CS-335a: Computer Networking

Department of Computer Science, University of Crete
Fall 2024
Deadline: 08/11/2024 **12pm (afternoon, before the lab/lecture)**
Professor: Maria Papadopouli
TA: Christina Brozi (brozi@csd.uoc.gr)

SUBMISSION GUIDELINES

Your report should be in **pdf** format Please submit your assignment via email to the TA. The subject of the email should be "**335a_assign2_csdXXXX**". You can submit your assignment as many times as needed. Only the last submission will be graded. The maximum grade you can get is 115 with 15 out of the 115 points being BONUS.

Assignment 2 - Application Layer

At the Top!

Exercise 1 - HTTP Theoretical (40 points)

i) Why do HTTP, FTP, SMTP and POP3 run on top of TCP rather than UDP? (2p)

These protocols require reliable data transfer.

ii) How does persistent HTTP improve web performance ? What are the potential drawbacks of persistent HTTP connections? (3p)

Reduces latency because the client doesn't have to establish a separate TCP connection for each HTTP request/response pair. It also improves throughput since multiple objects can be sent over the same TCP connection.

Potential drawbacks of persistent HTTP connections can be:

- Session hijacking: attackers can steal a web user's active session by acquiring their unique session ID. This lets them impersonate the user, accessing data or performing actions as if they were the legitimate user.
- If not managed properly they can lead to problems with resource allocation: resources may be kept occupied even when not needed and may not be available to others.

iii) Explain the concept of statelessness in HTTP. How does it affect the server's workload? (5p)

HTTP is a “stateless” protocol, meaning that each HTTP request is independent of any previous or subsequent requests. The server does not maintain any information about the client's state or history between requests.

This helps the server’s overload since the server can handle more concurrent requests and responses, because they do not consume any resources or memory on the server. Also, web servers can be scaled more easily, as each request can be handled independently.

iv) How can web caches be used to improve user experience in remote and poorly connected areas? List one drawback of the use of Web caches. (5p)

Web caching can bring the desired content “closer” to the user, possibly to the same LAN to which the user’s host is connected.

If caching is not set up correctly the browser might not be able to validate the cached content and the page may load outdated content.

Exercise 2 - HTTP GET/RESPONSE (15 points)

A. Consider the following string of ASCII characters that were captured by Wireshark when the browser sent an **HTTP GET** message (i.e., this is the actual content of an HTTP GET message). The characters <cr><lf> are *carriage return* and *line-feed* characters. **Answer the following questions, indicating where in the HTTP GET message below you find the answer.**

i) What is the name of the file that is being retrieved in this GET message?

quotation4.htm

ii) What version of HTTP is the client running? *HTTP/1.1*

iii) What type of files does the client accept? *Html, xml, png etc*

iv) What is the client's preferred version of English? *American English*

v) What is the client's least preferred version of English? *English*

vi) Will the client accept the German language? *Yes (de)*

vii) Does the client already have a cached copy of the file? *Yes (If-Modified-Since)*

viii) Does the browser request a non-persistent or a persistent connection?

Persistent (keep-alive)

ix) What is the IP address of the host on which the browser is running?

No such information is provided in the HTTP GET message

x) What type of browser initiates this message? Why is the browser type needed in an HTTP request message?

Browser: Mozilla 5.0. Since some webpages might have different versions depending on the browser, the server needs to know this type of information in order to send the correct version of a web page to the client.

```
GET /kurose_ross_sandbox/interactive/quotation4.htm
HTTP/1.1<cr><lf> Host: gaia.cs.umass.edu<cr><lf> Accept:
text/plain, text/html, text/xml, image/gif, image/png, audio/mpeg,
audio/vnf.wave, video/mp4, video/mpeg<cr><lf> Accept-Language:
en-us, en-gb;q=0.6, en;q=0.4, fr, fr-ch, da, de<cr><lf>
Keep-Alive: 300<cr><lf> Connection:keep-alive<cr><lf>
If-Modified-Since: Mon, 21 Oct 2024 02:05:42 -0700<cr><lf>
```

```
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:11.0)
Gecko/20100101 Firefox/11.0<cr><lf><cr><lf>
```

B. Consider now the following string of ASCII characters that were captured by Wireshark when the browser sent an **HTTP RESPONSE** message. **Answer the following questions, indicating where in the HTTP RESPONSE message below you find the answer**.

i) Was the server able to send the document successfully? *No, 404 Not Found*

ii) What time was the document reply provided? *Mon, 21 Oct 2024 09:51:44 +0000*

iii) How big is the document in bytes? *59780*

iv) Is the connection persistent or nonpersistent? *Persistent*

v) What is the type of file being sent by the server in response? *image/html*

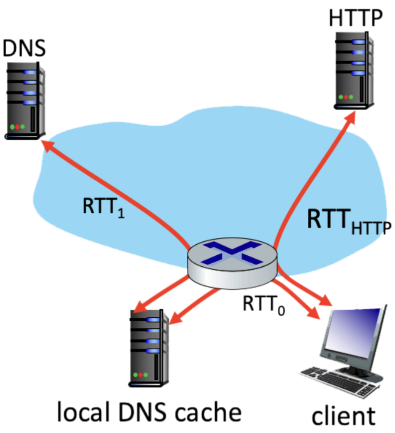
vi) What is the name of the server and its version? *Apache/2.2.3*

vii) Will the ETag change if the resource content at this particular resource location changes? *Yes*

```
HTTP/1.1 404 Not Found<cr><lf> Date: Mon, 21 Oct 2024 09:51:44
+0000<cr><lf> Server: Apache/2.2.3 (CentOS)<cr><lf> GMTETag:
”526c3-f22-a88a4c80”AcceptRanges: bytes<cr><lf> Content-Length:
59780<cr><lf> Keep-Alive: timeout=50, max=96<cr><lf> Connection:
Keep-alive<cr><lf> Content-type: image/html<cr><lf><cr><lf>
```

Exercise 3 - HTTP Response Time (20 points)

Suppose within your Web browser you click on a link to obtain a Web page. The IP address for the associated URL is not cached in your local host, so a DNS lookup is necessary to obtain the IP address. Suppose that two DNS servers are visited before your host receives the IP address from DNS. The first DNS server visited is the local DNS cache, with an RTT delay of RTT0 = 2 msecs. The second DNS server contacted has an RTT of 49 msecs. Initially, let's suppose that the Web page associated with the link contains exactly one object, consisting of a small amount of HTML text. Suppose the RTT between the local host and the Web server containing the object is RTTHTTP = 17 msecs.



i) Assuming zero transmission time for the HTML object, how much time (in msec) elapses from when the client clicks on the link until the client receives the object? (4p)

T_1 = RTT0 + RRT1 + 2RTT$_{HTTP}$ = 85 ms

ii) Now suppose the HTML object references 3 very small objects on the same server. Neglecting transmission times, how much time (in msec) elapses from when the client clicks on the link until the base object and all 3 additional objects are received from the web server at the client, assuming non-persistent HTTP and no parallel TCP connections? (4p)

Receiving the HTML body requires T_1 = 85ms. For the 3 images we need additionally 3 HTTP requests, therefore:

$T_2 = T_1 + 3*2*RTT_{HTTP}$ = 85 + 102 = 187ms.

iii) Suppose the HTML object references 3 very small objects on the same server, but assume that the client is configured to support a maximum of 5 parallel TCP connections, with non-persistent HTTP. (4p)

$T_3 = T_1 + 2*RTT_{HTTP}$ = 85 + 34 = 119ms

iv) Suppose the HTML object references 3 very small objects on the same server, but assume that the client is configured to support a maximum of 5 parallel TCP connections, with persistent HTTP. (5p)

$T_4 = T_1 + RTT_{HTTP}$ = 85 + 17 = 102ms

v) What are the differences between these methods: (3p)

a) Non-Persistent HTTP (without parallel connections)
b) Persistent HTTP without pipelining

c) Persistent HTTP with pipelining

Non-Persistent HTTP (without parallel connections)	Persistent HTTP without pipelining	Persistent HTTP with pipelining
A TCP connection must be established for each HTTP request	A single TCP connection is maintained for multiple HTTP requests and responses	Multiple HTTP requests can be sent over a single TCP connection without waiting for responses to previous requests.
Sequential requests: before you send a new request you must first wait for a response	Sequential requests: before you send a new request you must first wait for a response	Pipelined requests: requests can be sent without waiting for the responses

Exercise 4 - DNS Theoretical (15 points)

i) Name 4 services provided by DNS (4p)

1. Translates hostnames to IP addresses
2. Host aliasing
3. Mail server aliasing
4. Performs load distribution among replicated servers

ii) What's the role of the different DNS servers? (2p)

- Root Name Servers: maintain a directory of all TLD servers and their associated IP addresses
- Top-Level Domain (TLD) Name Servers: manage the domain extensions (e.g., .com, .org, .net).
- Authoritative Name Servers: store information about specific domain names. When a user queries a domain, these servers provide the corresponding IP address.

iii) Why isn't a single DNS server used to handle all queries? (2p)

Using a single DNS server to handle all requests could not work for the following reasons:

1. The centralized DNS would not be able to handle large amounts of requests
2. Single point of failure: if the centralized DNS stops working, so will the internet
3. Delays in requests coming from regions that are "far" from the DNS

iv) How does DNS caching work? (2p)

A DNS cache is a temporary storage space that stores information about domain names you've recently visited. By storing DNS information locally, DNS caching enables domains to be translated or "resolved" faster, while reducing network traffic.

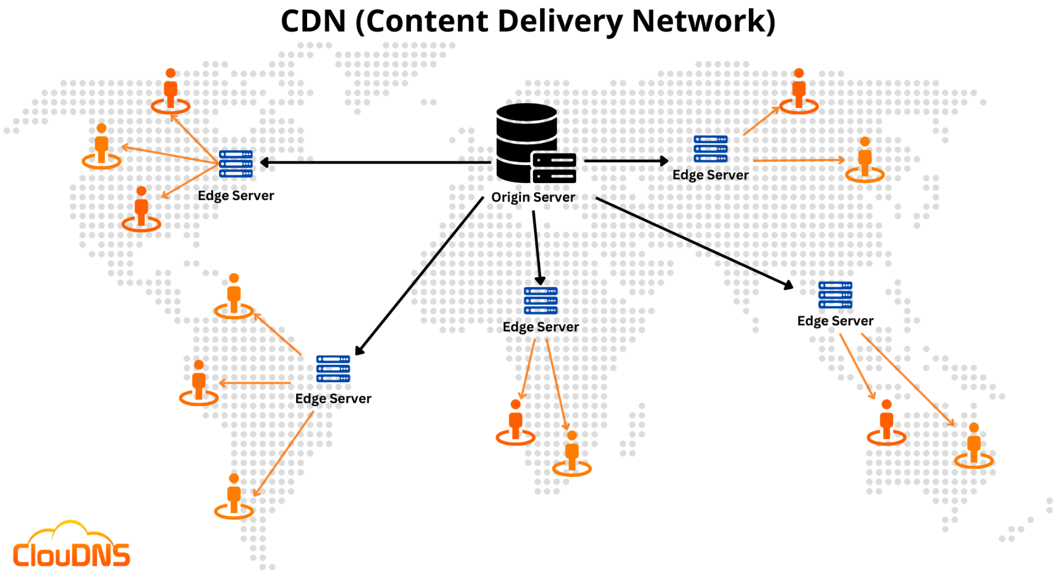
v) How does DNS load balancing work? (2p)

DNS load balancing distributes incoming network traffic across multiple servers to improve the availability, scalability, and performance of a service or application. It works by using the Domain Name System (DNS) to direct users to different IP addresses for the same domain name.

vi) Explain the role of DNS in content delivery networks (CDNs). (3p)

Content Delivery Network (CDN): network of geographically distributed servers designed to optimize the delivery of web content, such as images, videos, scripts, stylesheets, and other assets, to users based on their location.

CDNs use a technique known as "Anycast" to route users' requests to the nearest CDN server. When a user types a URL in their browser, the DNS resolution process identifies the nearest CDN server by selecting the optimal PoP based on factors such as network proximity and server health.



Exercise 5 - DNS Queries (10 points)

Suppose a web browser wants to know the IP address of www.csd.uoc.gr. Describe the name resolution process assuming:

- **i)** an iterative query is used
- **ii)** a recursive query is used

Make sure to list all the different DNS servers that are involved in the name resolution process. Which type of query is considered best practice: iterative or recursive? Explain your answer.

i) Iterative query

1) The host sends a DNS query to the local DNS server for the IP address of `www.csd.uoc.gr`.
2) The local DNS server forwards the message to the root DNS server which replies with the IP address of the TLD servers responsible for `.gr` domains.
3) The local DNS server sends the query to a TLD DNS server, which in turn replies with the IP address of the authoritative DNS server, or the IP address of an intermediate DNS server who knows the authoritative DNS server.
4) If the TLD DNS server uses an intermediate DNS server, the intermediate server replies to the local DNS server with the IP address of the authoritative DNS server.
5) The local DNS server sends the message to the authoritative DNS server, which replies with the IP address of the hostname `www.csd.uoc.gr.`

ii) Recursive query

1) The host sends a DNS query to the local DNS server for the IP address of www.csd.uoc.gr.
2) If the IP address is not found in the local DNS server, it sends a recursive query to a root DNS server.
3) The root DNS server directs the query to the appropriate TLD server (for the .gr domain).
4) The TLD server directs the query to the authoritative DNS server for the specific domain.
5) The authoritative DNS server returns the IP address to the TLD server, the TLD returns it to the root server which then returns it to the local DNS server.

Iterative queries are generally more efficient because they distribute the load among multiple servers.

Exercise 6 - Nslookup (10 points)

For this exercise you can use the department's Linux machines. ***Make sure to provide a screenshot with the output of each command that you run.***

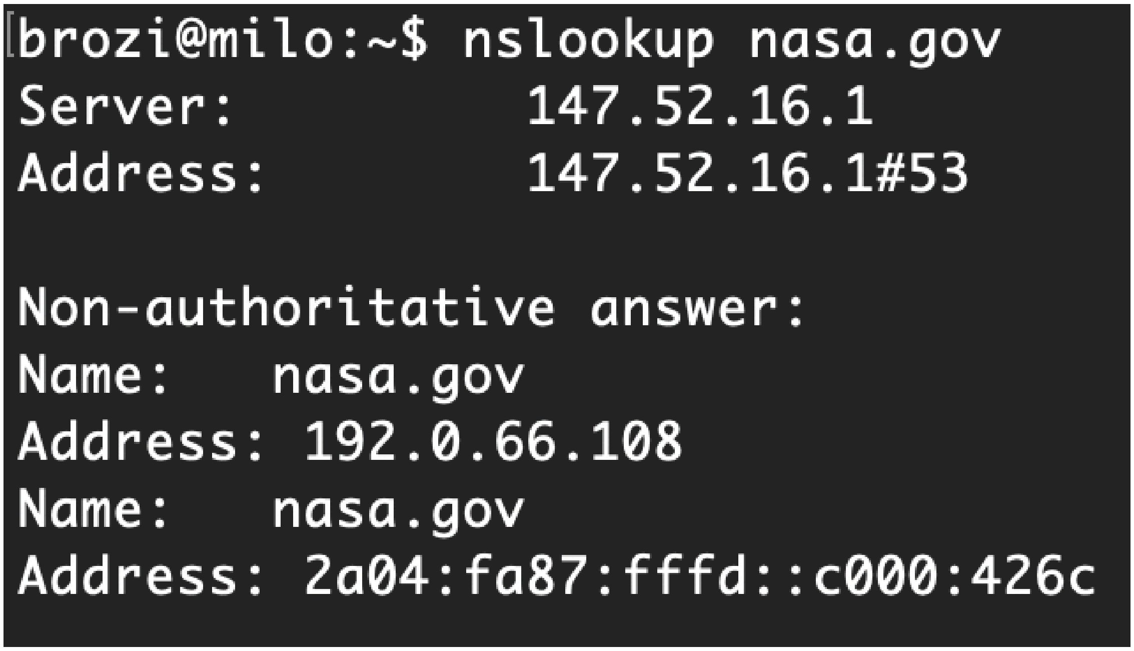
`Nslookup` is a command that operates by sending queries to DNS servers to retrieve information about domain names, IP addresses, and other DNS records.

Run the following command to retrieve the record for the host www.nasa.gov:

```
$ nslookup www.nasa.gov
```

i) What type of DNS record does this command return?
If we don’t specify the type, it will return a type A record. Type A records indicate the IP address of the given domain.

ii) What is the IP address of the DNS server that provides the information?
The local DNS server that provides the information has IP address: 147.52.16.1



```
brozi@milo:~$ nslookup nasa.gov
Server:         147.52.16.1
Address:        147.52.16.1#53

Non-authoritative answer:
Name:   nasa.gov
Address: 192.0.66.108
Name:   nasa.gov
Address: 2a04:fa87:fffd::c000:426c
```

iii) What type of DNS server provides the answer?
The answer is provided by a non-authoritative DNS server, i.e. a local DNS server

iv) What is the IP address of the host nasa.gov? How many IP addresses are provided in the answer?
The response provides 2 IP addresses: the IPv4 address 192.0.66.108 and the IPv6 address 2a04:fa87:fffd::c000:426c

Now let's try to find some information about authoritative DNS servers. Firstly, run **`nslookup`** on your terminal (without any arguments) and then you should be able to see an angle bracket prompt (>).

Run the following:

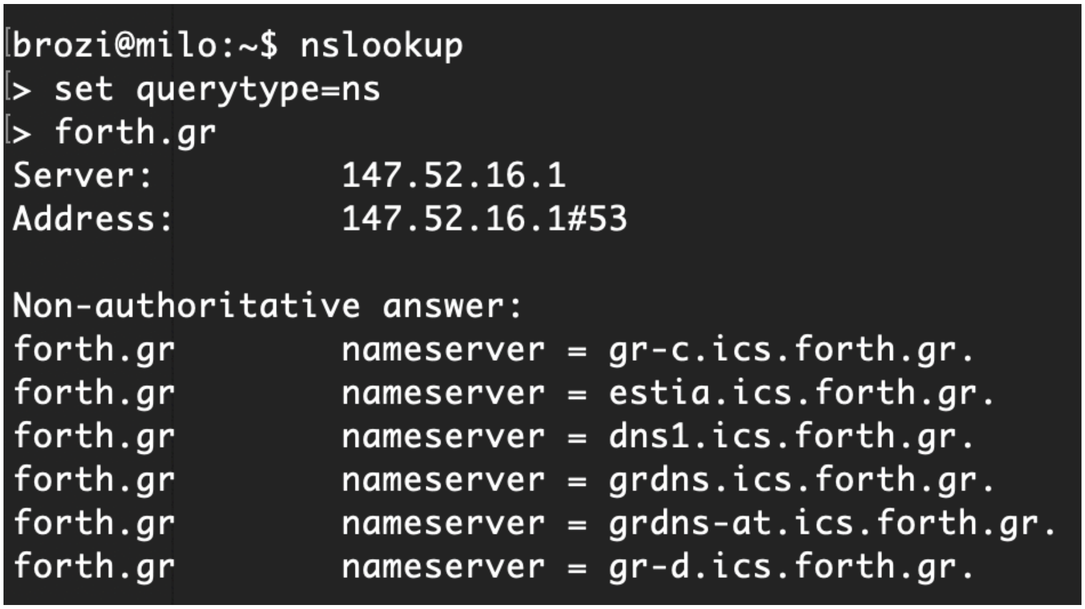
```
> set querytype=ns
> forth.gr
```

v) What does "`querytype=ns`" do? What type of DNS records does it return ?

The flag querytype=ns instructs the nslookup utility to return NS records. NS records provide information about the authoritative name servers for a domain.

vi) What information is provided in the `Non-authoritative answer` section ?

It provides information about DNS servers authoritative for the domain forth.gr



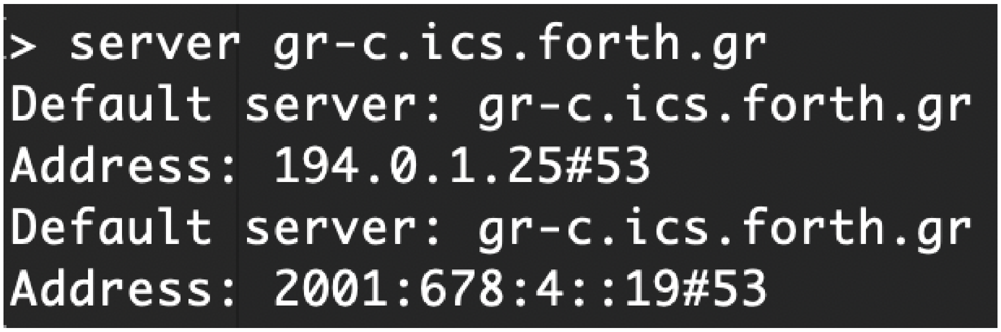
```
[brozi@milo:~$ nslookup
[> set querytype=ns
[> forth.gr
Server:         147.52.16.1
Address:        147.52.16.1#53

Non-authoritative answer:
forth.gr        nameserver = gr-c.ics.forth.gr.
forth.gr        nameserver = estia.ics.forth.gr.
forth.gr        nameserver = dns1.ics.forth.gr.
forth.gr        nameserver = grdns.ics.forth.gr.
forth.gr        nameserver = grdns-at.ics.forth.gr.
forth.gr        nameserver = gr-d.ics.forth.gr.
```

Let's try to obtain the IP address of one of the authoritative DNS servers for `forth.gr`. Use the **`server`** command, followed by the name of one of the authoritative DNS server names.

```
> server [nameserver]
> set querytype=any
```

vii) What does the `server` command do?

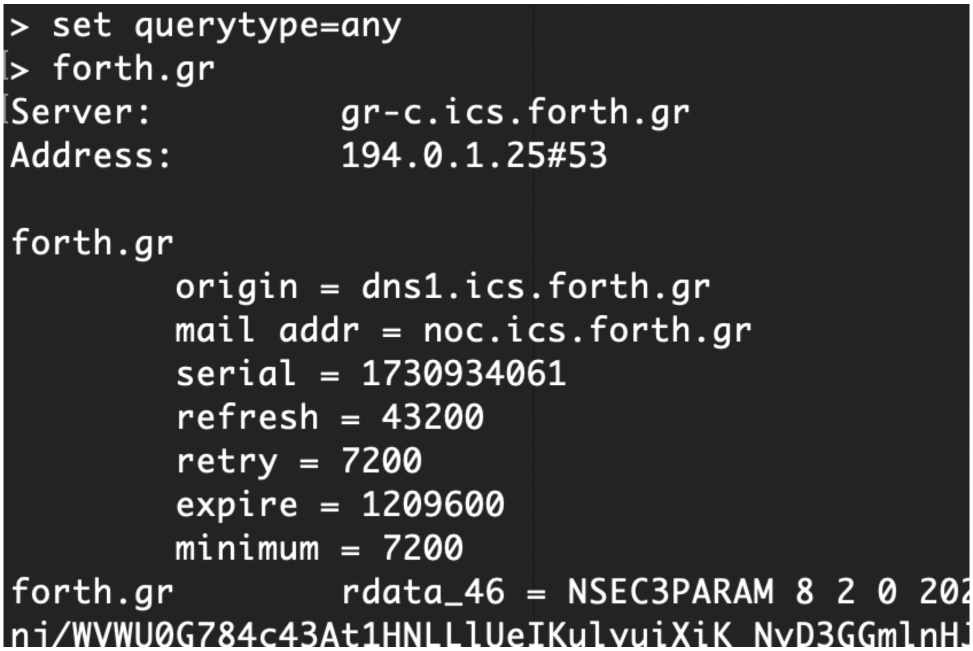
The server command specifies the DNS server to use for the query.



```
> server gr-c.ics.forth.gr
Default server: gr-c.ics.forth.gr
Address: 194.0.1.25#53
Default server: gr-c.ics.forth.gr
Address: 2001:678:4::19#53
```

viii) Run `> forth.gr` again. Which server provided the answer this time?

This time the answer is provided by the server gr-c.ics.forth.gr which is an authoritative server for forth.gr



```
> set querytype=any
> forth.gr
Server:         gr-c.ics.forth.gr
Address:        194.0.1.25#53

forth.gr
        origin = dns1.ics.forth.gr
        mail addr = noc.ics.forth.gr
        serial = 1730934061
        refresh = 43200
        retry = 7200
        expire = 1209600
        minimum = 7200
forth.gr        rdata_46 = NSEC3PARAM 8 2 0 202
nj/WVWU0G784c43At1HNLLlUeIKulyuiXiK NyD3GGmlnHJ
```

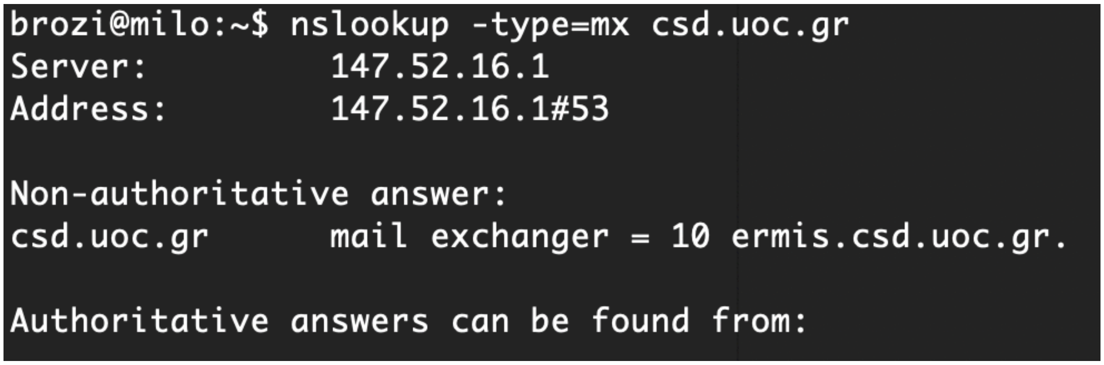
Exercise 7 - Telnet (5 points)

For this exercise we will use **telnet** to send an email. First, log in to one of the department's Unix machines. Before we start, we need to find the SMTP server of CSD. To do so, run the following command:

```
$ nslookup [flag] csd.uoc.gr
```

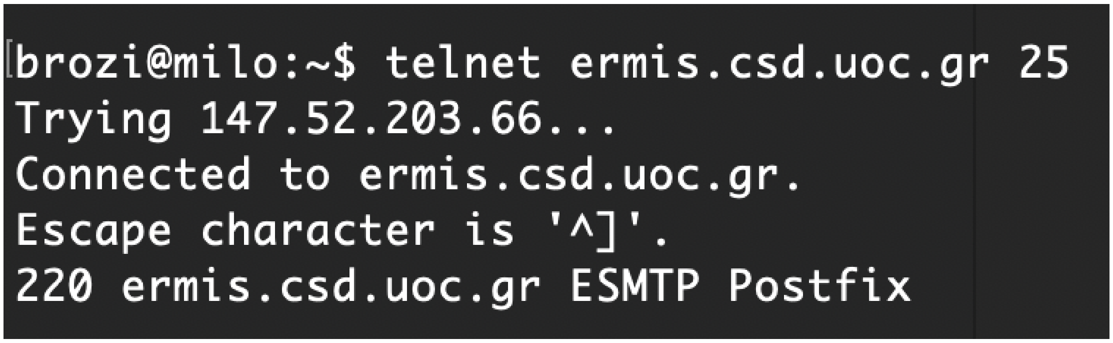
Replace `[flag]` with the appropriate `nslookup` type (e.g. A, NS, SOA, MX). Briefly answer the following questions:

i) What does the previous command do?
The nslookup type=MX command returns mail server information for the domain csd.uoc.gr. The mail server of csd.uoc.gr is ermis.csd.uoc.gr.



```
brozi@milo:~$ nslookup -type=mx csd.uoc.gr
Server:         147.52.16.1
Address:        147.52.16.1#53

Non-authoritative answer:
csd.uoc.gr      mail exchanger = 10 ermis.csd.uoc.gr.

Authoritative answers can be found from:
```

ii) What port does SMTP use ? Does SMTP use TCP or UDP and why?
SMT uses port 25. SMTP uses TCP because it provides reliable data transfer, unlike UDP.



```
[brozi@milo:~$ telnet ermis.csd.uoc.gr 25
Trying 147.52.203.66...
Connected to ermis.csd.uoc.gr.
Escape character is '^]'.
220 ermis.csd.uoc.gr ESMTP Postfix
```

Now let's send an email ! Start by running the following commands:

```
$ telnet [SMTP server name] [port]

$ HELO [csdXXXX@csd.uoc.gr]
```

Where `csdXXXX@csd.uoc.gr` is your email address (without the brackets "[ ]").

```
$ mail from: [csdXXXX@csd.uoc.gr]
$ rcpt to: [csdXXXX@csd.uoc.gr]
$ data
```

Now we will start typing the actual email we want to send. First we type in the subject.

```
$ subject: hy335a 2024 assignment 2
```

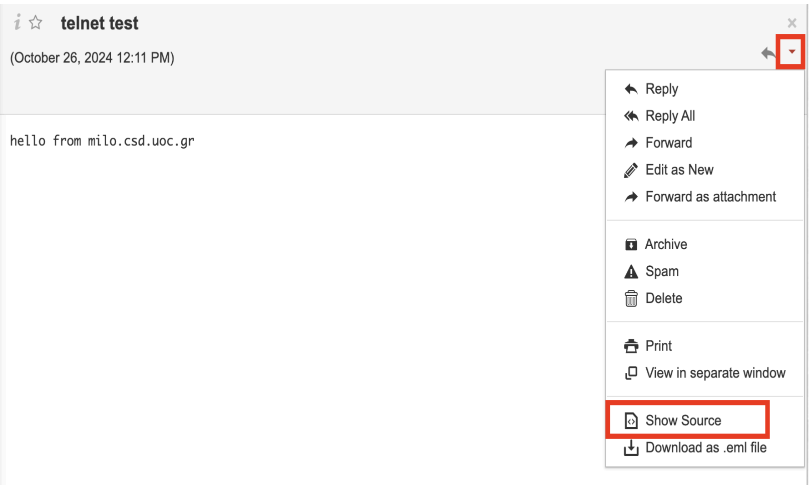
And now type the body of the email. You can write anything you like.

```
$ [your text goes here]
```

Now we are done with writing our email and want to send it to the recipient. How do we indicate that ? Run the appropriate command on `telnet`. After that you should receive a response of the form “`250 2.0.0 Ok: queued as 1C68D3C035F`”.

iii) Provide a screenshot of your terminal where all the commands from the previous steps and their outputs are visible.

iv) Log into `webmail.csd.uoc.gr` and find the email you just sent to yourself. Find the “source” of the email and provide a screenshot. An example on how to do that is shown below:



telnet test

(October 26, 2024 12:11 PM)

```
hello from milo.csd.uoc.gr
```