**Luke Pepin - CSE3300: Computer Networking**

Homework 4

Due Date: 10/10/2024, Thursday. Submission through HuskyCT.

Full score: 100 for CSE3300 students; 120 for CSE5299 students (will be normalized to 100 when entering the grade in HuskyCT).

1. **DNS (30 points).** In this problem, we will use the dig tool available on Unix and Linux hosts to explore the hierarchy of DNS servers.

Recall that DNS has a hierarchical architecture – a DNS server higher in the DNS hierarchy delegates a DNS query to a DNS server lower in the hierarchy, by sending back to the DNS client the name of that lower-level DNS server.

First read the man page for dig at https://linux.die.net/man/1/dig, and then answer the following questions.

* + 1. If you run dig cse.uconn.edu, you will get an answer with the IP address for cse.uconn.edu immediately. If you run dig +trace cse.uconn.edu, you will get an answer for the IP address of cse.uconn.edu, together with the sequence of DNS servers that were queried (because of the +trace option). You will notice that the hierarchy of the DNS name servers is similar to what is shown in Fig. 1. Specifically, it contains a list of root DNS servers, followed by a list of .edu TLD DNS servers, and then a list of authoritative DNS servers for cse.uconn.edu.

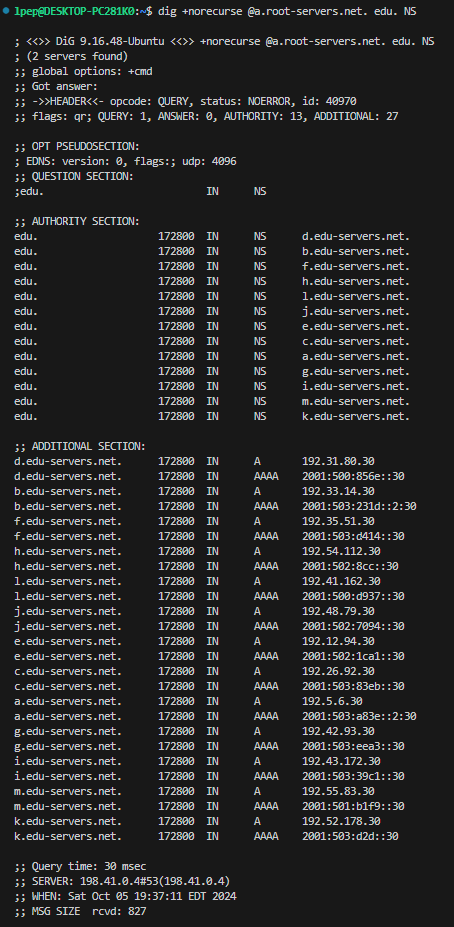
In this question, instead of using the +trace option, you will manually traverse the DNS hierarchy one level at a time to eventually obtain the IP address of cse.uconn.edu by using dig. Specifically, you should use a sequence of dig commands, first query a root server to get a list of .edu TLD servers, then query an .edu TLD server to get a list of authoritative servers, and then query an authoritative server to resolve the IP address for cse.uconn.edu. In other words, what you are doing is similar to steps 2-3, steps 4-5, and steps 7-8 as shown in Fig. 1(a).

*Hint:* The default mode of dig is doing a recursive type of query resolution. To be able to achieve the above step-by-step query, you will need to select the proper parameters in dig to disable recursive query. You can use @server option in dig to pass in the DNS server that you want to query.

Show the list of your commands. For each command, show the names of DNS servers in the delegation chain in answering your query. Please include screenshots.

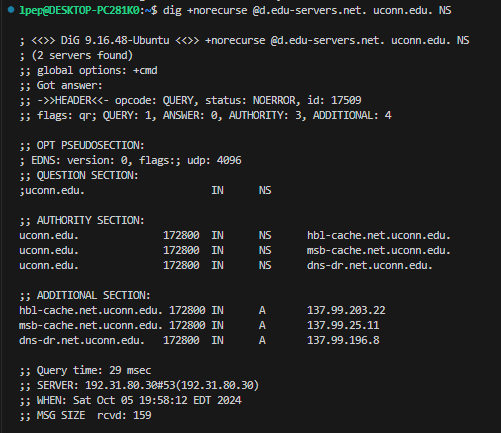
I. dig +norecurse @a.root-servers.net. edu. NS

Query of a root server to get a list of .edu TLD servers ‘a.root-servers.net’



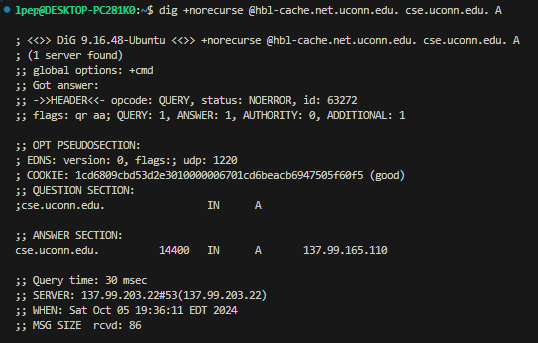
II. dig +norecurse @a.edu-servers.net. uconn.edu. NS

Query of an .edu TLD server to get a list of authoritative servers ‘uconn.edu’



III. dig +norecurse @hbl-cache.net.uconn.edu. cse.uconn.edu. A

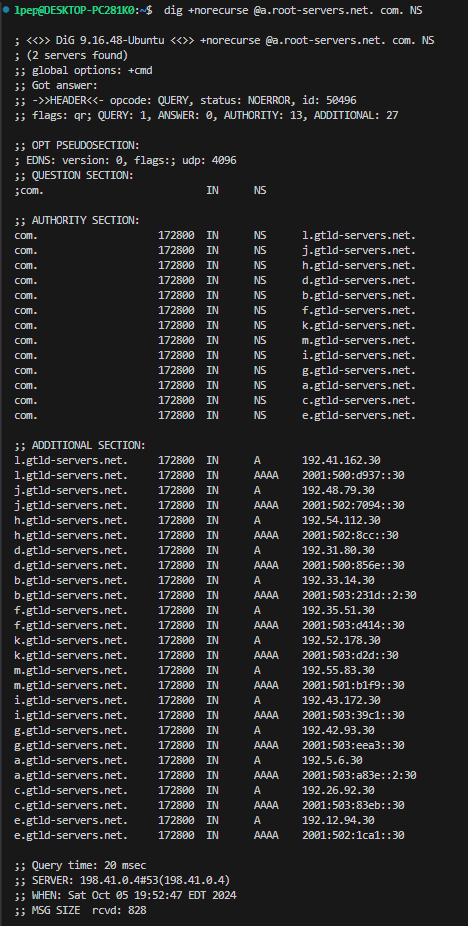
Query of an authoritative server to resolve the IP address for cse.uconn.edu (137.99.203.22)



* + 1. Repeat part a) for google.com. Please include screenshots.

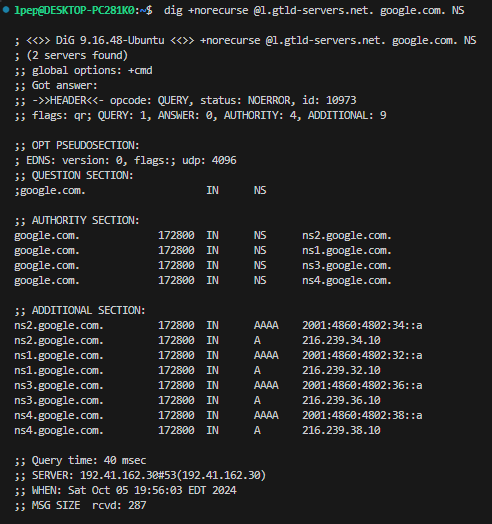
I. dig +norecurse @a.root-servers.net. com. NS

Query of a root server to get a list of .edu TLD servers ‘a.root-servers.net’



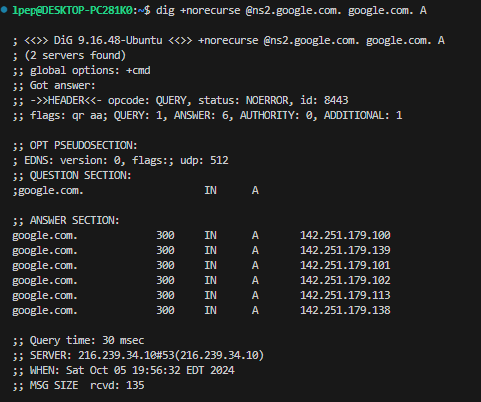
II. dig +norecurse @l.gtld-servers.net. google.com. NS

Query of an .edu TLD server to get a list of authoritative servers ‘google.com



III. dig +norecurse @ns2.google.com. google.com. A

Query of an authoritative server to resolve the IP address for google.com (216.239.34.10)



1. **Email and SMTP (10 points).** What is the difference between MAIL FROM: in SMTP and FROM: in the mail message itself? Use an example to explain the difference.

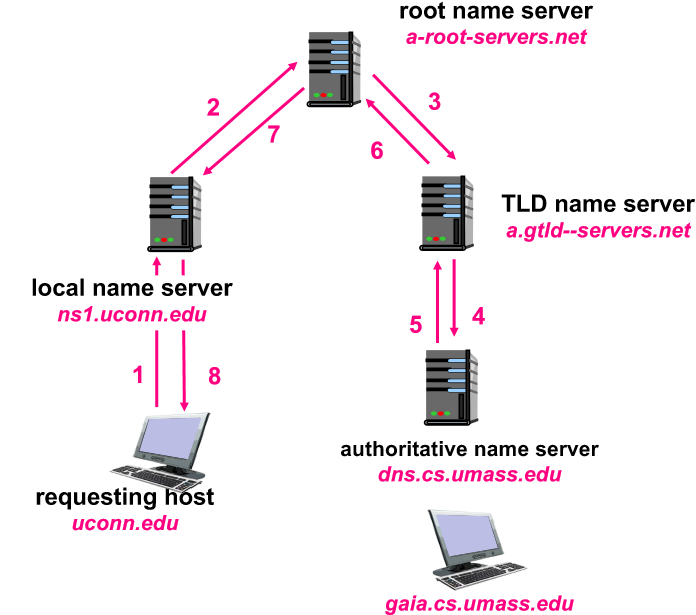
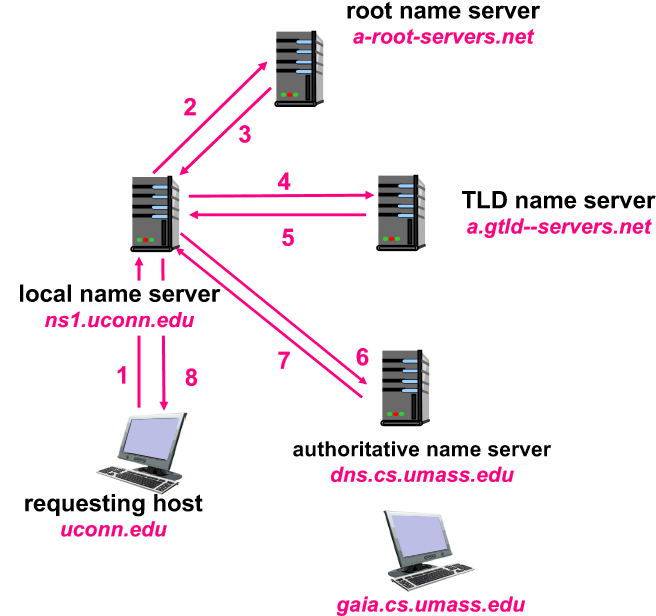


Figure 1: (a) Iterative DNS resolution, (b) Recursive DNS resolution.

The difference between ‘MAIL FROM:’ in SMTP and ‘FROM:’ in the mail message itself its the difference between the piece of the protocol for sending the message and a part of the message. MAIL FROM: as used in SMTP is part of the envelope and is used by mail servers to route and deliever the email. While FROM in the mail message is part of the content and is included for the recipient.

Example:

SMTP Session:

MAIL [FROM:<luke.pepin@uconn.edu](mailto:FROM:<luke.pepin@uconn.edu)>

RCPT [TO:<bing@uconn.edu](mailto:TO:<bing@uconn.edu)>

DATA

From: Luke Pepin [<luke.pepin@uconn.edu](mailto:<luke.pepin@uconn.edu)>

To: Bing Wang [<bing@uconn.edu](mailto:<bing@uconn.edu)>

Subject: HW4

-Content Abbreivated-

-

QUIT

As seen in Professor Wang’s inbox:

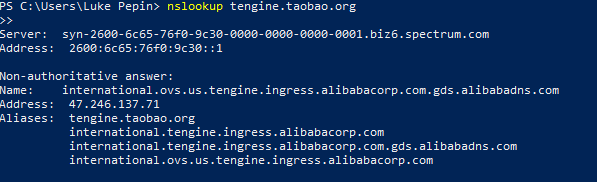
From: Luke Pepin [<luke.pepin@uconn.edu](mailto:<luke.pepin@uconn.edu)>

To: Bing Wang [<bing@uconn.edu](mailto:<bing@uconn.edu)>

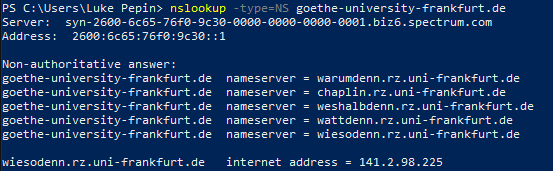
Subject: HW4

1. **Wireshark labs (30 points).** Answer questions 1-8 of the DNS Wireshark lab in HuskyCT.

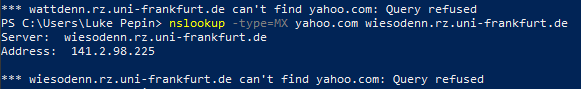
1. Run nslookup to obtain the IP address of a Web server in Asia. What is the IP address of that server? 47.246.137.71

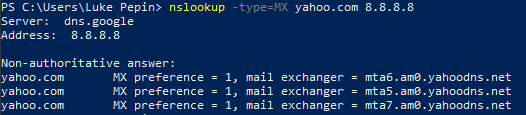


2. Run nslookup to determine the authoritative DNS servers for a university in Europe.

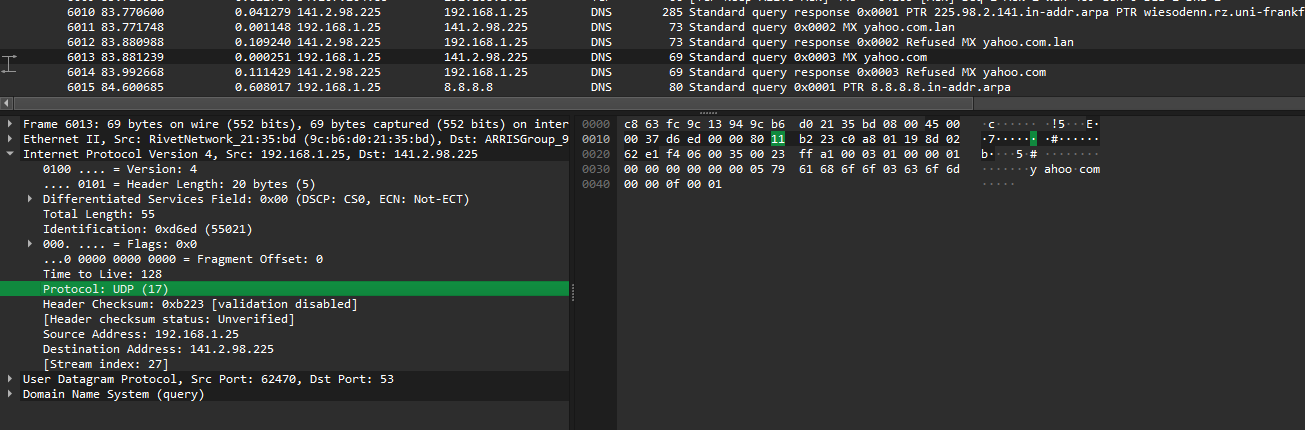


3. Run nslookup so that one of the DNS servers obtained in Question 2 is queried for the mail servers for Yahoo! mail. What is its IP address? Inconclusive for German University, Google’s public DNS server (8.8.8.8) on the otherhand ...

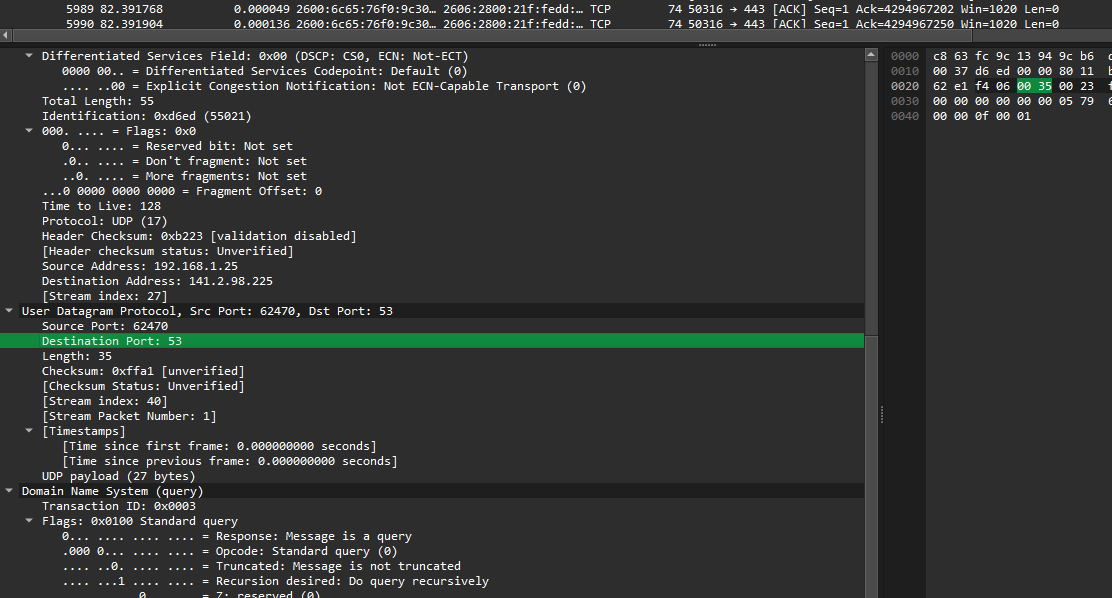




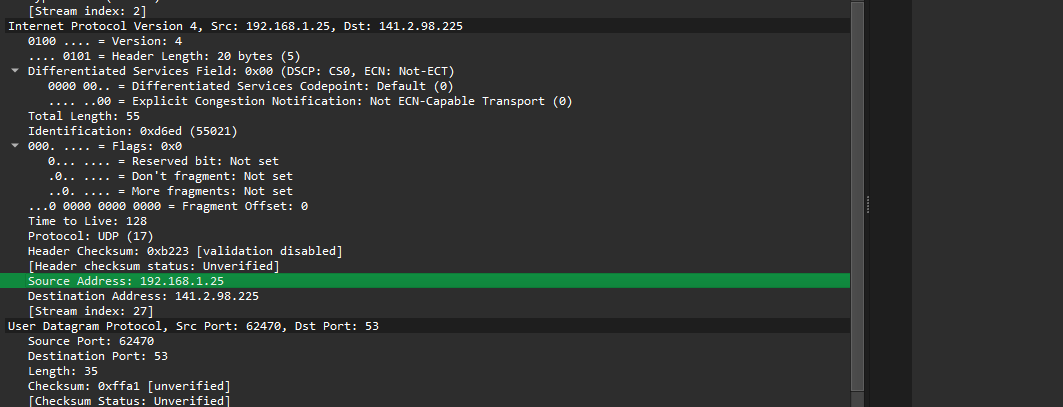
4. Locate the DNS query and response messages. Are then sent over UDP or TCP? UDP (17)

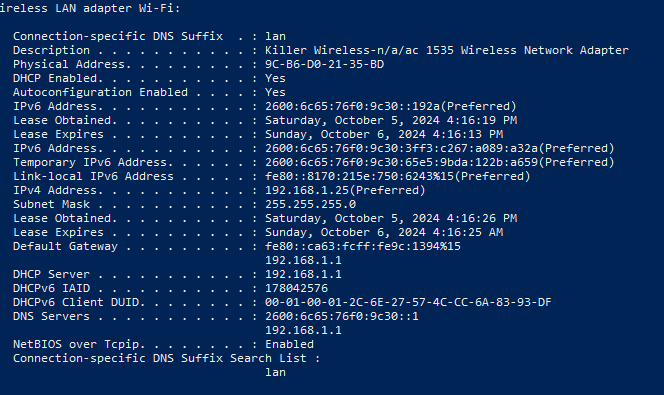


5. What is the destination port for the DNS query message? What is the source port of DNS response message? Destination Port: 53, Source Port: 62470

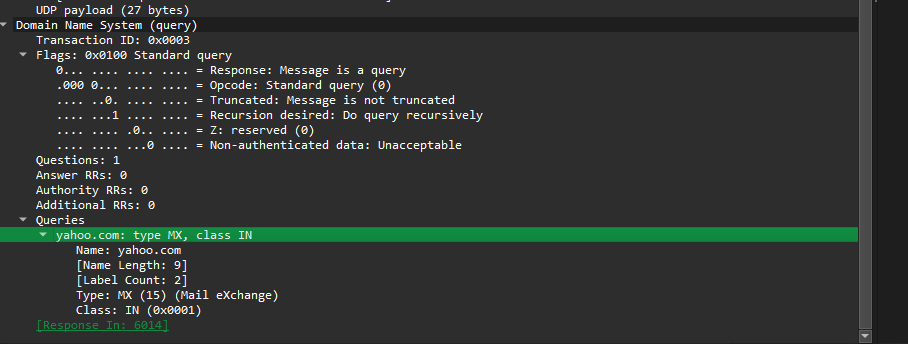


6. To what IP address is the DNS query message sent? Use ipconfig to determine the IP address of your local DNS server. Are these two IP addresses the same? Destination Address: 141.2.98.225, Local DNS server: 192.168.1.1, different

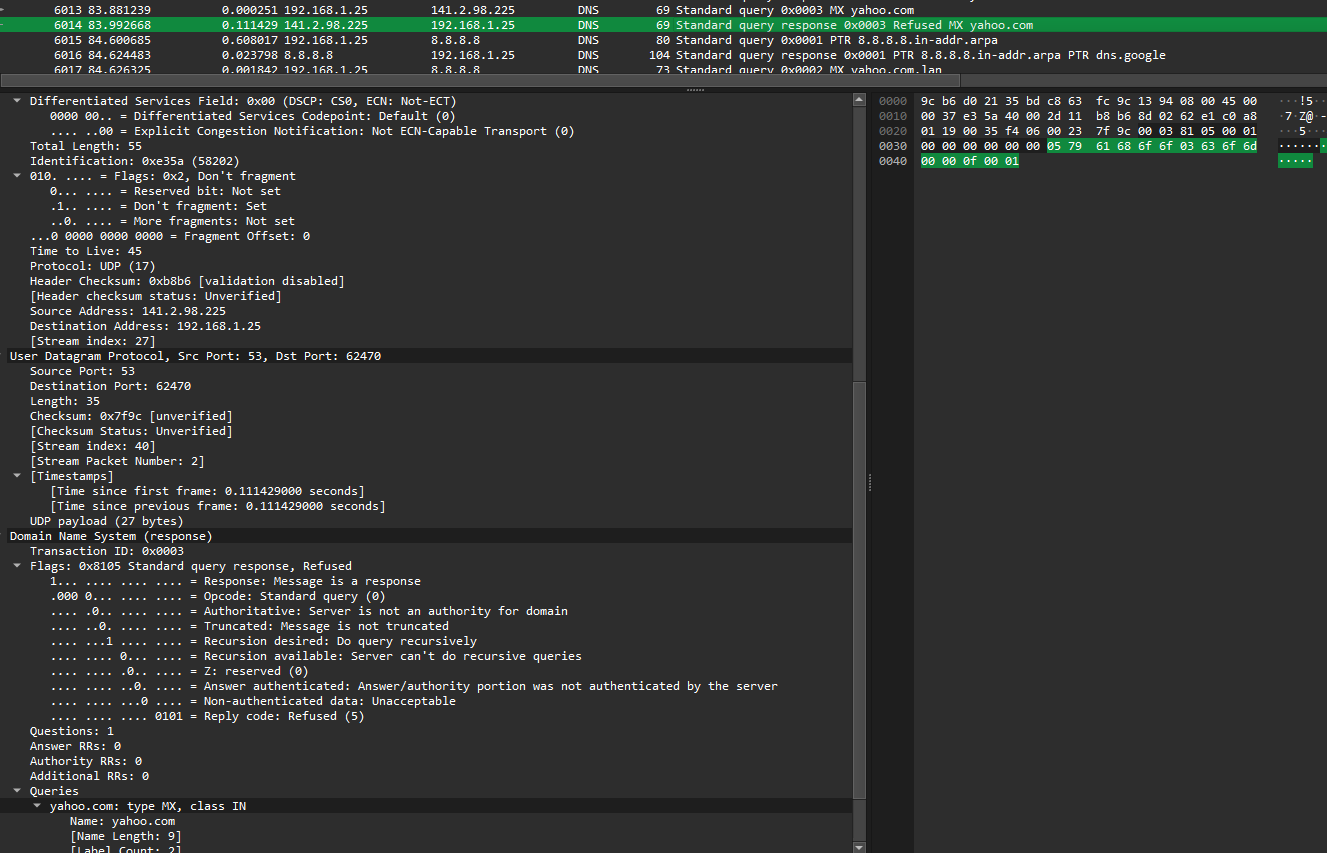




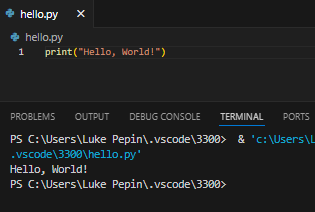
7. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”? MX, Mail Exchange record query. The query does not contain answers it only conatins the section askign for the MX records.



8. Examine the DNS response message. How many “answers” are provided? What do each of these answers contain? Answers are not provided as this is a refused response message. They contain only the response infomation that it is refused

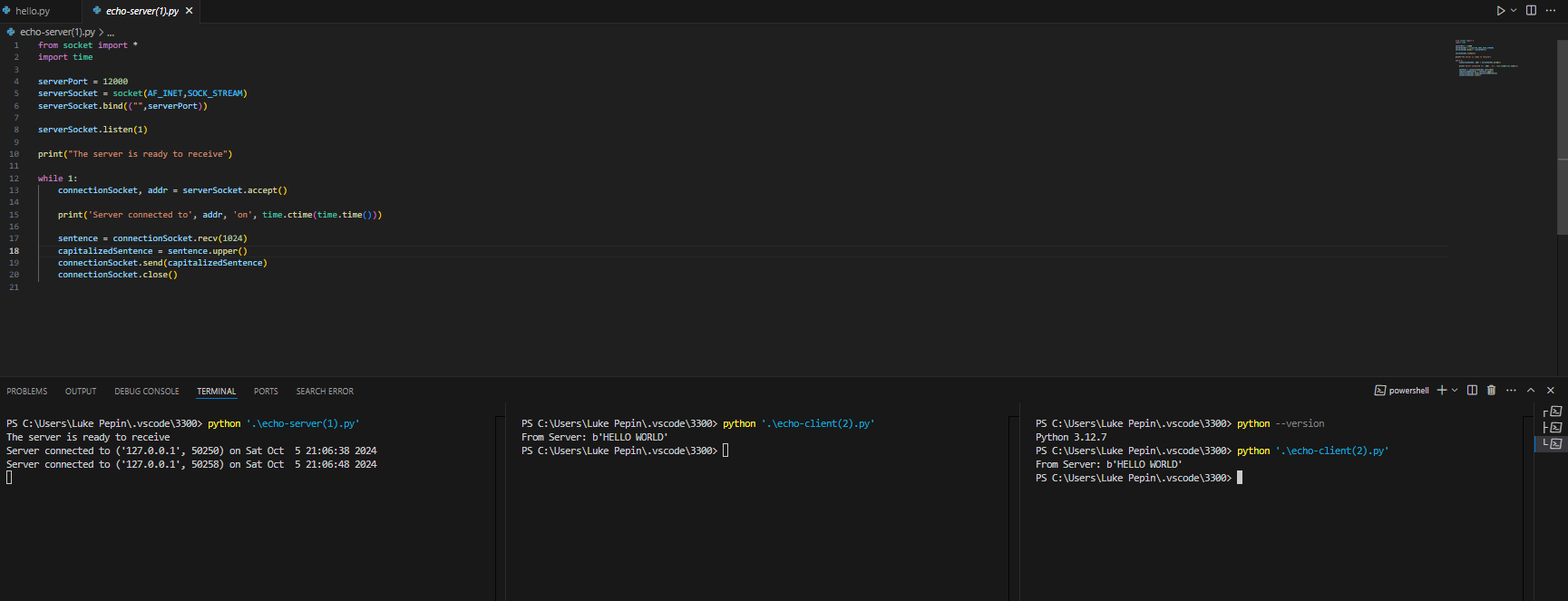


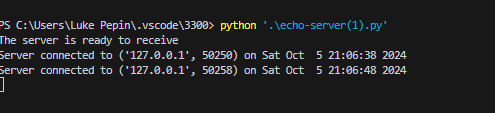
1. **Socket programming (30 points).** The goal of this problem is to help you get ready for the first programming assignment, which will be assigned after this homework.
   * **Set up environment.** You will need to have an environment that you can run python. Ideally, you want to run python3.



* + **Test sample code (the simple version). (10 points)** You can first test a simple version of echo-server and echo-client (attached to this homework) that does not have much in handling errors. Run the server in one terminal. Start a client in another terminal. After that, start another client. (1) Use screenshots to show what you have done and describe the clients’ port numbers. (2) Modify the source code so that the client requires entering a phrase from the keyboard. Again use screenshots to show what you have done (include the source code of the client that you have modified).

1.

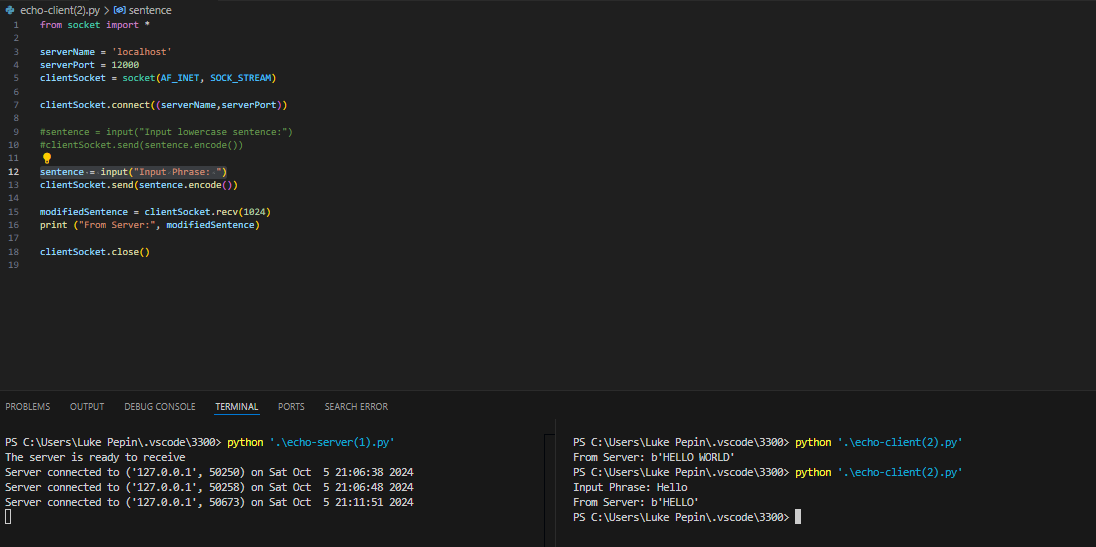
Zoom in of prev image



Port numbers 1. 50250, 2. 50258

2.



Altered code attached with submission: echo-client-lp.py

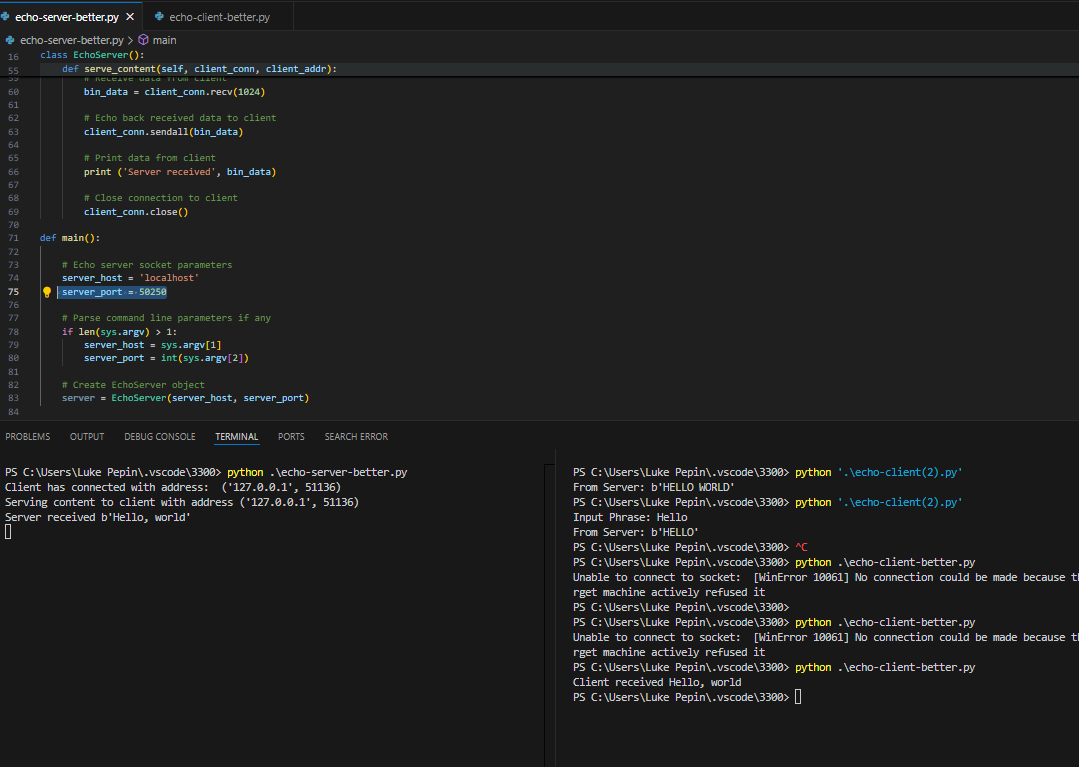
* + **Test sample code (the better version). (20 points)** Test echo-server-better and echo-client-better (attached to this homework) that include code in handling errors and has more elaborate code. The comment in the code suggests using python3, but you can run the code using python. Run the server in one terminal. Start a client in another terminal. (1) Run the client as python echo-client-better.py. Why is it not working? Please be very specific in explaining what is wrong. (2) Run the client in another way to make it talk to the server successfully. Describe the command line to run the client properly.

1.



This error occurred because the client was trying to connect to a port that the server was not listening on. Specifically, the port number set in the client code (50006) did not match the port number on which the server was running (50008).

2.



By ensuring that both the client and server are using the same port number (50250), the client was able to successfully connect to the server and communicate as expected. The execution of the code was the same as previously python ./echo-server-better.py and python ./echo-client-better.py in different terminals at the same time. However the change to the port variable in both codes to 50250 allowed the code to run properly.