Prelab

HTTP Questions

1. [7 pts] Choose 5 HTTP status codes and describe each one.

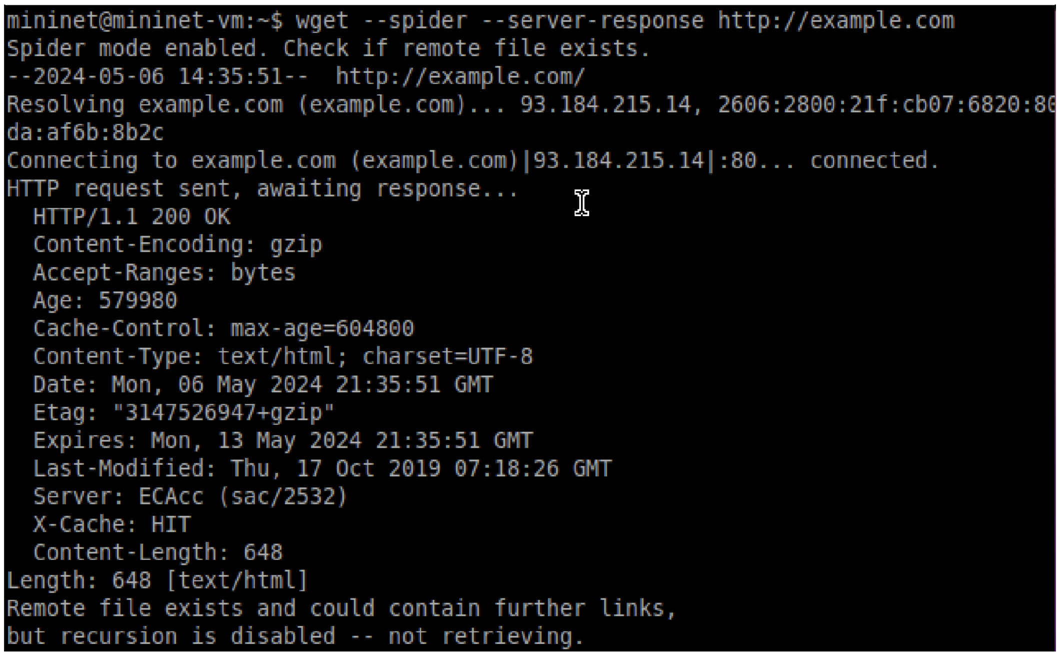
200 OK - This is the standard response for successful HTTP requests. It indicates that the request has succeeded. Server has successfully processed the request and is returning the requested information.
404 Not Found - Indicates that the server cannot find the requested resource. It is commonly used when a requested URL does not correspond to an existing resource on the server.
500 Internal Server Error - Indicates that the server encountered an unexpected condition that prevented it from fulfilling the request. It is a generic error message that could be caused by various issues on the server side, such as misconfigurations, programming errors, or resource exhaustion.
401 Unauthorized - Indicates that the request has not been applied because it lacks valid authentication credentials for the target resource. It is typically used when the client needs to authenticate itself before accessing the requested resource, but fails to do so or provides invalid credentials.
302 Found - Indicates that the requested resource has been temporarily moved to a different URL. It is often used for redirection purposes, instructing the client to make a new request to the provided URL. The client should repeat the request using the new URL provided in the response.

2. [7 pts] List the 8 HTTP 1.1 methods and explain what they do.

GET - Requests a representation of the specified resource. It retrieves data from the server without modifying anything on the server. This method is commonly used for fetching web pages, images, or other resources.
POST - Used to submit data to be processed to a specified resource. It is commonly used when submitting forms or uploading files to the server. Unlike GET, POST can modify data on the server and is not idempotent, meaning that multiple identical requests may have different effects.
HEAD - Similar to GET but only requests the headers of the response, not the actual content. It is useful for retrieving metadata about a resource without needing to download the entire content. This can be helpful for checking the validity or availability of a resource.
PUT - Used to upload a representation of a resource to the specified URI. It replaces the current representation of the target resource with the request payload. PUT is idempotent, meaning that multiple identical requests will have the same effect as a single request.
DELETE - Requests that the server deletes the specified resource. It is used to remove resources from the server. Like PUT, DELETE is idempotent, meaning that making multiple identical requests will have the same effect as a single request.
OPTIONS - Requests information about the communication options available for the target resource. It is used to determine which HTTP methods are supported by the server, or to request information about the supported methods and other capabilities.
TRACE - Echoes the received request so that the client can see what changes or additions have been made by intermediate servers. It is primarily used for diagnostic purposes to debug or troubleshoot connections.

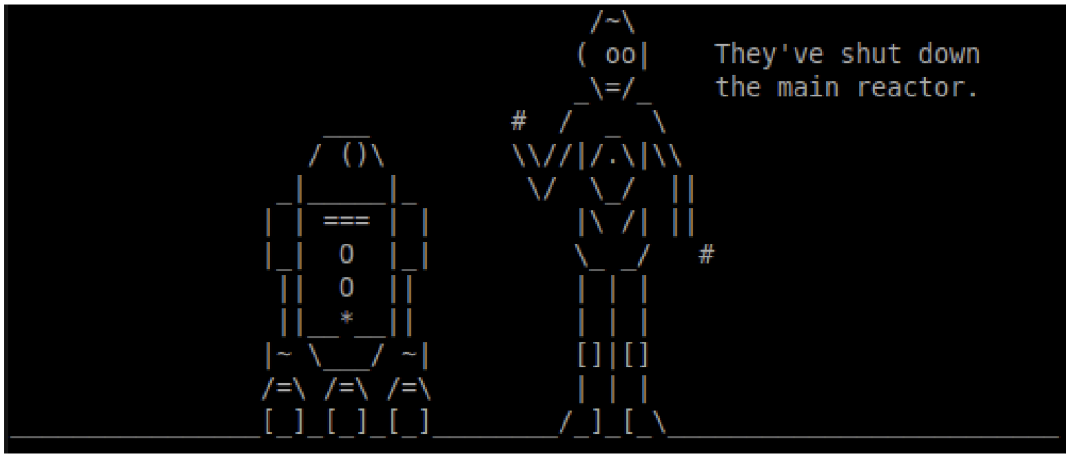
CONNECT - Establishes a tunnel to the server identified by the target resource. It is typically used to establish a secure SSL/TLS connection through an HTTP proxy. This method is often used in HTTPS communication.

3. [7 pts] Use wget on example.com to view the last modified date of the webpage. What was the HTTP return status given and what command was used to do this? (The command should not download the file! Hint: Look into the wget man page.)
   We ran 'wget –spider –server-response http://example.com' and got HTTP code 200 OK. The last modified date was Thu Oct 17, 2019.



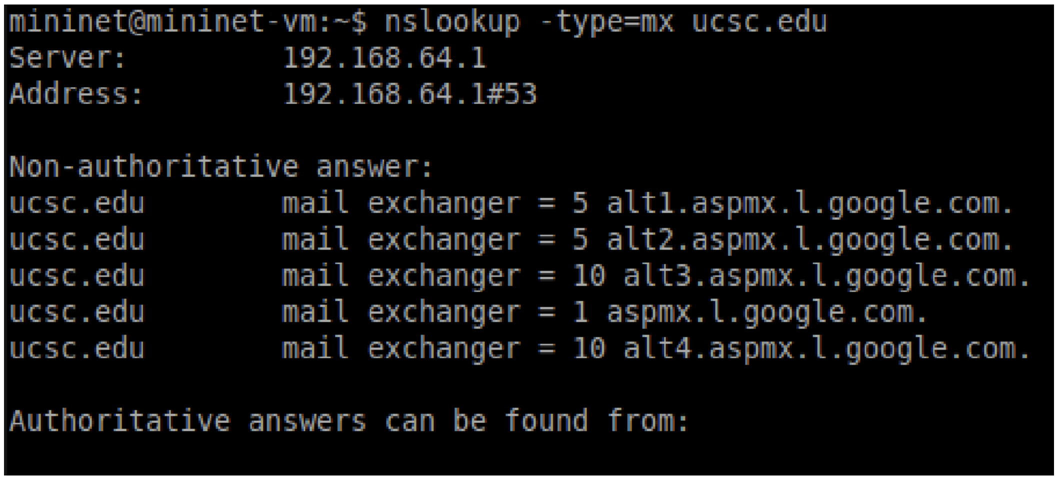
```
mininet@mininet-vm:~$ wget --spider --server-response http://example.com
Spider mode enabled. Check if remote file exists.
--2024-05-06 14:35:51--  http://example.com/
Resolving example.com (example.com)... 93.184.215.14, 2606:2800:21f:cb07:6820:80
da:af6b:8b2c
Connecting to example.com (example.com)|93.184.215.14|:80... connected.
HTTP request sent, awaiting response...
  HTTP/1.1 200 OK
  Content-Encoding: gzip
  Accept-Ranges: bytes
  Age: 579980
  Cache-Control: max-age=604800
  Content-Type: text/html; charset=UTF-8
  Date: Mon, 06 May 2024 21:35:51 GMT
  Etag: "3147526947+gzip"
  Expires: Mon, 13 May 2024 21:35:51 GMT
  Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT
  Server: ECAcc (sac/2532)
  X-Cache: HIT
  Content-Length: 648
Length: 648 [text/html]
Remote file exists and could contain further links,
but recursion is disabled -- not retrieving.
```

4. [7 pts] Look up the telnet command. Use telnet to connect to www.telehack.com, then type starwars. What does this telnet server do?
   It begins playing a Star Wars movie.



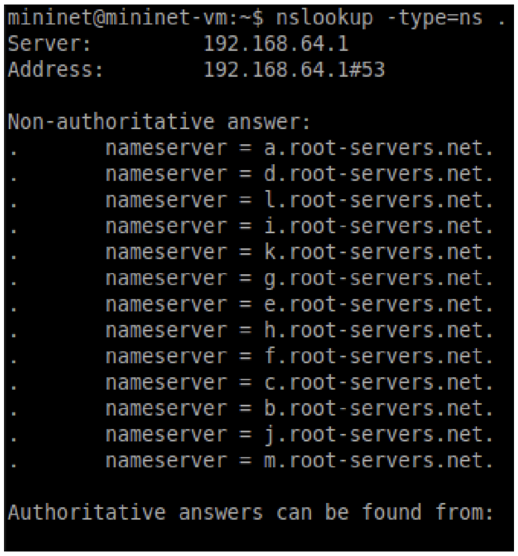
DNS Questions

5. [7 pts] In your own words describe what a DNS resource record (RR) is. Now using the command line tool nslookupfind the MX resource record of ucsc.edu. What does this resource record mean?
   A DNS resource record (RR) is a specific type of data entry that contains information associated with a specific domain name or hostname. An MX (Mail Exchange) resource record specifically indicates the mail servers responsible for handling email delivery for a domain. It specifies the priority and the domain name of the mail server. Mail servers listed in MX records are typically configured to accept and route email messages addressed to the domain.



```
mininet@mininet-vm:~$ nslookup -type=mx ucsc.edu
Server:         192.168.64.1
Address:        192.168.64.1#53

Non-authoritative answer:
ucsc.edu        mail exchanger = 5 alt1.aspmx.l.google.com.
ucsc.edu        mail exchanger = 5 alt2.aspmx.l.google.com.
ucsc.edu        mail exchanger = 10 alt3.aspmx.l.google.com.
ucsc.edu        mail exchanger = 1 aspmx.l.google.com.
ucsc.edu        mail exchanger = 10 alt4.aspmx.l.google.com.

Authoritative answers can be found from:
```

6. [7 pts] What does the command nslookup -type=ns . do? Explain its output. (Note: the . is part of the command!)
   The output will typically display the authoritative name servers for the root domain. These authoritative name servers are responsible for managing the DNS root zone and delegating authority for the top-level domains.



```
mininet@mininet-vm:~$ nslookup -type=ns .
Server:         192.168.64.1
Address:        192.168.64.1#53

Non-authoritative answer:
.       nameserver = a.root-servers.net.
.       nameserver = d.root-servers.net.
.       nameserver = l.root-servers.net.
.       nameserver = i.root-servers.net.
.       nameserver = k.root-servers.net.
.       nameserver = g.root-servers.net.
.       nameserver = e.root-servers.net.
.       nameserver = h.root-servers.net.
.       nameserver = f.root-servers.net.
.       nameserver = c.root-servers.net.
.       nameserver = b.root-servers.net.
.       nameserver = j.root-servers.net.
.       nameserver = m.root-servers.net.

Authoritative answers can be found from:
```

TCP Questions

7. [10 pts] How can multiple application services running on a single machine with a single IP address be uniquely identified?

Multiple application services running on a single machine with a single IP address can be uniquely identified using port numbers. Each network communication endpoint (socket) consists of an IP address and a port number. When a packet arrives at a machine with a destination IP address, the port number in the packet header helps the operating system direct the packet to the appropriate application or service. By assigning unique port numbers to each service, multiple services running on the same machine can coexist and be uniquely identified.

8. [9 pts] What is the purpose of the window mechanism in TCP?

The purpose of the window mechanism is to control the flow of data between the sender and receiver and to optimize the performance of data transmission over a network. The window mechanism allows the sender to dynamically adjust the amount of data it sends before waiting for an acknowledgment from the receiver. The sender sends multiple data segments without waiting for individual acknowledgments for each segment. The receiver advertises its receive window size to the sender, indicating the amount of data it can accept. The sender then adjusts its transmission rate based on the receiver's window size, ensuring efficient utilization of network resources.

9. [9 pts] What is an MTU? What happens when a packet is larger than the MTU?

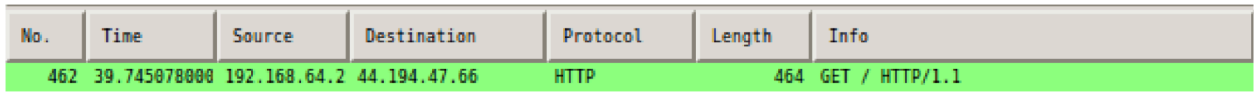
MTU, or Maximum Transmission Unit, is the largest packet size that can be sent over a network without fragmentation. When a packet exceeds the MTU, it gets fragmented into smaller packets for transmission. Excessive fragmentation can lead to performance issues or packet loss.

Lab

Part 1 - HTTP:

In this section, we will observe how the HTTP protocol operates. We will do this by using the Mininet VM. Begin by opening Wireshark and listening on the 'any' interface.
Open Chromium and navigate to http://httpbin.org

1. [10 pts] Find the HTTP packet that corresponds to the initial request that your computer made. Take a screenshot of this packet. What HTTP method did your computer use to make this request?

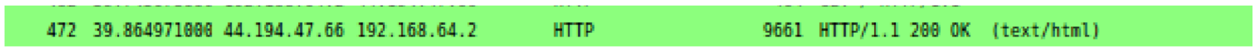


No.	Time	Source	Destination	Protocol	Length	Info
462	39.745078000	192.168.64.2	44.194.47.66	HTTP	464	GET / HTTP/1.1

GET was used.

2. [10 pts] Find the HTTP packet that corresponds to the initial response the server made to your request. Take a screenshot of this packet. What HTTP status code did the server

return? What is the content type of the response the server is sending back?

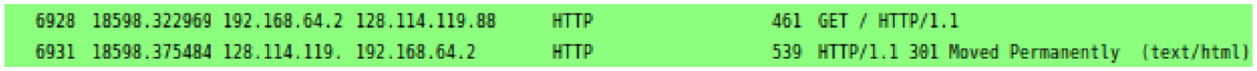


```
472  39.864971000 44.194.47.66 192.168.64.2       HTTP            9661  HTTP/1.1 200 OK  (text/html)
```

200 OK was the status code, and the server is sending html.

Using Chromium and navigate to http://ucsc.edu

3. [10 pts] Find the HTTP packets that correspond to the initial request and response that your computer made. Take a screenshot of these packets. What's different? Explain.

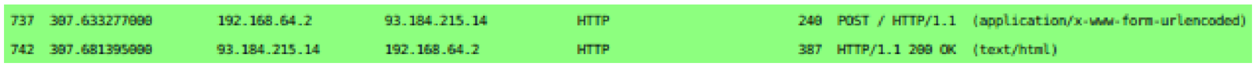


```
6928  18598.322969 192.168.64.2 128.114.119.88     HTTP            461  GET / HTTP/1.1
6931  18598.375484 128.114.119. 192.168.64.2       HTTP            539  HTTP/1.1 301 Moved Permanently  (text/html)
```

The difference is the 301 Moved Permanently return code. This indicates the resource has been permanently moved to a new location.

Using Chromium (or any other Linux utility you are comfortable with), find a way to make a HTTP packet with a method other than GET.

4. [10 pts] Take a screenshot of your packet, and explain what you did to create it.



```
737  307.633277000     192.168.64.2     93.184.215.14     HTTP          240  POST / HTTP/1.1  (application/x-www-form-urlencoded)
742  307.681395000     93.184.215.14    192.168.64.2      HTTP          387  HTTP/1.1 200 OK  (text/html)
```

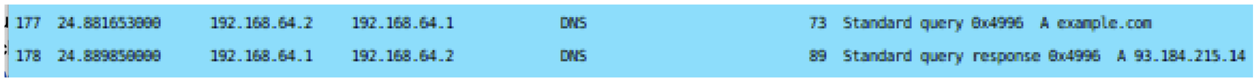
I used 'curl -X POST example.com -d "param1=value1&param2=value2"'

Part 2 - DNS:

In this section, we will observe how the DNS protocol operates. We will do this by using the Mininet VM. Begin by opening Wireshark and listening on the 'any' interface.

Open Chromium and navigate to www.example.com.

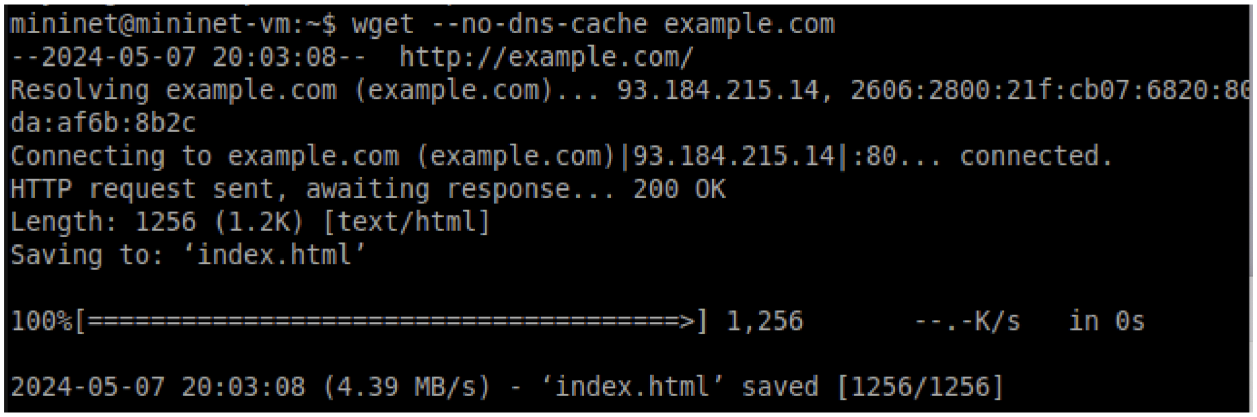
5. [10 pts] Were any steps taken by your computer before the web page was loaded? If so, using your captured packets in Wireshark, find the packets that allowed your computer to successfully load http://neverssl.com/ or http://www.example.com. Take a screenshot of these packets, and explain why you think these are the correct packets. What's the IP address of www.example.com?



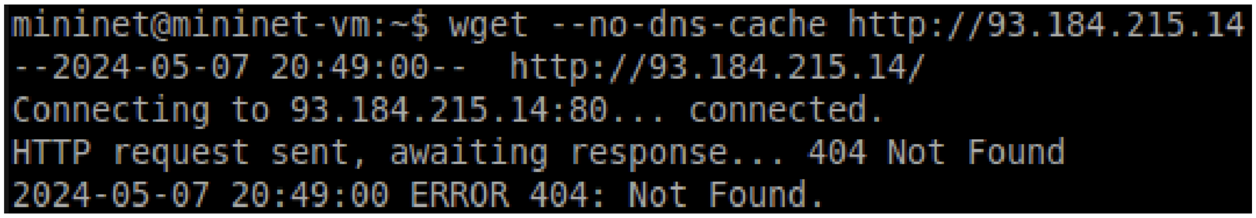
```
177  24.881653000     192.168.64.2     192.168.64.1       DNS          73  Standard query 0x4996  A example.com
178  24.889850000     192.168.64.1     192.168.64.2       DNS          89  Standard query response 0x4996  A 93.184.215.14
```

We have a standard query and a response to it to open example.com

6. [10 pts] Open a terminal window. Execute the command to flush your DNS cache: sudo /etc/init.d/networking restart (if your VM doesn't allow you to do so, you can skip this step and add "--no-dns-cache" flag for the next step when using "wget") Using wget, download the same content of www.example.com with its IP address you discovered in question 5, without sending DNS requests. What command did you use to accomplish that? Take a screenshot of related packets and explain why you think these are the correct packets.

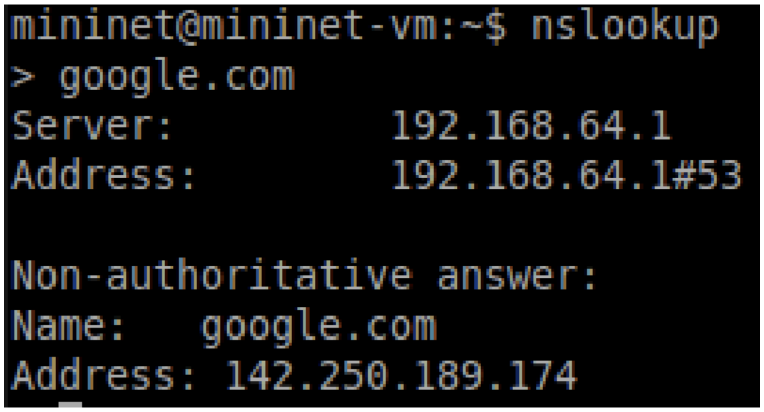


```
mininet@mininet-vm:~$ wget --no-dns-cache example.com
--2024-05-07 20:03:08--  http://example.com/
Resolving example.com (example.com)... 93.184.215.14, 2606:2800:21f:cb07:6820:80
da:af6b:8b2c
Connecting to example.com (example.com)|93.184.215.14|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1256 (1.2K) [text/html]
Saving to: ‘index.html’

100%[=======================================>] 1,256       --.-K/s   in 0s

2024-05-07 20:03:08 (4.39 MB/s) - ‘index.html’ saved [1256/1256]
```



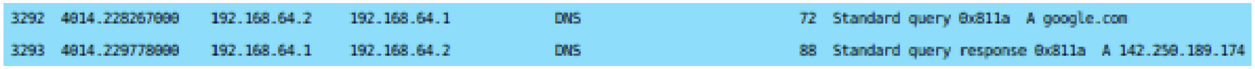
```
mininet@mininet-vm:~$ wget --no-dns-cache http://93.184.215.14
--2024-05-07 20:49:00--  http://93.184.215.14/
Connecting to 93.184.215.14:80... connected.
HTTP request sent, awaiting response... 404 Not Found
2024-05-07 20:49:00 ERROR 404: Not Found.
```

Open a terminal window. Using nslookup, find the A records for www.google.com.(If you can’t access Google, for example, you are in China, you could replace the domain name with www.baidu.com)



```
mininet@mininet-vm:~$ nslookup
> google.com
Server:         192.168.64.1
Address:        192.168.64.1#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.189.174
```

7. [10 pts] Take a screenshot of the packets corresponding to your request, and the response from the server. If the request was resolved, what is the IP address you were given for www.google.com?



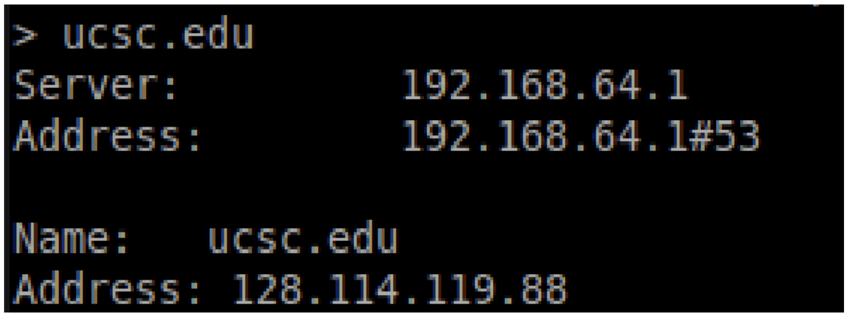
```
3292  4014.228267000   192.168.64.2   192.168.64.1    DNS    72  Standard query 0x811a  A google.com
3293  4014.229778000   192.168.64.1   192.168.64.2    DNS    88  Standard query response 0x811a  A 142.250.189.174
```

IP is 142.250.189.74.

8. [10 pts] Did your computer want to complete the request recursively? How do you know? Take a screenshot proving your answer.

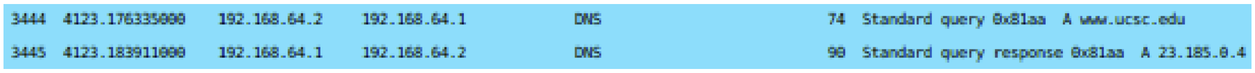
Given that there are DNS queries as google.com is loaded in, that indicates that the IP is being resolved recursively.

Using nslookup, find the A records for ucsc.edu.



```
> ucsc.edu
Server:         192.168.64.1
Address:        192.168.64.1#53

Name:   ucsc.edu
Address: 128.114.119.88
```

9. [10 pts] Take a screenshot of the packets corresponding to your request, and the response from the server. If the request was resolved, what is the IP address you were given for ucsc.edu?



```
3444  4123.176335000   192.168.64.2   192.168.64.1   DNS   74  Standard query 0x81aa  A www.ucsc.edu
3445  4123.183911000   192.168.64.1   192.168.64.2   DNS   90  Standard query response 0x81aa  A 23.185.0.4
```

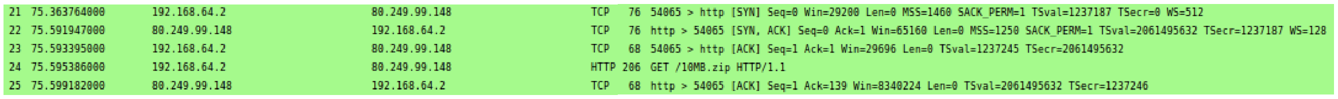
IP is 128.114.119.88.

10. [10 pts] What is the authoritative name server for the ucsc.edu domain? How do you know? Take a screenshot proving your answer.
The authoritative is ucsc.edu at address 128.114.119.88.

Part 3 - TCP:

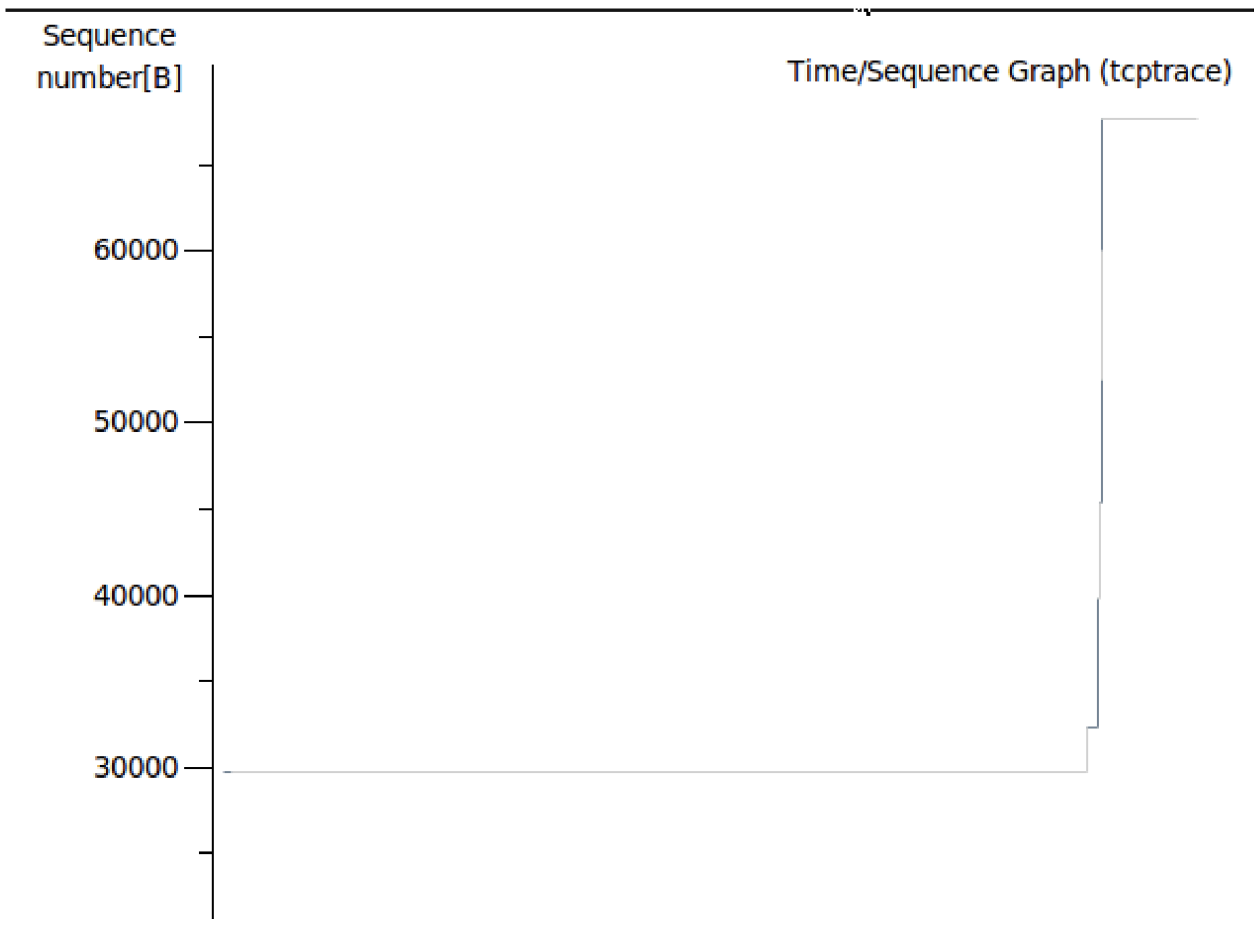
In this section, we will observe how the TCP protocol operates. We will do this by using the Mininet VM. Begin by opening Wireshark and listening on the 'any' interface. Open a terminal window. Using wget, download the file http://ipv4.download.thinkbroadband.com/10MB.zip

11. [10 pts] Find the packets corresponding with the SYN, SYN-ACK, and ACK that initiated the TCP connection for this file transfer. Take a screenshot of these packets. What was the initial window size that your computer advertised to the server? What was the initial window size that the server advertised to you?



```
21  75.363764000   192.168.64.2    80.249.99.148   TCP   76  54065 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1237187 TSecr=0 WS=512
22  75.591947000   80.249.99.148   192.168.64.2    TCP   76  http > 54065 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1250 SACK_PERM=1 TSval=2061495632 TSecr=1237187 WS=128
23  75.593395000   192.168.64.2    80.249.99.148   TCP   68  54065 > http [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=1237245 TSecr=2061495632
24  75.595386000   192.168.64.2    80.249.99.148   HTTP 206  GET /10MB.zip HTTP/1.1
25  75.599182000   80.249.99.148   192.168.64.2    TCP   68  http > 54065 [ACK] Seq=1 Ack=139 Win=8340224 Len=0 TSval=2061495632 TSecr=1237246
```

12. [10 pts] Find a packet from the download with a source of the server and a destination of your computer. Create a tcptrace graph with this packet selecrated. Take a screenshot of the graph and explain what it is showing. Look into theWireshark documentation if you need assistance making this graph. In the next section, we will be simulating loss, the command tc qdisc will be needed. When you first use the command you should use add dev for the device you plan on changing. It only needs to be set on the sender's side. After adding the device use change dev.

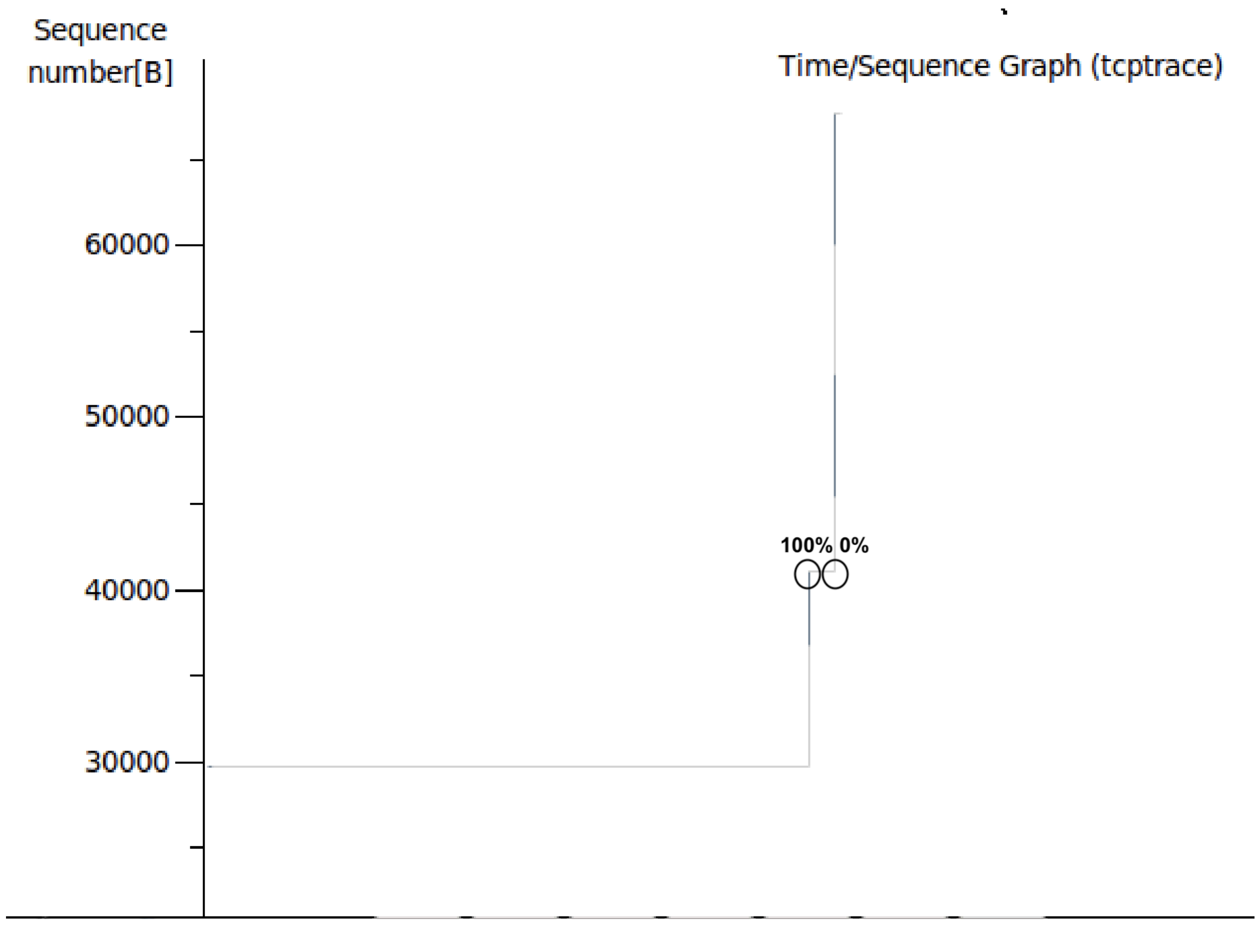


Example:

```
sudo tc qdisc add dev eth0 root netem loss 0%
sudo tc qdisc change dev eth0 root netem loss 100%
```

Read through the following paragraph before starting the next step. Open 2 terminals and have the commands typed and ready before you begin. In one terminal, download the 10MB.zip file again. While the download is in progress, change loss to 100%. After a few seconds, change loss to 0%.

13. [10 pts] Find a packet from the download with a source of the server and a destination of your computer. Create a tcptrace graph with this packet selected. Take a screenshot of the graph and explain what it is showing. Using an image editing program, circle the areas where the 0% loss is shown, as well as where TCP is in slow-start and congestion-avoidance.


Sequence number[B]
Time/Sequence Graph (tcptrace)
60000
50000
40000
30000
100%
0%