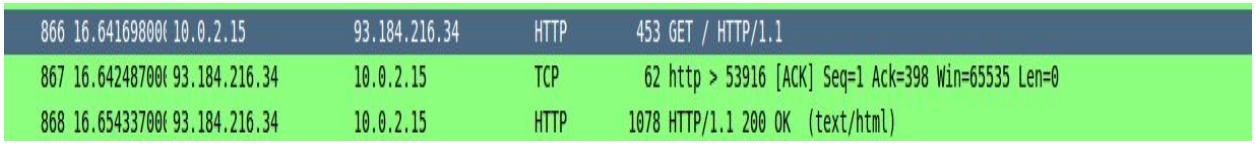
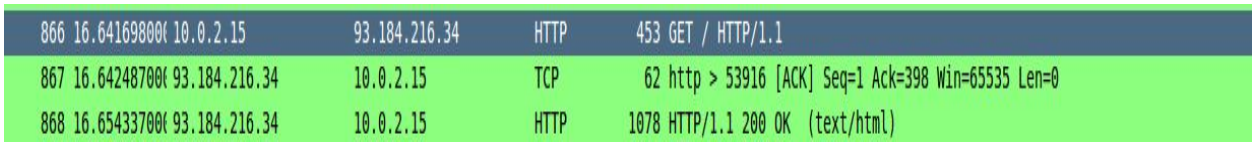
Part 1: HTTP In this section, we will observe how the HTTP protocol operates. We will do this by using the Mininet VM. Begin by opening Wireshark and listening on the ‘any’ interface. Open Chromium and navigate to http://www.example.com (not https!):

1. (5) Find the packet that corresponds to the initial HTTP request that your computer issued. Take a screenshot of this packet. What HTTP method did your computer use to make this request? What URI did your computer request from the server, as present in the HTTP request? (note: NOT the URL). Explain.



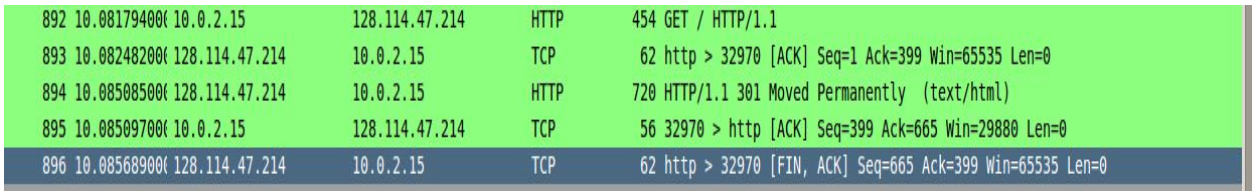
The http method used was get and the http/1.1 was requested.

2. (5) Find the packet that corresponds to the initial HTTP response the server issued in response 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? Explain.



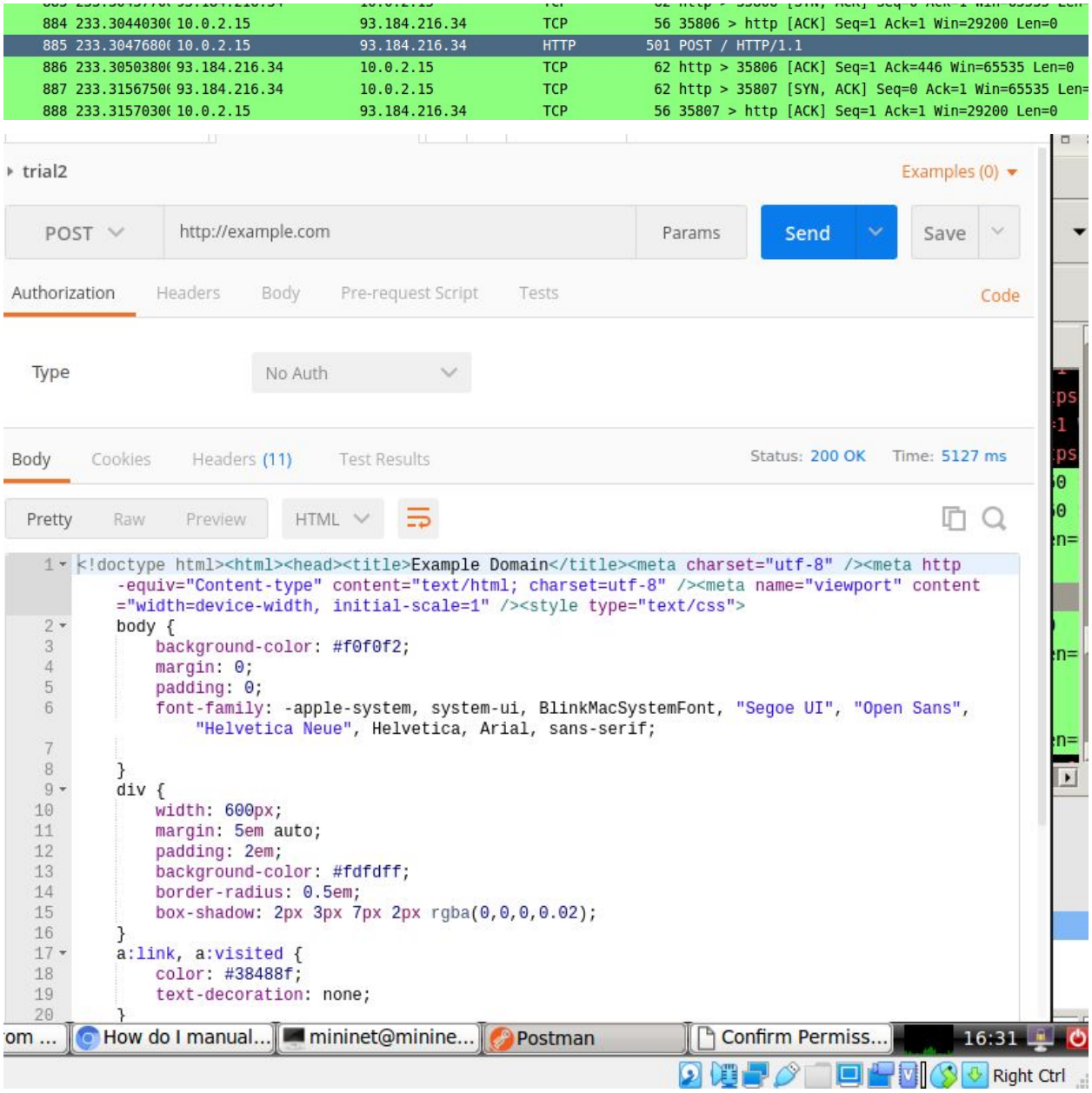
The http status code that was returned was 200 meaning okay.
The content type being sent back is text/html

Using Chromium, navigate to http://www.soe.ucsc.edu (not https!, Make sure to clear your browser cache):
3. (10) Find the packets that correspond to the initial HTTP request and response that your computer issued/received. Take a screenshot of these packets. What’s different?



The difference between these and the previous are 454 vs 453. As well as in terms of the response 301 (moved permanently (text/htm) is carried over in the message.

4. Explain. Using Chromium (or any other Linux utility you are comfortable with), find a way to create an HTTP message using a method other than GET.
(10) Take a screenshot of your packet and explain what you did to create it.
f

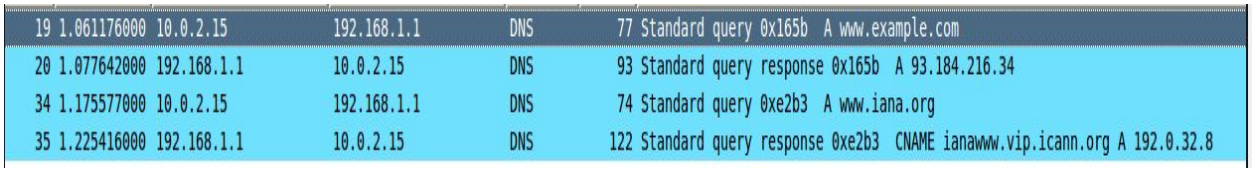


Using a chromium extension I was able to sent a post request to http://example.com
In doing so i was able to make html packet request utilizing post which is able to send and receive data unlike get which can only receive.

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. (5) 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. If not, explain why your computer did not need to take these steps.

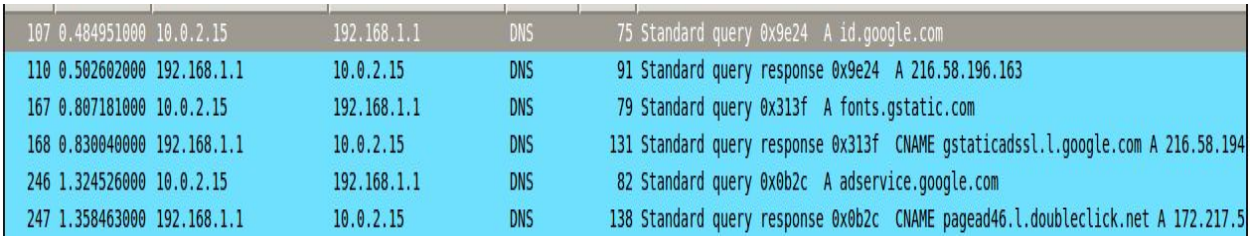


19	1.061176000	10.0.2.15	192.168.1.1	DNS	77 Standard query 0x165b A www.example.com
20	1.077642000	192.168.1.1	10.0.2.15	DNS	93 Standard query response 0x165b A 93.184.216.34
34	1.175577000	10.0.2.15	192.168.1.1	DNS	74 Standard query 0xe2b3 A www.iana.org
35	1.225416000	192.168.1.1	10.0.2.15	DNS	122 Standard query response 0xe2b3 CNAME ianawww.vip.icann.org A 192.0.32.8

I believe these are the correct packets because they query eample.com then we get a response and then the webpage is opened. We can see that from standary query to example.com then we receive a response from example.com and and from then onward we see the webpage.

In Chromium, navigate to http://216.58.193.68.

6. (5) 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://216.58.193.68. Take a screenshot of these packets, and explain why you think these are the correct packets. If not, explain why your computer did not need to take these steps.



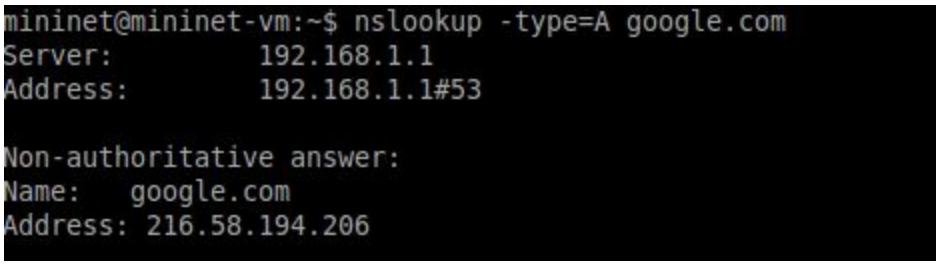
107	0.484951000	10.0.2.15	192.168.1.1	DNS	75 Standard query 0x9e24 A id.google.com
110	0.502602000	192.168.1.1	10.0.2.15	DNS	91 Standard query response 0x9e24 A 216.58.196.163
167	0.807181000	10.0.2.15	192.168.1.1	DNS	79 Standard query 0x313f A fonts.gstatic.com
168	0.830040000	192.168.1.1	10.0.2.15	DNS	131 Standard query response 0x313f CNAME gstaticadssl.l.google.com A 216.58.194
246	1.324526000	10.0.2.15	192.168.1.1	DNS	82 Standard query 0x0b2c A adservice.google.com
247	1.358463000	192.168.1.1	10.0.2.15	DNS	138 Standard query response 0x0b2c CNAME pagead46.l.doubleclick.net A 172.217.5

Yes it appears that there was a query made to id.google.com before the webpage was loaded. We also had to query and receive the response and then query fonts.gstatic.com before we were sent to the appropriate page.

Open a terminal window.

Using nslookup, find the A records for www.google.com.

7. (5) 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?

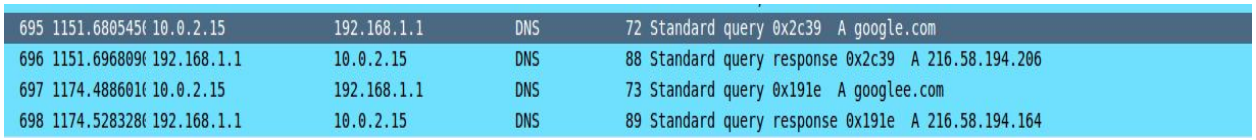


```
mininet@mininet-vm:~$ nslookup -type=A google.com
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
Name:   google.com
Address: 216.58.194.206
```

Address is 216.58.194.206

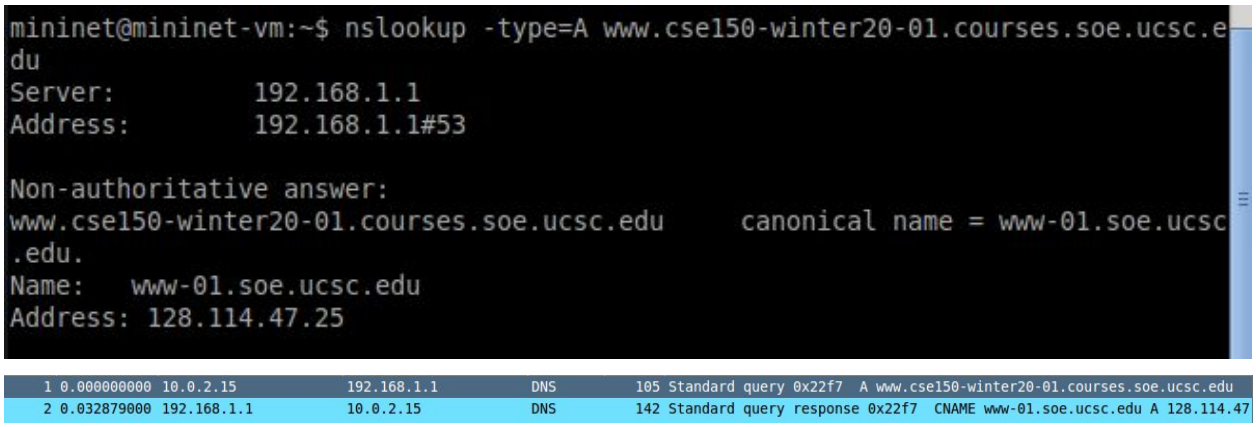
8. (5) Did your computer want to complete the request recursively? How do you know? Take a screenshot proving your answer.



695	1151.680545(	10.0.2.15	192.168.1.1	DNS	72 Standard query 0x2c39 A google.com
696	1151.696809(	192.168.1.1	10.0.2.15	DNS	88 Standard query response 0x2c39 A 216.58.194.206
697	1174.488601(	10.0.2.15	192.168.1.1	DNS	73 Standard query 0x191e A googlee.com
698	1174.528328(	192.168.1.1	10.0.2.15	DNS	89 Standard query response 0x191e A 216.58.194.164

Yes it does want to do that because its repeatedly querying the server based off the packets above.

Using nslookup,
find the A records for www.cse150-winter20-01.courses.soe.ucsc.edu
9. (5) 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.cse150-winter20-01.courses.soe.ucsc.edu?



```
mininet@mininet-vm:~$ nslookup -type=A www.cse150-winter20-01.courses.soe.ucsc.e
du
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
www.cse150-winter20-01.courses.soe.ucsc.edu     canonical name = www-01.soe.ucsc
.edu.
Name:   www-01.soe.ucsc.edu
Address: 128.114.47.25
```


1	0.000000000	10.0.2.15	192.168.1.1	DNS	105 Standard query 0x22f7 A www.cse150-winter20-01.courses.soe.ucsc.edu
2	0.032879000	192.168.1.1	10.0.2.15	DNS	142 Standard query response 0x22f7 CNAME www-01.soe.ucsc.edu A 128.114.47

Address is 128.114.47.25
The packets corresponding to the ns lookup are above.

10.(5) What is the authoritative name server for the ucsc.edu domain? How do you know? Take a screenshot proving your answer.

```
ininet@mininet-vm:~$ nslookup -type=soa ucsc.edu
erver:         192.168.1.1
ddress:        192.168.1.1#53

on-authoritative answer:
csc.edu
        origin = adns1.ucsc.edu
        mail addr = hostmaster.ucsc.edu
        serial = 17663742
        refresh = 10800
        retry = 900
        expire = 2419200
        minimum = 900

uthoritative answers can be found from:
```
\

No answer provided.
It would showup thought =soa is the correct prompt.

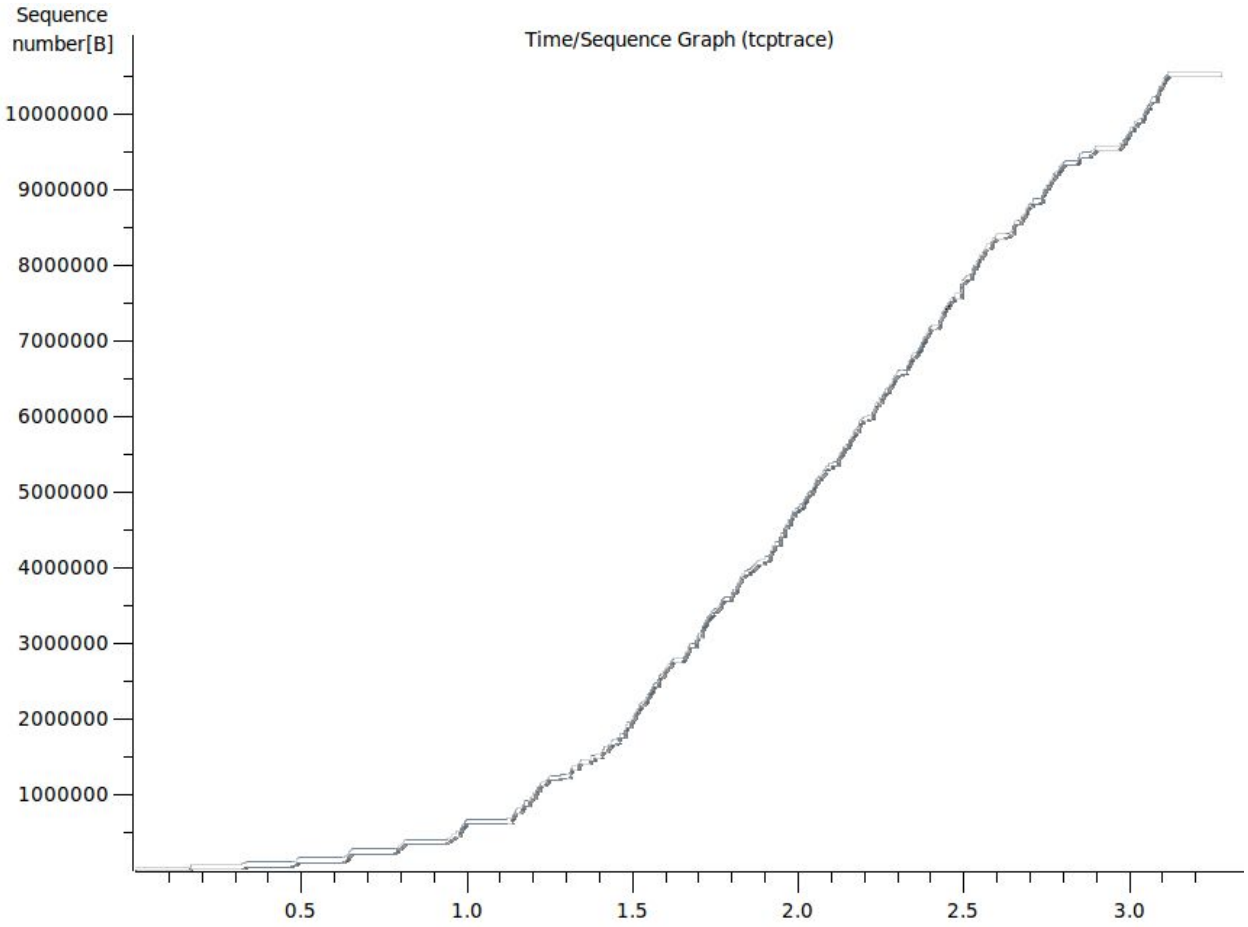
Part 3: TCP In this section, we will observe how the TCP protocol operates. We will do this by using the Mininet VM. Begin by opening Wireshark and listening on the ‘any’ interface. Open a terminal window. Using wget, download the file
http://ipv4.download.thinkbroadband.com/10MB.zip

11.(15) Find the packets corresponding to 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?

```
  5 0.463101000 10.0.2.15      80.249.99.148   TCP      76 48856 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=4294
  6 0.671851000 80.249.99.148  10.0.2.15       TCP      62 http > 48856 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
  7 0.671893000 10.0.2.15      80.249.99.148   TCP      56 48856 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
  8 0.671978000 10.0.2.15      80.249.99.148   HTTP    194 GET /10MB.zip HTTP/1.1
```

Window size 29200 - 6535-29200

12.(10) Find a packet from the download whose source address is the server’s address and the destination address is your computer’s address. 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

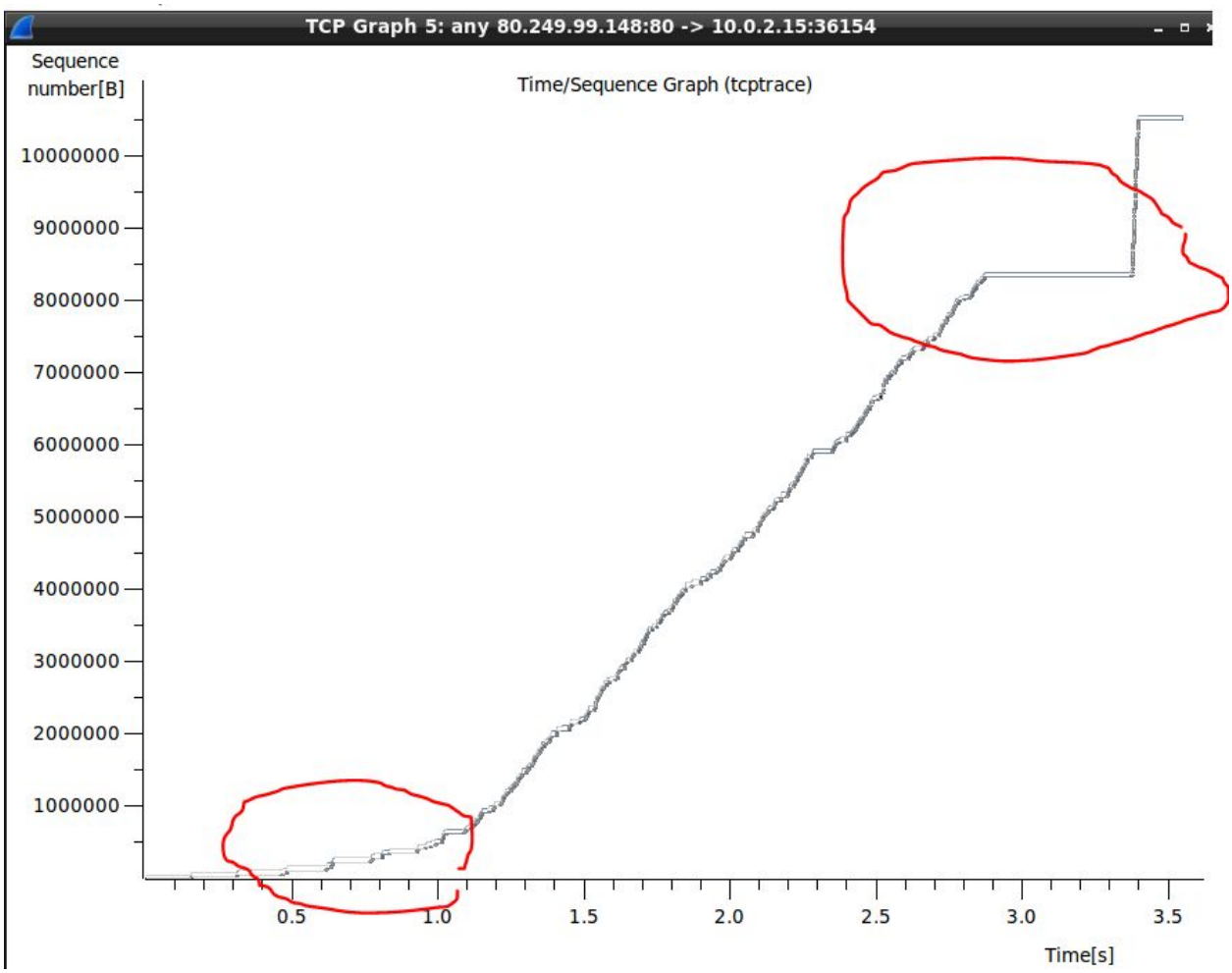


This graph is showing it took 3.5 seconds to get all of the data (10mbs)

. In the next section, we will be simulating loss using the command **tc qdisc**. 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. sudo tc qdisc add dev eth0 root netem loss 0% sudo tc qdisc change dev eth0 root netem loss 100% Read through this 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.(15) Find a packet from the download whose source address is the address of the server and destination address is your computer's address. 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.



We can see that in the beginning we see the tcp is slow start s for congestion avoidance. The second circle the horizontal line reflects the change to 0. We can see the file be downloaded and the effects of changing from 0 to 100 percent.
Congestion avoidance is utilized so as to account for potential traffic build up on a network.