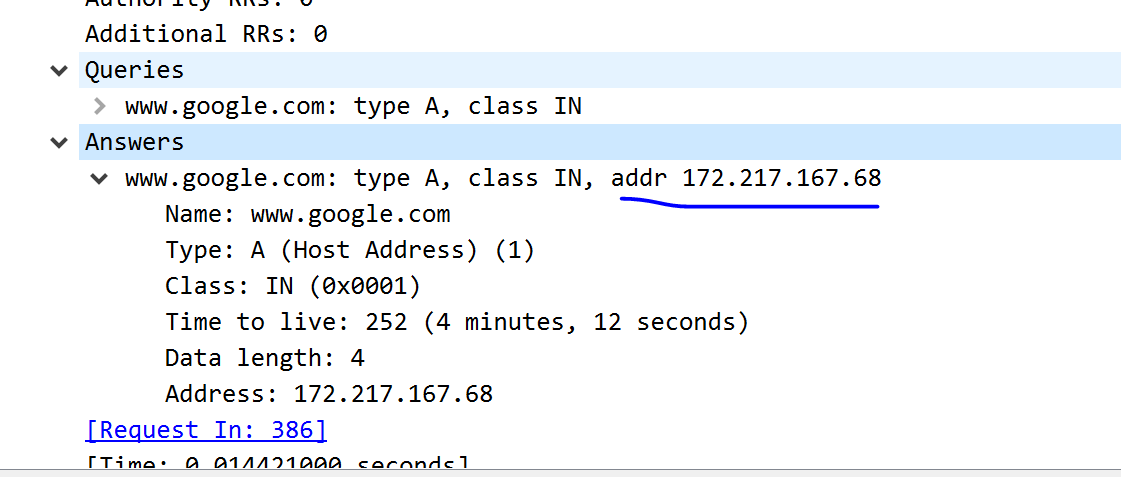
No the DNS server did not ever respond to the source’s request. In the capture, there are no DNS responses. Also, it is clear by the 1, 2, 4sec delay between requests, that the requests have been timed out and restarted, further adding evidence that there was no response.

* + 1. Can you identify what the cause of this problem is and the proposed solution? Comparing with “captureDNS1.pcap” above

There public DNS server is down/not responding. This may be due to a variety of issues: DDoS, maintenance issues. The proposed solution is to try a different DNS server.

* + 1. The DNS server when you expand Queries says type A, in a sentence or 2 briefly explain what information would be returned from a Type A query.

A type A query asks for an IP address for a given host name. The server will would return an IP address for the given host name.

* + 1. Use Wireshark to capture the DNS response for a Type A query for www.google.com. Use a screenshot of the appropriate portion of the Packet Details pane to support your answer in 4.2.6.

The response gives the corresponding IP address for [www.google.com](http://www.google.com)