Prelab 2 - HTTP, DNS, and TCP:

Suggested Resources:

https://www.ietf.org/rfc/rfc2616.txt
https://www.ietf.org/rfc/rfc1035.txt https://linux.die.net/man/
http://www.tcpipguide.com/free/

HTTP Questions

1. **[7 pts] Choose 5 HTTP status codes and describe each one.**
   a. 404- Not found, the server has not found anything that matches the request URL
   b. 304- Not modified, when user perforces a GET request but the document has not been modified, code 304 should be
   c. 200- Success OK, the request from the user has succeeded, responses are different depending on the request
   d. 400 – Bad request, general error when bad syntax meaning request could not understood by server
   e. 500 – server has an unexpected condition meaning it could not complete the request from the user
2. **[7 pts] List the 8 HTTP 1.1 methods and explain what they do.**
   a. OPTIONS- is used to find communication options(ie, HTTP methods) available which are supported by a webserver
   b. GET- I used to retrieve data from a webserver given a URI, this request should only get data not modify it
   c. HEAD- similar to GET by the server will only respond with the response line and headers
   d. POST- is used to send data to the server such as a file
   e. PUT – is used to store the file at the specified URL
   f. DELETE – is used to delete a file at the specified URL
   g. TRACE – allows user to see what the other end-user is receiving, this will echo the contents of an HTTP request back to the user
   h. CONNECT- this establishes a tunnel/connection to a webserver

wget and *telnet* are two commonly known command line tools for testing and debugging. Answer the following questions by using your Mininet VM's terminal or the Unix timeshare (see Lab 1 for instructions on connecting to the timeshare).

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.)

```
-bash-4.2$ wget --server-response --spider http://www.example.com/
Spider mode enabled. Check if remote file exists.
--2019-10-16 18:22:46--  http://www.example.com/
Resolving www.example.com (www.example.com)... 93.184.216.34, 2606:2800:220:1:248:1893:25c8:1946
Connecting to www.example.com (www.example.com)|93.184.216.34|:80... connected.
HTTP request sent, awaiting response...
  HTTP/1.1 200 OK
  Content-Encoding: gzip
  Accept-Ranges: bytes
  Cache-Control: max-age=604800
  Content-Type: text/html; charset=UTF-8
  Date: Thu, 17 Oct 2019 01:22:49 GMT
  Etag: "1541025663"
  Expires: Thu, 24 Oct 2019 01:22:49 GMT
  Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
  Server: ECS (oxr/831E)
  X-Cache: HIT
  Content-Length: 606
Length: 606 [text/html]
Remote file exists and could contain further links,
but recursion is disabled -- not retrieving.
```

HTTP return status is 200 OK

Command is: wget --server-response --spider example.com

4. [7 pts] **Look up the *telnet* command. Use *telnet* to connect to *towel.blinkenlights.nl*. What does this telnet server do?**
This telnet server connects and opens a terminal that will play stars wars using ascii characters

DNS Questions

5. [7 pts] **In your own words describe what a DNS resource record (RR) is. Now using the command line tool *nslookup* find the *MX* resource record of *ucsc.edu*. What does this resource record mean?**
A DNS resource record determines what type of record will be used. Different types include:

- Start of authority (SOA)
- Name server (NS)
- Pointer record (PTR)
- Address (A)
- IPv6 Address (AAAA)
- Mail exchange (MX)
- Canonical name (CNAME)
- Windows Internet Naming Service (WINS)
- WINS Reverse Look up (WINSR)

This resource record shows the mail exchange records of ucsc.edu server

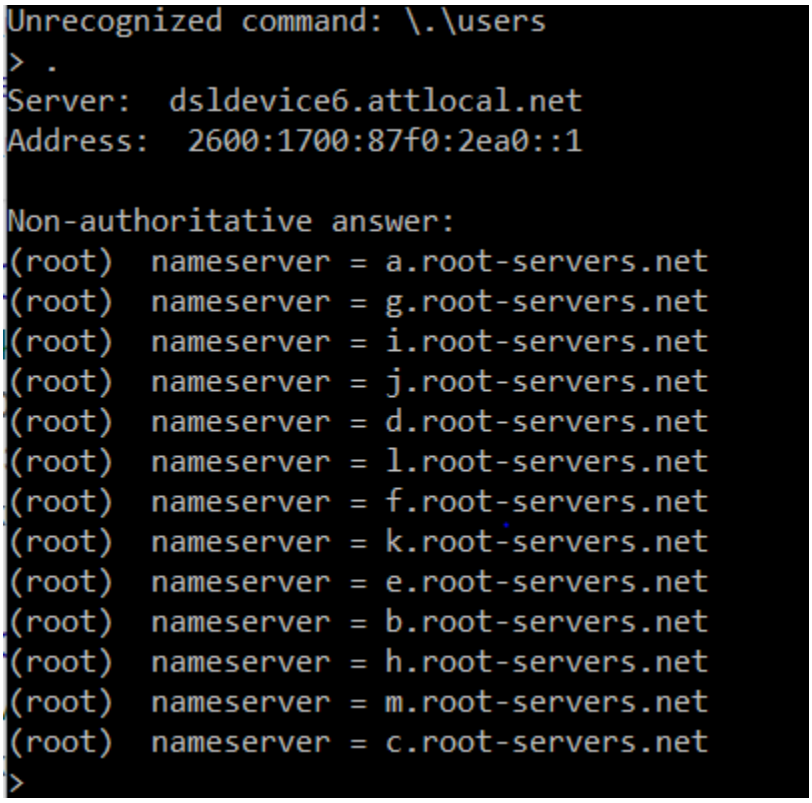
```
> ucsc.edu
Server:  router.asus.com
Address:  192.168.1.1

Non-authoritative answer:
Name:    ucsc.edu
Address:  128.114.109.5

> set q=mx
> ucsc.edu
Server:  router.asus.com
Address:  192.168.1.1

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

alt4.aspmx.l.google.com AAAA IPv6 address = 2607:f8b0:400d:c0b::1b
alt1.aspmx.l.google.com internet address = 209.85.146.26
aspmx.l.google.com      internet address = 74.125.195.26
alt2.aspmx.l.google.com internet address = 172.253.112.26
alt2.aspmx.l.google.com AAAA IPv6 address = 2607:f8b0:4023::1a
>
```

The mx records shows the ucsc.edu mail exchange servers

6. **[7 pts] What does the command *nslookup -type=ns .* do*?* Explain its output. (Note:**

   the . is part of the command!)

   This command displays the query of the NS(Name Server) Record of the current device



TCP Questions

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

   Different application services running on a single manchiene with a single IP address can be uniquely identified by using different port number/addresss

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

   The window mechanism in TCP is when the send can send one or more data segments and the receiver will acknowledge all messages sent within the window size. The purpose of this is to control that flow of packets between sender and receiver.

9. **[9 pts] What is an MTU? What happens when a packet is larger than the MTU?**
   A MTU is a maximum transmission unit which is the biggest packet that be sent in a transmission.
   Depending on the interface and the Don't Fragment(DF) bit, the router will fragment the packet into smaller units which can cause a delay.

Lab 2 - HTTP, DNS, and TCP:

Suggested Resources:

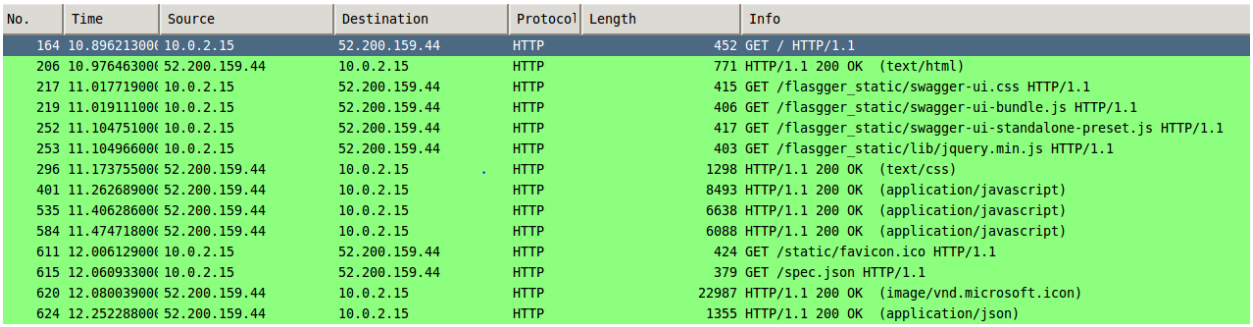
http://packetbomb.com/understanding-the-tcptrace-time-sequence-graph-in-wireshark/
https://wiki.linuxfoundation.org/networking/netem

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

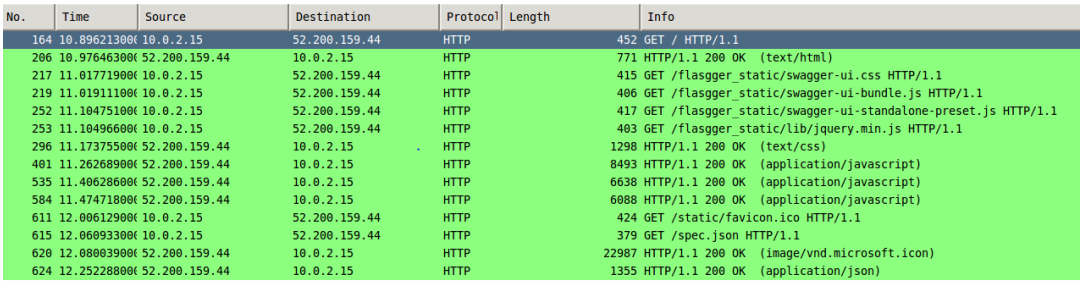
1. [10 pts] **Find the HTTP packet that corresponds to the initial request that your computer made. Take a screenshot of this packet. What HTTP method did your computer use to make this request?**



No.	Time	Source	Destination	Protocol	Length	Info
164	10.89621300(	10.0.2.15	52.200.159.44	HTTP	452	GET / HTTP/1.1
206	10.97646300(	52.200.159.44	10.0.2.15	HTTP	771	HTTP/1.1 200 OK (text/html)
217	11.01771900(	10.0.2.15	52.200.159.44	HTTP	415	GET /flasgger_static/swagger-ui.css HTTP/1.1
219	11.01911100(	10.0.2.15	52.200.159.44	HTTP	406	GET /flasgger_static/swagger-ui-bundle.js HTTP/1.1
252	11.10475100(	10.0.2.15	52.200.159.44	HTTP	417	GET /flasgger_static/swagger-ui-standalone-preset.js HTTP/1.1
253	11.10496600(	10.0.2.15	52.200.159.44	HTTP	403	GET /flasgger_static/lib/jquery.min.js HTTP/1.1
296	11.17375500(	52.200.159.44	10.0.2.15	HTTP	1298	HTTP/1.1 200 OK (text/css)
401	11.26268900(	52.200.159.44	10.0.2.15	HTTP	8493	HTTP/1.1 200 OK (application/javascript)
535	11.40628600(	52.200.159.44	10.0.2.15	HTTP	6638	HTTP/1.1 200 OK (application/javascript)
584	11.47471800(	52.200.159.44	10.0.2.15	HTTP	6088	HTTP/1.1 200 OK (application/javascript)
611	12.00612900(	10.0.2.15	52.200.159.44	HTTP	424	GET /static/favicon.ico HTTP/1.1
615	12.06093300(	10.0.2.15	52.200.159.44	HTTP	379	GET /spec.json HTTP/1.1
620	12.08003900(	52.200.159.44	10.0.2.15	HTTP	22987	HTTP/1.1 200 OK (image/vnd.microsoft.icon)
624	12.25228800(	52.200.159.44	10.0.2.15	HTTP	1355	HTTP/1.1 200 OK (application/json)

The HTTP method used to make the request was the GET method in frame 164

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?**

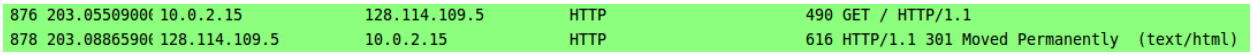


No.	Time	Source	Destination	Protocol	Length	Info
164	10.89621300(	10.0.2.15	52.200.159.44	HTTP	452	GET / HTTP/1.1
206	10.97646300(	52.200.159.44	10.0.2.15	HTTP	771	HTTP/1.1 200 OK (text/html)
217	11.01771900(	10.0.2.15	52.200.159.44	HTTP	415	GET /flasgger_static/swagger-ui.css HTTP/1.1
219	11.01911100(	10.0.2.15	52.200.159.44	HTTP	406	GET /flasgger_static/swagger-ui-bundle.js HTTP/1.1
252	11.10475100(	10.0.2.15	52.200.159.44	HTTP	417	GET /flasgger_static/swagger-ui-standalone-preset.js HTTP/1.1
253	11.10496600(	10.0.2.15	52.200.159.44	HTTP	403	GET /flasgger_static/lib/jquery.min.js HTTP/1.1
296	11.17375500(	52.200.159.44	10.0.2.15	HTTP	1298	HTTP/1.1 200 OK (text/css)
401	11.26268900(	52.200.159.44	10.0.2.15	HTTP	8493	HTTP/1.1 200 OK (application/javascript)
535	11.40628600(	52.200.159.44	10.0.2.15	HTTP	6638	HTTP/1.1 200 OK (application/javascript)
584	11.47471800(	52.200.159.44	10.0.2.15	HTTP	6088	HTTP/1.1 200 OK (application/javascript)
611	12.00612900(	10.0.2.15	52.200.159.44	HTTP	424	GET /static/favicon.ico HTTP/1.1
615	12.06093300(	10.0.2.15	52.200.159.44	HTTP	379	GET /spec.json HTTP/1.1
620	12.08003900(	52.200.159.44	10.0.2.15	HTTP	22987	HTTP/1.1 200 OK (image/vnd.microsoft.icon)
624	12.25228800(	52.200.159.44	10.0.2.15	HTTP	1355	HTTP/1.1 200 OK (application/json)

The Status code returned is 200 OK and the content type of the response is (text/html)

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

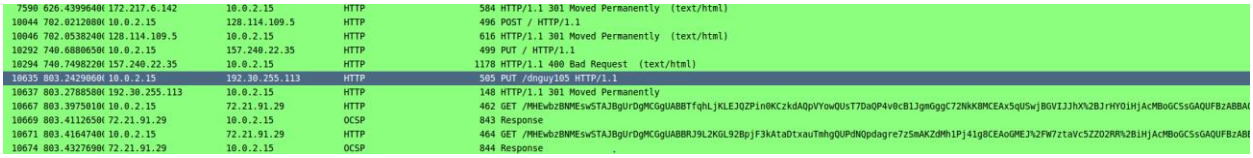
3. [**10 pts] Find the HTTP packets that correspond to the initial request and response that your computer made. Take a screenshot of these packets. What's different? Explain.**



The difference between this request and the previous is that status code 301 was returned meaning the requested resource has been assignment a new URI.

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.**



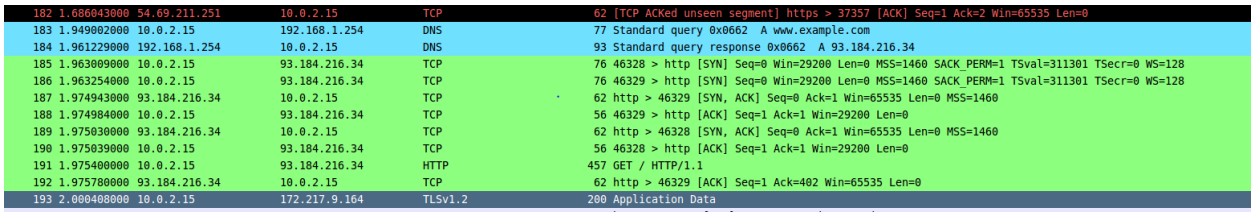
To create this PUT method, I downloaded the Postman extension in chromium which allows users to send different types of HTTP request to certain URLS. When using the application, there is a button that will let you select which request to choose from and a text box to enter in a url for the request to be sent to.

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://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?**



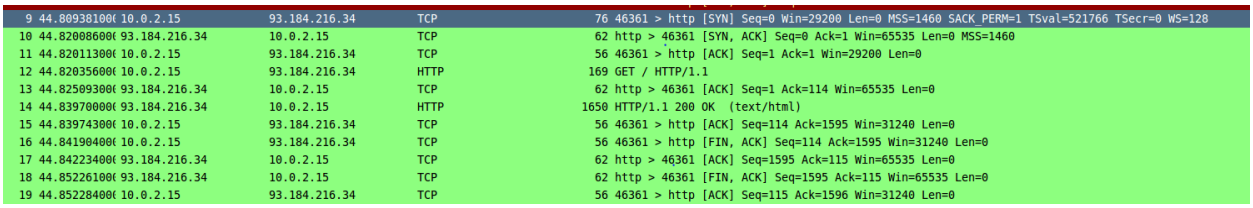
182 1.686043000	54.69.211.251	10.0.2.15	TCP	62 [TCP ACKed unseen segment] https > 37357 [ACK] Seq=1 Ack=2 Win=65535 Len=0
183 1.949002000	10.0.2.15	192.168.1.254	DNS	77 Standard query 0x0662 A www.example.com
184 1.961229000	192.168.1.254	10.0.2.15	DNS	93 Standard query response 0x0662 A 93.184.216.34
185 1.963009000	10.0.2.15	93.184.216.34	TCP	76 46328 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=311301 TSecr=0 WS=128
186 1.963254000	10.0.2.15	93.184.216.34	TCP	76 46329 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=311301 TSecr=0 WS=128
187 1.974943000	93.184.216.34	10.0.2.15	TCP	62 http > 46329 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
188 1.974984000	10.0.2.15	93.184.216.34	TCP	56 46329 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
189 1.975030000	93.184.216.34	10.0.2.15	TCP	62 http > 46328 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
190 1.975039000	10.0.2.15	93.184.216.34	TCP	56 46328 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
191 1.975400000	10.0.2.15	93.184.216.34	HTTP	457 GET / HTTP/1.1
192 1.975780000	93.184.216.34	10.0.2.15	TCP	62 http > 46329 [ACK] Seq=1 Ack=402 Win=65535 Len=0
193 2.000408000	10.0.2.15	172.217.9.164	TLSv1.2	200 Application Data

Steps taken before the webpage was loaded is done by the a DNS server because the computer ask for the IP address of www.example.com. The server then responses with numerical IPv4 address of the website which the computer then uses TCP to access.

The IP addr for www.example.com is 93.184.216.34

6. [10 pts**] Open a terminal window. Execute the command to flush your DNS cache: sudo /etc/init.d/networking restart**
   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.



9 44.80938100(	10.0.2.15	93.184.216.34	TCP	76 46361 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=521766 TSecr=0 WS=128
10 44.82008600(	93.184.216.34	10.0.2.15	TCP	62 http > 46361 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
11 44.82011300(	10.0.2.15	93.184.216.34	TCP	56 46361 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
12 44.82035600(	10.0.2.15	93.184.216.34	HTTP	169 GET / HTTP/1.1
13 44.82509300(	93.184.216.34	10.0.2.15	TCP	62 http > 46361 [ACK] Seq=1 Ack=114 Win=65535 Len=0
14 44.83970000(	93.184.216.34	10.0.2.15	HTTP	1650 HTTP/1.1 200 OK (text/html)
15 44.83974300(	10.0.2.15	93.184.216.34	TCP	56 46361 > http [ACK] Seq=114 Ack=1595 Win=31240 Len=0
16 44.84190400(	10.0.2.15	93.184.216.34	TCP	56 46361 > http [FIN, ACK] Seq=114 Ack=1595 Win=31240 Len=0
17 44.84223400(	93.184.216.34	10.0.2.15	TCP	62 http > 46361 [ACK] Seq=1595 Ack=115 Win=65535 Len=0
18 44.85226100(	93.184.216.34	10.0.2.15	TCP	62 http > 46361 [FIN, ACK] Seq=1595 Ack=115 Win=65535 Len=0
19 44.85228400(	10.0.2.15	93.184.216.34	TCP	56 46361 > http [ACK] Seq=115 Ack=1596 Win=31240 Len=0

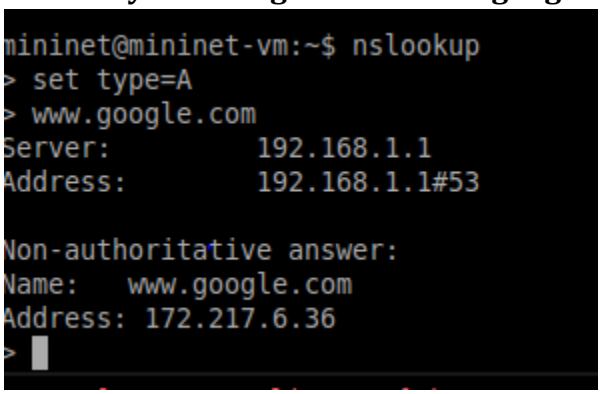
Sudo wget --header host:www.example.com 93.184.216.34

These are the correct packets because there are no packets using the DNS protocol meaning the computer does not need to translate www.example.com to 93.184.216.34. The command uses wget and specfies the host incase the IP address is hosting different sites

Open a terminal window. Using nslookup, find the A records for www.google.com.

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?



3719	1206.270110(	10.0.2.15	192.168.1.1	DNS	76 Standard query 0xd645 A www.google.com
3720	1206.272884(	192.168.1.1	10.0.2.15	DNS	92 Standard query response 0xd645 A 172.217.6.36

The IP addr given for www.google.com is 172.217.6.36

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

   This complete the request recursively. This means instead of the original server asking the primary server for the IP address and having the primary server respond to the original server, which in turn responds to the client.
   iteratively means the server asks the primary server for the addresses which then sends the client the IP addresses.
   This can be seen because the destination of the first packet and the source of the second packet are the same

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

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?**

35	1621.005338(	192.168.1.254	10.0.2.15	DNS	143 Standard query response 0x6c6f No such name
36	1631.646184(	10.0.2.15	192.168.1.254	DNS	70 Standard query 0x1ca9 A ucsc.edu
37	1631.676823(	192.168.1.254	10.0.2.15	DNS	86 Standard query response 0x1ca9 A 128.114.109.5
38	1667.062940(	10.0.2.15	192.168.1.254	DNS	70 Standard query 0xaeed A ucsc.edu
39	1667.093344(	192.168.1.254	10.0.2.15	DNS	86 Standard query response 0xaeed A 128.114.109.5
40	1672.069955(	CadmusCo_27:c6:3a		ARP	44 Who has 10.0.2.2? Tell 10.0.2.15
41	1672.070344(	RealtekU_12:35:02		ARP	62 10.0.2.2 is at 52:54:00:12:35:02

   IP addr is 128.114.109.5

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

```
Authoritative answers can be found from:
> set type=soa
> ucsc.edu
Server:         192.168.1.254
Address:        192.168.1.254#53

Non-authoritative answer:
ucsc.edu
        origin = adns1.ucsc.edu
        mail addr = hostmaster.ucsc.edu
        serial = 16566025
        refresh = 10800
        retry = 900
        expire = 2419200
        minimum = 900

Authoritative answers can be found from:
>
```

The authoritative name server for ucsc.edu domain is adns1.ucsc.edu, this is explained in the link below. Setting the type to SOA(Start of Authority) then displays origin which is the primary server.

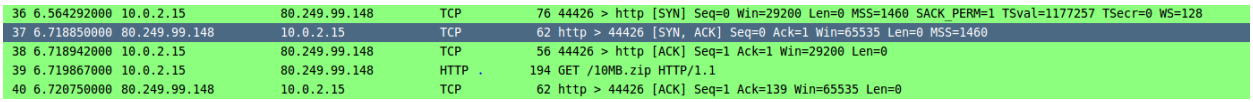
https://stackoverflow.com/questions/38021/how-do-i-find-the-authoritative-name-server-for-a-domain-name

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?**

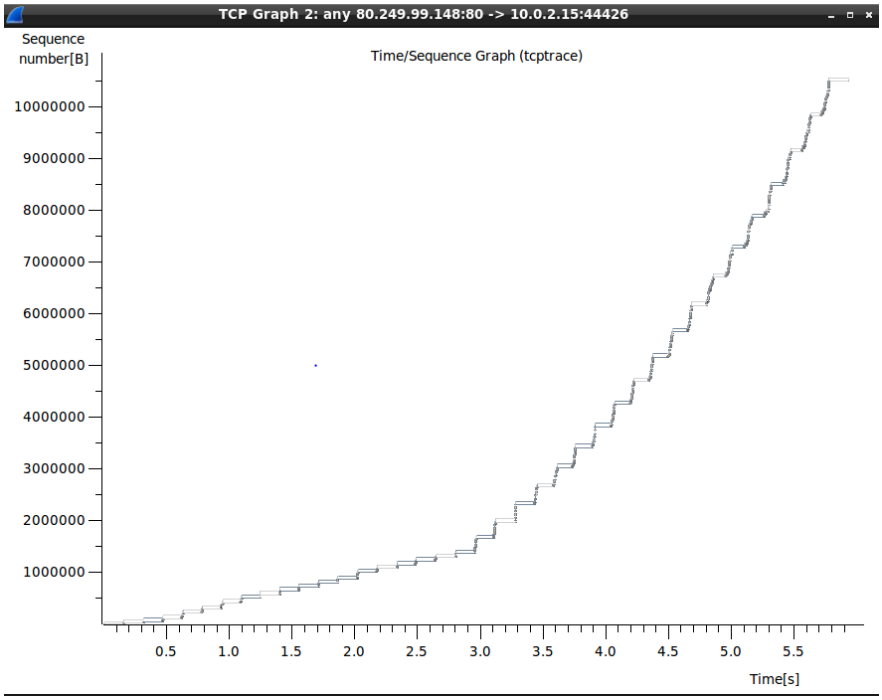


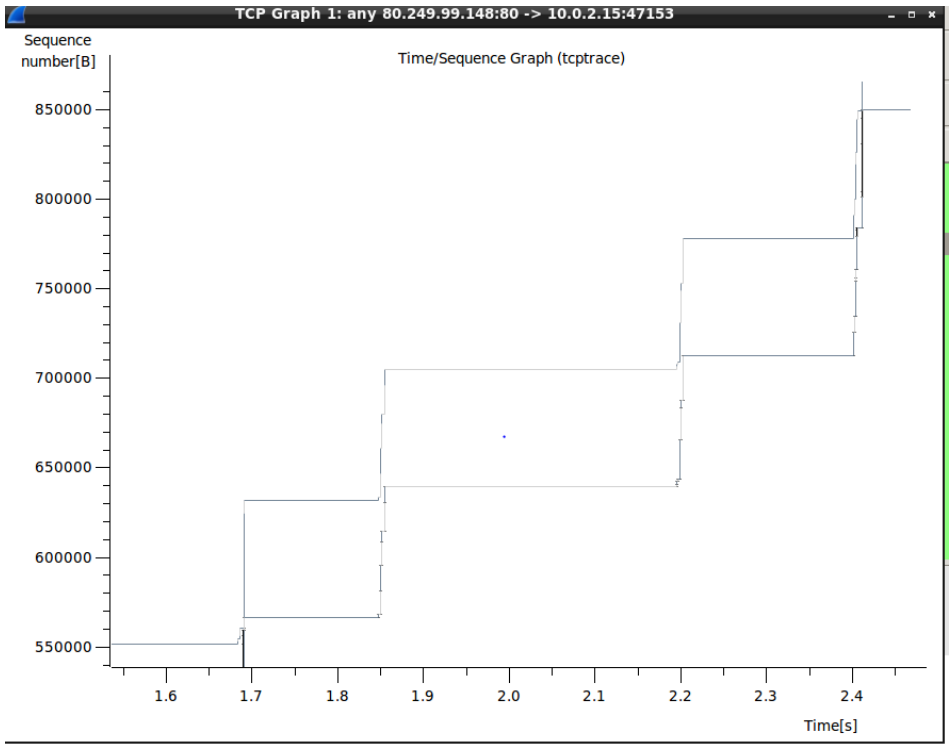
Window size advertised to server = 29200 segments

Window size advertised to you = 65535 segments

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**

selected. Take a screenshot of the graph and explain what it is showing. Look into the Wireshark documentation if you need assistance making this graph.





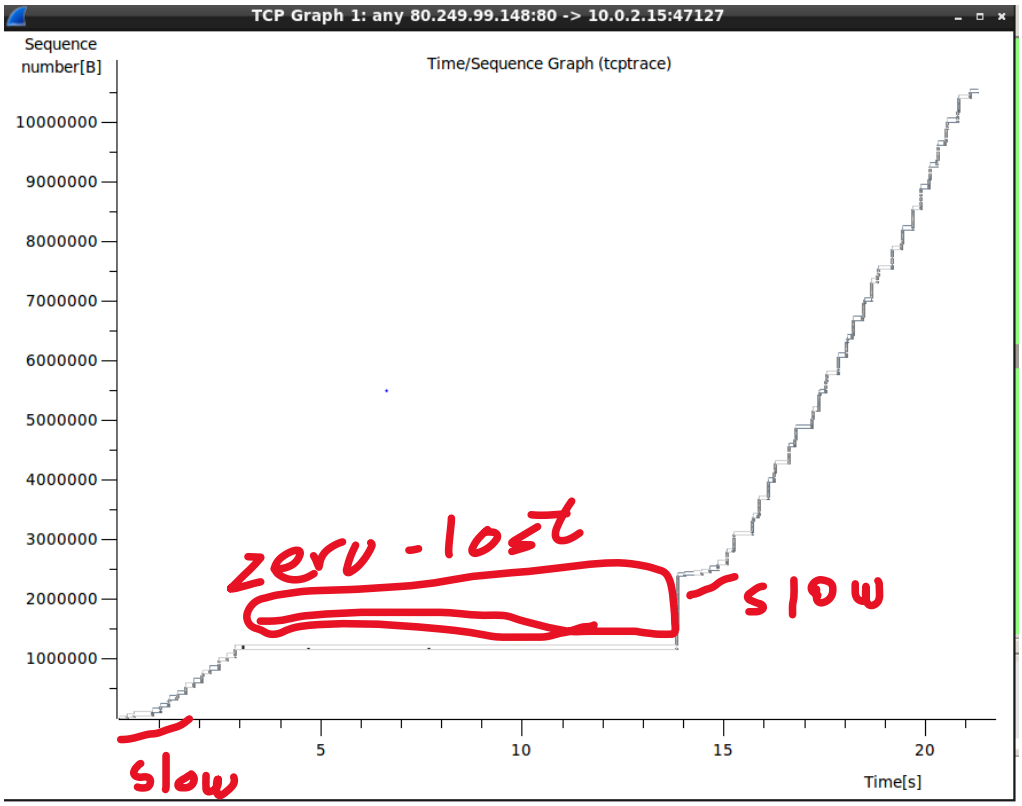
On the X-axis time is shown, and the Y-axis is the TCP sequence number.
The upper lighter line shows the receive window while the lower line represents the sequence number for the data.

The graph also grows exponentially because of the slow start rule. This is to not saturate the bandwidth.
To see the slow start, the graph will have an exponential shape and turn into a linear line

In the next section, we will be simulating loss, the command *tc qdisc* will be needed. When you first use the command, xyou 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*.

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 editting program, circle the areas where the 0% loss is shown, as well as where TCP is in slow-start and congestion-avoidance.**



In this graph, the slow start can be seen as normal but then stops at the first lost

event. After the lost event, the slow start will begin again as seen around 14-15 seconds. The graph then grows linearly after when it reaches the bandwidth.

https://netbeez.net/blog/how-to-use-the-linux-traffic-control/