Shawn Chumbar
CMPE 150
Professor Christina Parsa

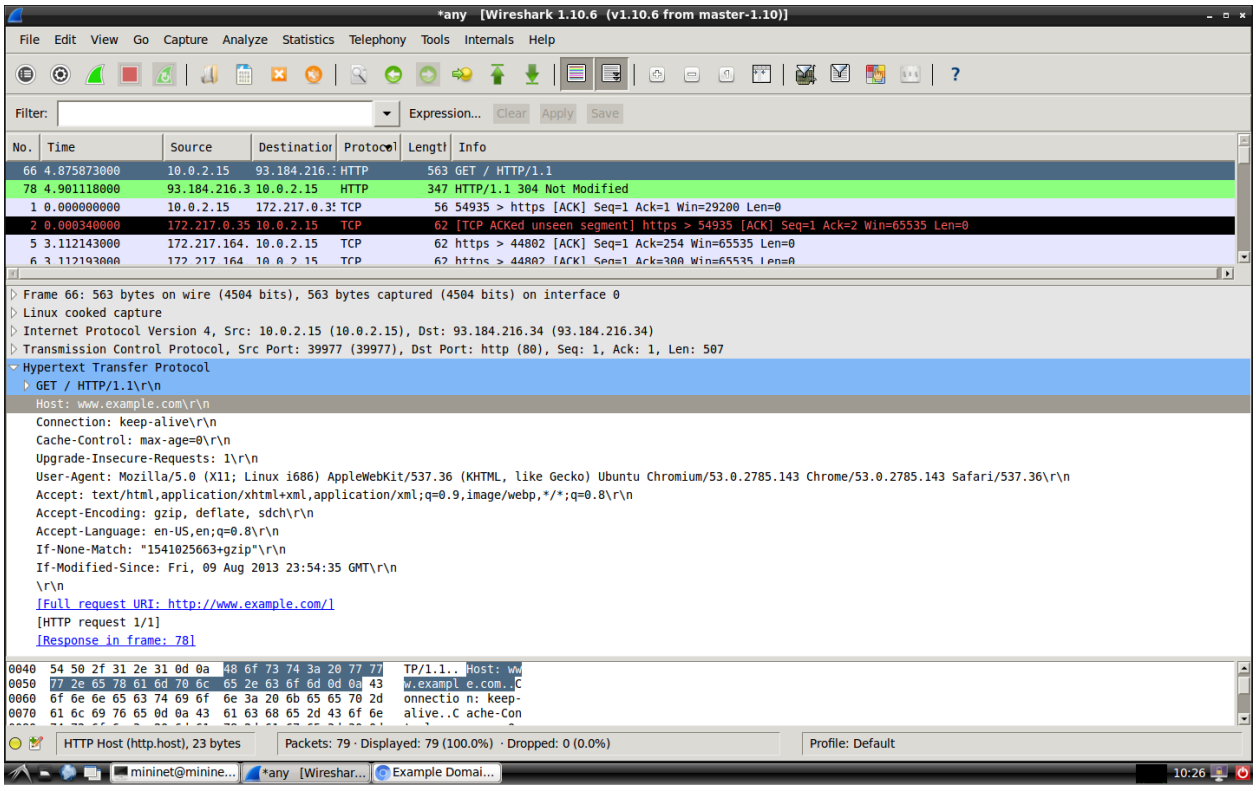
Lab 2

Part1: 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!):

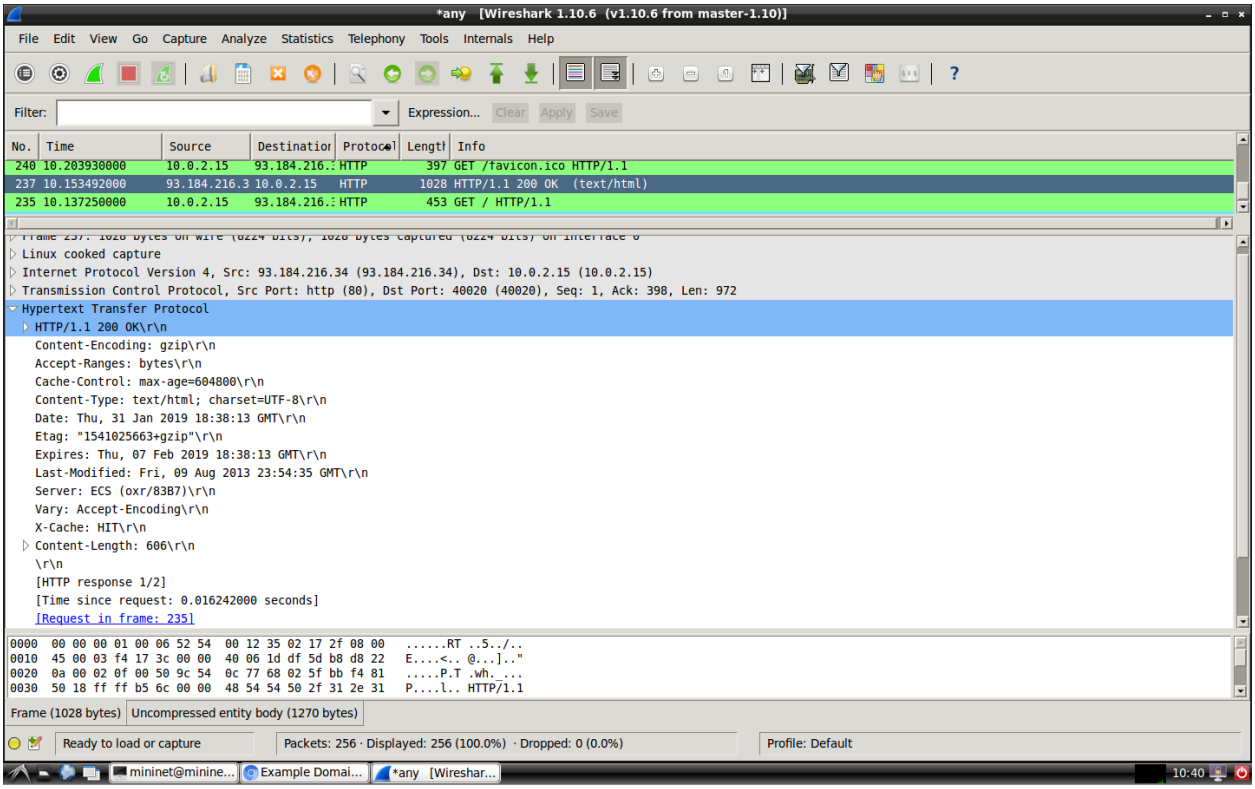
1. (5) Find the packet that corresponds to the initial HTTP request that your computer issued. Take a screenshot of this packet. What HTTP method did your computer use to make this request? What URI did your computer request from the server, as present in the HTTP request? (Note: NOT the URL). Explain.

The HTTP method used was GET, and http://www.example.com/ was the URI requested from the server. The GET method means that the computer is trying to get the webpage form the webserver.



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.

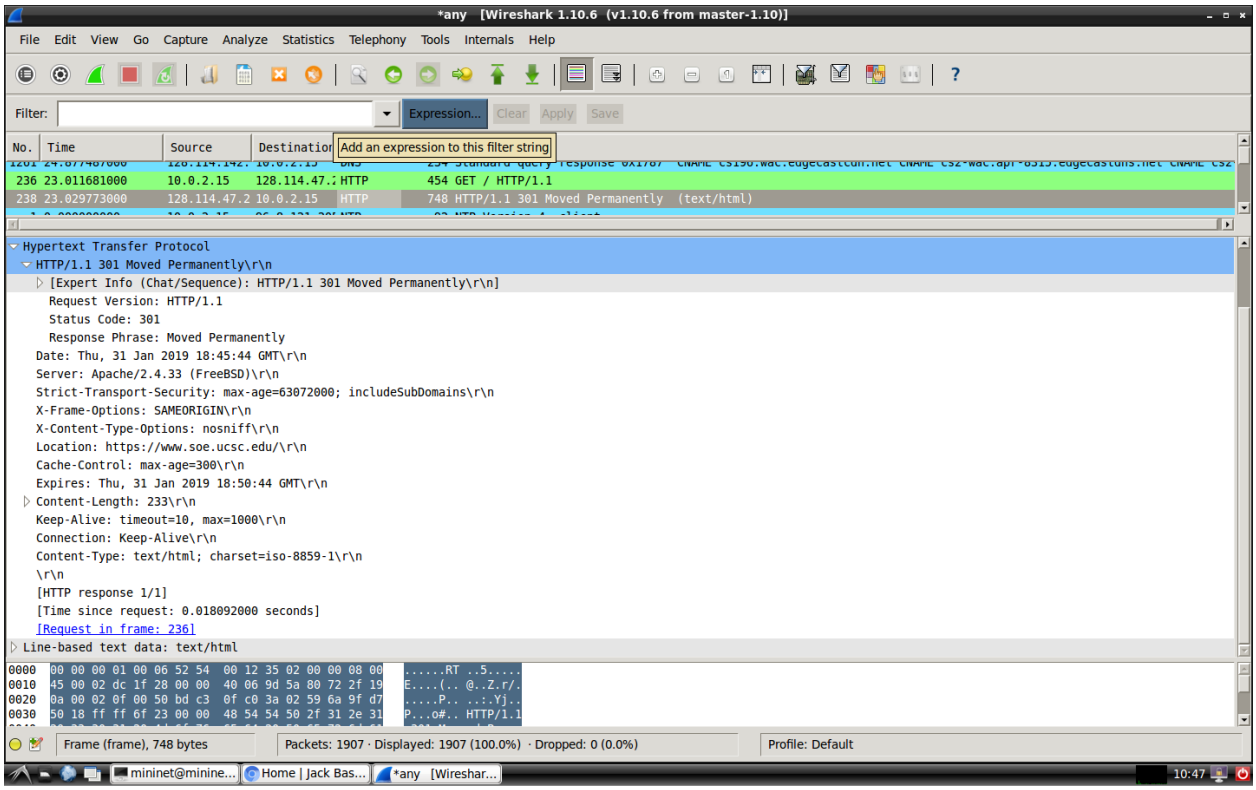
The initial HTTP response that the server issued in response to the GET request was a 200 OK response. The content type of the response that the server is sending back is text/html.

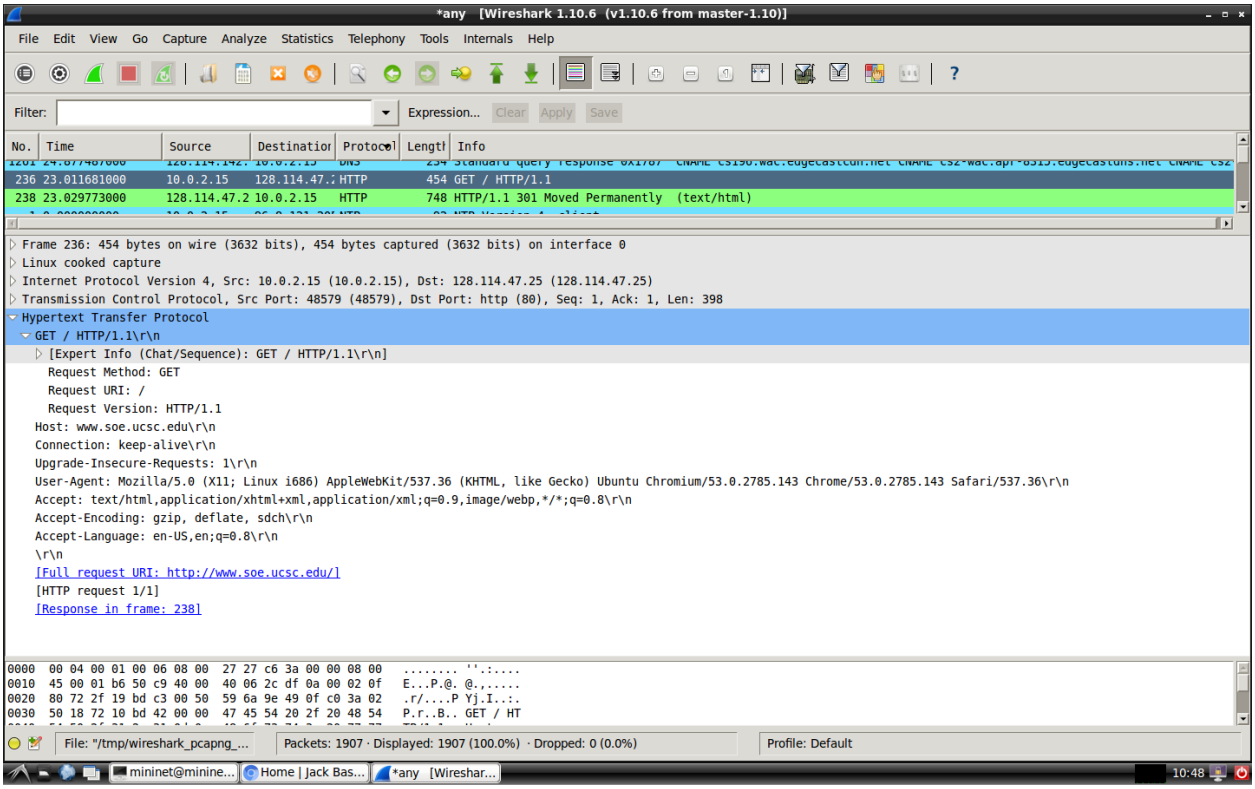


Using Chromium, navigate to http://www.soe.ucsc.edu (not https!):

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? Explain.

Instead of returning a status code of 200 OK, the response that was returned was a 301 Moved Permanently and the Request URI of the request was /. This is most likely due to the fact that the website was moved to https://soe.ucsc.edu to be more secure.



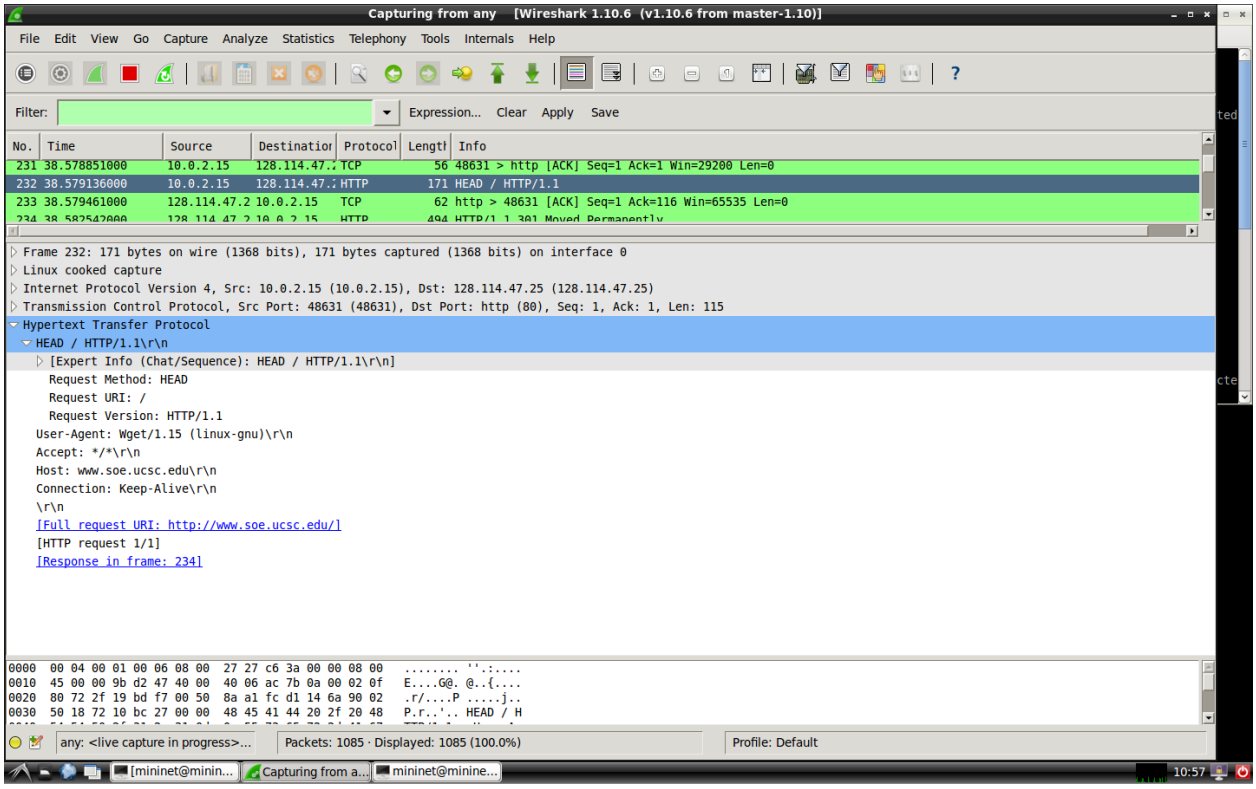

*any [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply Save
No. Time Source Destination Protocol Length Info
236 23.011681000 10.0.2.15 128.114.47.. HTTP 454 GET / HTTP/1.1
238 23.029773000 128.114.47.2 10.0.2.15 HTTP 748 HTTP/1.1 301 Moved Permanently (text/html)
Frame 236: 454 bytes on wire (3632 bits), 454 bytes captured (3632 bits) on interface 0
Linux cooked capture
Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 128.114.47.25 (128.114.47.25)
Transmission Control Protocol, Src Port: 48579 (48579), Dst Port: http (80), Seq: 1, Ack: 1, Len: 398
Hypertext Transfer Protocol
GET / HTTP/1.1\r\n
[Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
Request Method: GET
Request URI: /
Request Version: HTTP/1.1
Host: www.soe.ucsc.edu\r\n
Connection: keep-alive\r\n
Upgrade-Insecure-Requests: 1\r\n
User-Agent: Mozilla/5.0 (X11; Linux i686) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/53.0.2785.143 Chrome/53.0.2785.143 Safari/537.36\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
Accept-Encoding: gzip, deflate, sdch\r\n
Accept-Language: en-US,en;q=0.8\r\n
\r\n
[Full request URI: http://www.soe.ucsc.edu/]
[HTTP request 1/1]
[Response in frame: 238]
0000 00 04 00 01 00 06 08 00 27 27 c6 3a 00 00 08 00 .... ''.:....
0010 45 00 01 b6 50 c9 40 00 40 06 2c df 0a 00 02 0f E...P.@. @.,.....
0020 80 72 2f 19 bd c3 00 50 59 6a 9e 49 0f c0 3a 02 .r/....P Yj.I..:.
0030 50 18 72 10 bd 42 00 00 47 45 54 20 2f 20 48 54 P.r..B.. GET / HT
File: "/tmp/wireshark_pcapng_...
Packets: 1907 · Displayed: 1907 (100.0%) · Dropped: 0 (0.0%)
Profile: Default
mininet@minine...
Home | Jack Bas...
*any [Wireshar...
10:48

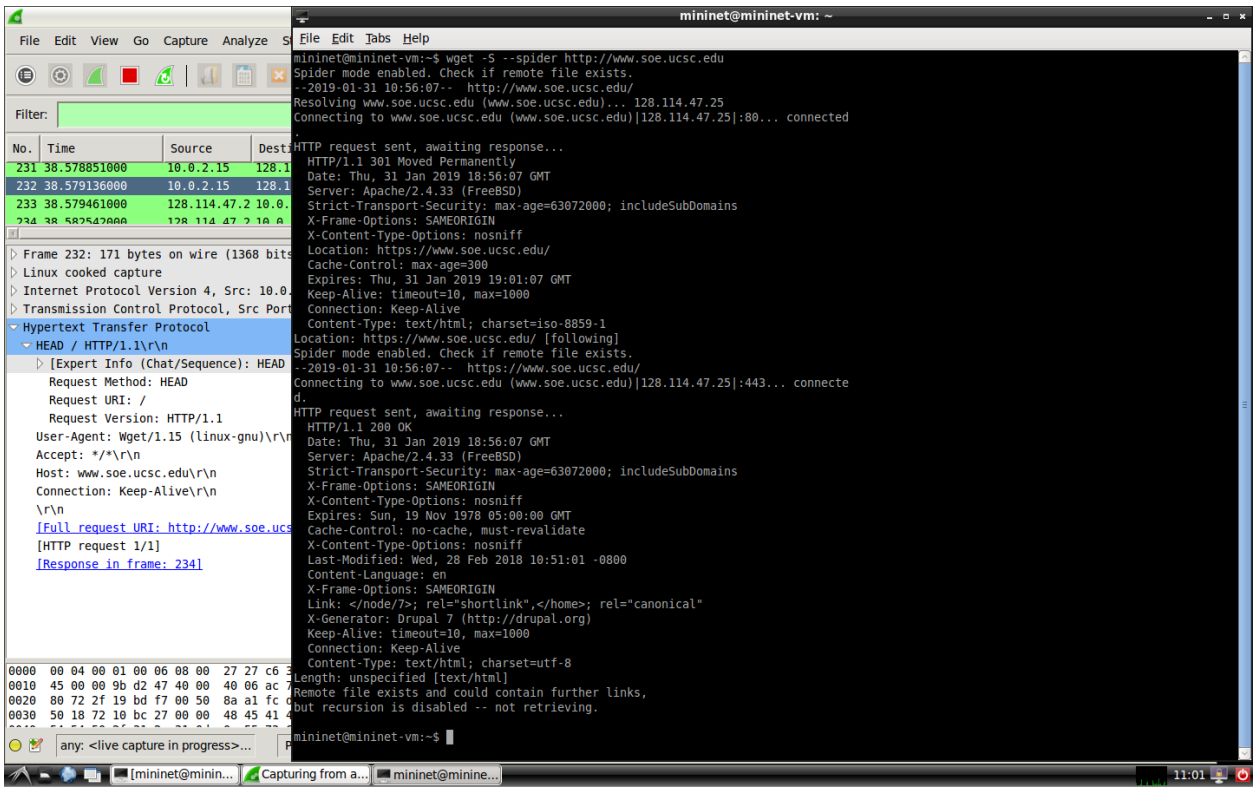
Using Chromium (or any other Linux utility you are comfortable with), find a way to create an HTTP message using a method other than GET.

4. (10) Take a screenshot of your packet and explain what you did to create it.

To create an HTTP message using the HEAD method, I simply opened up the Linux terminal and typed in the following command:

wget –S –spider http://soe.ucsc.edu



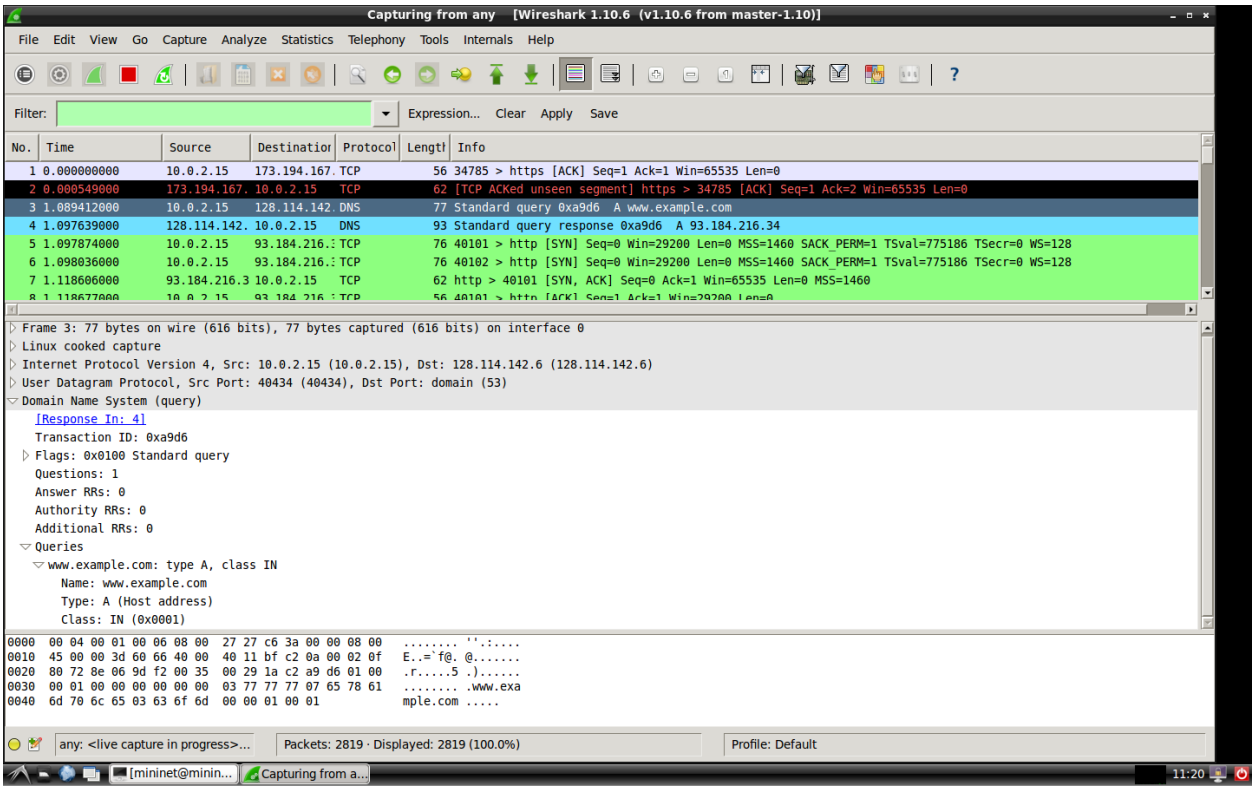

File Edit View Go Capture Analyze
Filter:
No. Time Source
231 38.578851000 10.0.2.15
232 38.579136000 10.0.2.15
233 38.579461000 128.114.47.2
Frame 232: 171 bytes on wire (1368 bits
Linux cooked capture
Internet Protocol Version 4, Src: 10.0.
Transmission Control Protocol, Src Port
Hypertext Transfer Protocol
HEAD / HTTP/1.1\r\n
[Expert Info (Chat/Sequence): HEAD
Request Method: HEAD
Request URI: /
Request Version: HTTP/1.1
User-Agent: Wget/1.15 (linux-gnu)\r\n
Accept: */*\r\n
Host: www.soe.ucsc.edu\r\n
Connection: Keep-Alive\r\n
\r\n
[Full request URI: http://www.soe.ucs
[HTTP request 1/1]
[Response in frame: 234]
any: <live capture in progress>...
mininet@mininet-vm: ~
File Edit Tabs Help
mininet@mininet-vm:~$ wget -S --spider http://www.soe.ucsc.edu
Spider mode enabled. Check if remote file exists.
--2019-01-31 10:56:07-- http://www.soe.ucsc.edu/
Resolving www.soe.ucsc.edu (www.soe.ucsc.edu)... 128.114.47.25
Connecting to www.soe.ucsc.edu (www.soe.ucsc.edu)|128.114.47.25|:80... connected
.
HTTP request sent, awaiting response...
HTTP/1.1 301 Moved Permanently
Date: Thu, 31 Jan 2019 18:56:07 GMT
Server: Apache/2.4.33 (FreeBSD)
Strict-Transport-Security: max-age=63072000; includeSubDomains
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
Location: https://www.soe.ucsc.edu/
Cache-Control: max-age=300
Expires: Thu, 31 Jan 2019 19:01:07 GMT
Keep-Alive: timeout=10, max=1000
Connection: Keep-Alive
Content-Type: text/html; charset=iso-8859-1
Location: https://www.soe.ucsc.edu/ [following]
Spider mode enabled. Check if remote file exists.
--2019-01-31 10:56:07-- https://www.soe.ucsc.edu/
Connecting to www.soe.ucsc.edu (www.soe.ucsc.edu)|128.114.47.25|:443... connecte
d.
HTTP request sent, awaiting response...
HTTP/1.1 200 OK
Date: Thu, 31 Jan 2019 18:56:07 GMT
Server: Apache/2.4.33 (FreeBSD)
Strict-Transport-Security: max-age=63072000; includeSubDomains
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
Expires: Sun, 19 Nov 1978 05:00:00 GMT
Cache-Control: no-cache, must-revalidate
X-Content-Type-Options: nosniff
Last-Modified: Wed, 28 Feb 2018 10:51:01 -0800
Content-Language: en
X-Frame-Options: SAMEORIGIN
Link: </node/7>; rel="shortlink",</home>; rel="canonical"
X-Generator: Drupal 7 (http://drupal.org)
Keep-Alive: timeout=10, max=1000
Connection: Keep-Alive
Content-Type: text/html; charset=utf-8
Length: unspecified [text/html]
Remote file exists and could contain further links,
but recursion is disabled -- not retrieving.
mininet@mininet-vm:~$
[mininet@minin...
Capturing from a...
mininet@minine...
11:01

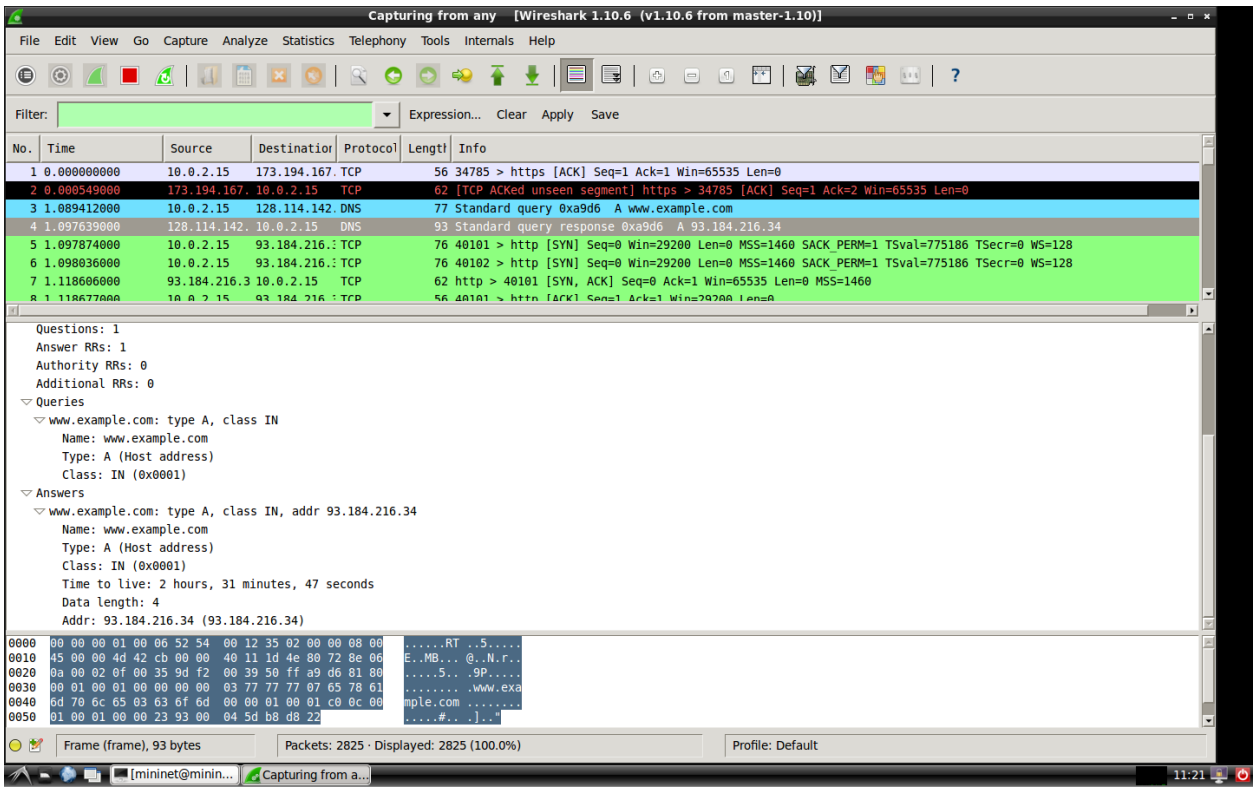
Part 2: DNS

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.

There were some DNS steps taken before the web page was loaded. This included a DNS query and a DNS response. I believe that these are the correct packets because of the fact that Wireshark is showing that these packets were used to translate the domain name into an IP address.

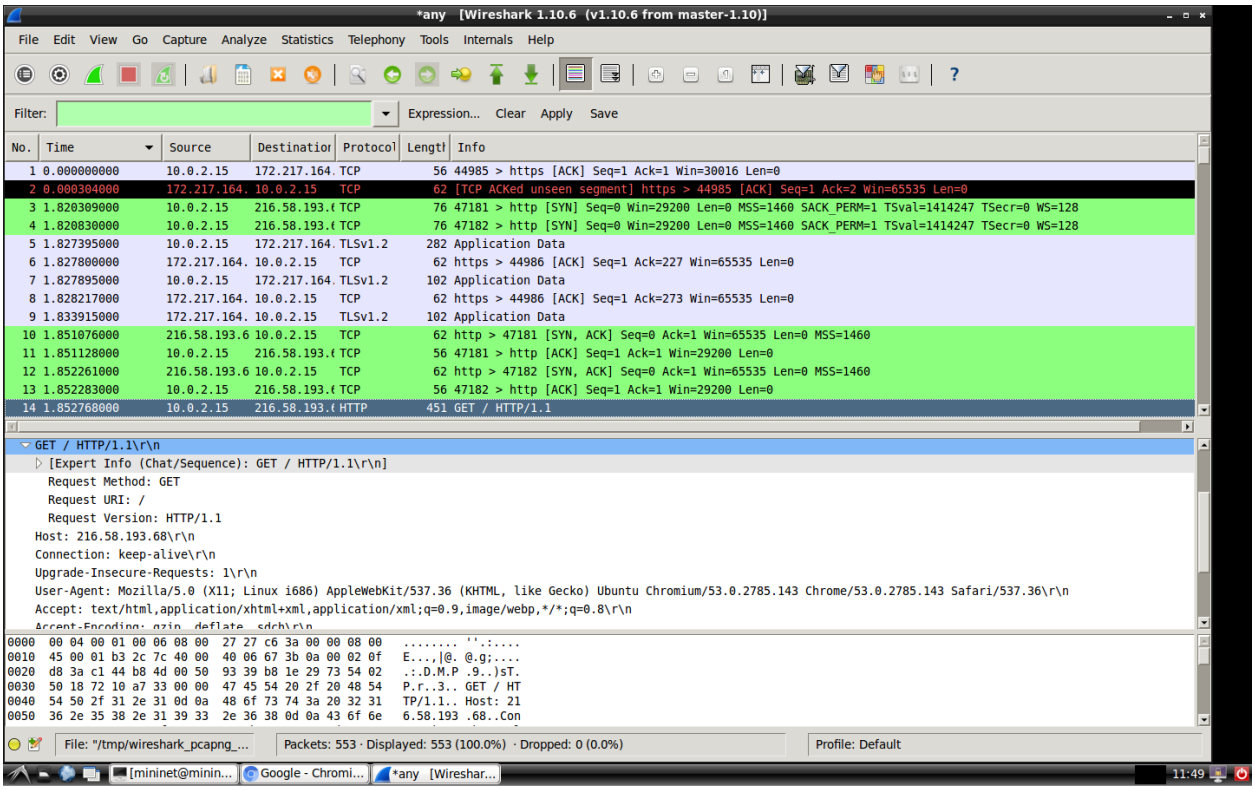



Capturing from any [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply Save
No. Time Source Destination Protocol Length Info
1 0.000000000 10.0.2.15 173.194.167. TCP 56 34785 > https [ACK] Seq=1 Ack=1 Win=65535 Len=0
2 0.000549000 173.194.167. 10.0.2.15 TCP 62 [TCP ACKed unseen segment] https > 34785 [ACK] Seq=1 Ack=2 Win=65535 Len=0
3 1.089412000 10.0.2.15 128.114.142. DNS 77 Standard query 0xa9d6 A www.example.com
4 1.097639000 128.114.142. 10.0.2.15 DNS 93 Standard query response 0xa9d6 A 93.184.216.34
5 1.097874000 10.0.2.15 93.184.216. TCP 76 40101 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=775186 TSecr=0 WS=128
6 1.098036000 10.0.2.15 93.184.216. TCP 76 40102 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=775186 TSecr=0 WS=128
7 1.118606000 93.184.216.3 10.0.2.15 TCP 62 http > 40101 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
Questions: 1
Answer RRs: 1
Authority RRs: 0
Additional RRs: 0
Queries
www.example.com: type A, class IN
Name: www.example.com
Type: A (Host address)
Class: IN (0x0001)
Answers
www.example.com: type A, class IN, addr 93.184.216.34
Name: www.example.com
Type: A (Host address)
Class: IN (0x0001)
Time to live: 2 hours, 31 minutes, 47 seconds
Data length: 4
Addr: 93.184.216.34 (93.184.216.34)
0000 00 00 00 01 00 06 52 54 00 12 35 02 00 00 08 00 ......RT ..5.....
0010 45 00 00 4d 42 cb 00 00 40 11 1d 4e 80 72 8e 06 E..MB... @..N.r..
0020 0a 00 02 0f 00 35 9d f2 00 39 50 ff a9 d6 81 80 .....5.. .9P.....
0030 00 01 00 01 00 00 00 00 03 77 77 77 07 65 78 61 .... .www.exa
0040 6d 70 6c 65 03 63 6f 6d 00 00 01 00 01 c0 0c 00 mple.com ....
0050 01 00 01 00 00 23 93 00 04 5d b8 d8 22 .....#.. .]..."
Frame (frame), 93 bytes
Packets: 2825 · Displayed: 2825 (100.0%)
Profile: Default
[mininet@minin...
Capturing from a...
11:21

In Chromium, navigate to http://216.58.193.68.

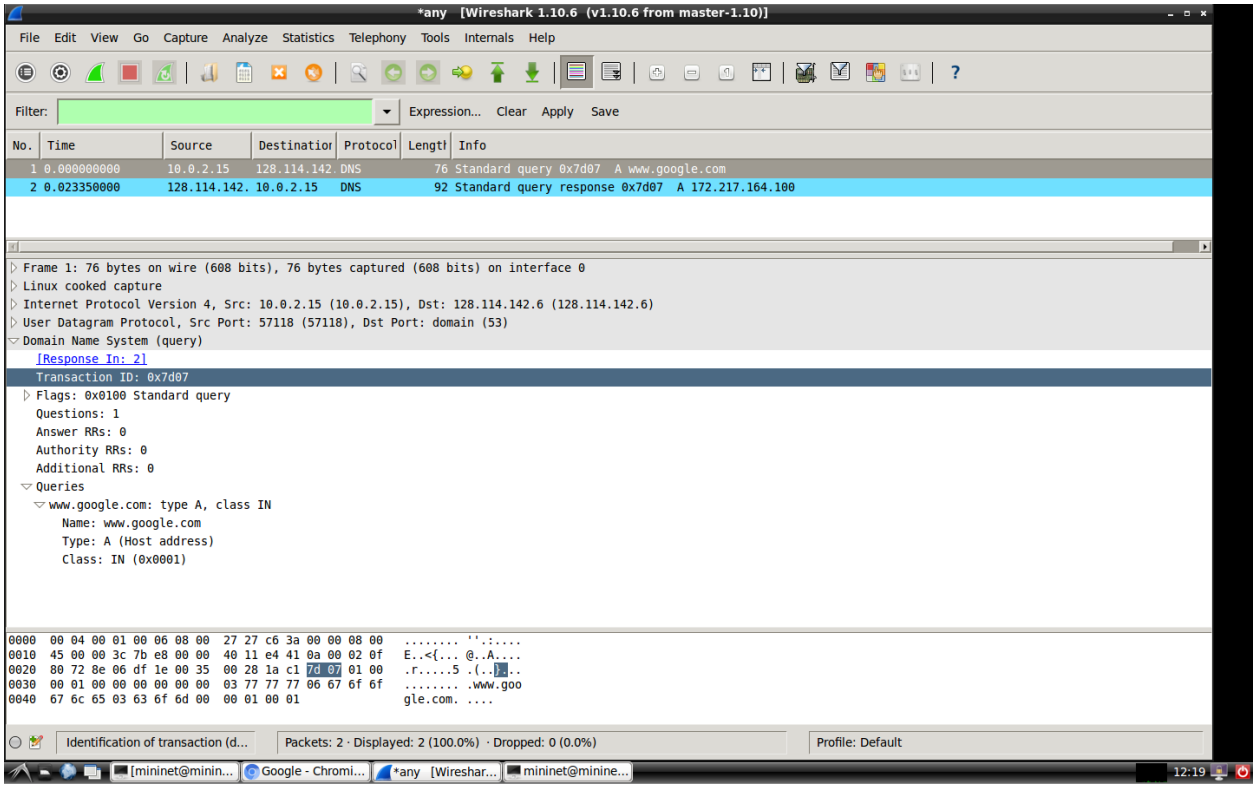
6. (5) Were any steps taken by your computer before the web page was loaded? If so, using your captured packets in Wireshark, find the packets that allowed your computer to successfully load http://216.58.193.68. Take a screenshot of these packets, and explain why you think these are the correct packets. If not, explain why your computer did not need to take these steps.

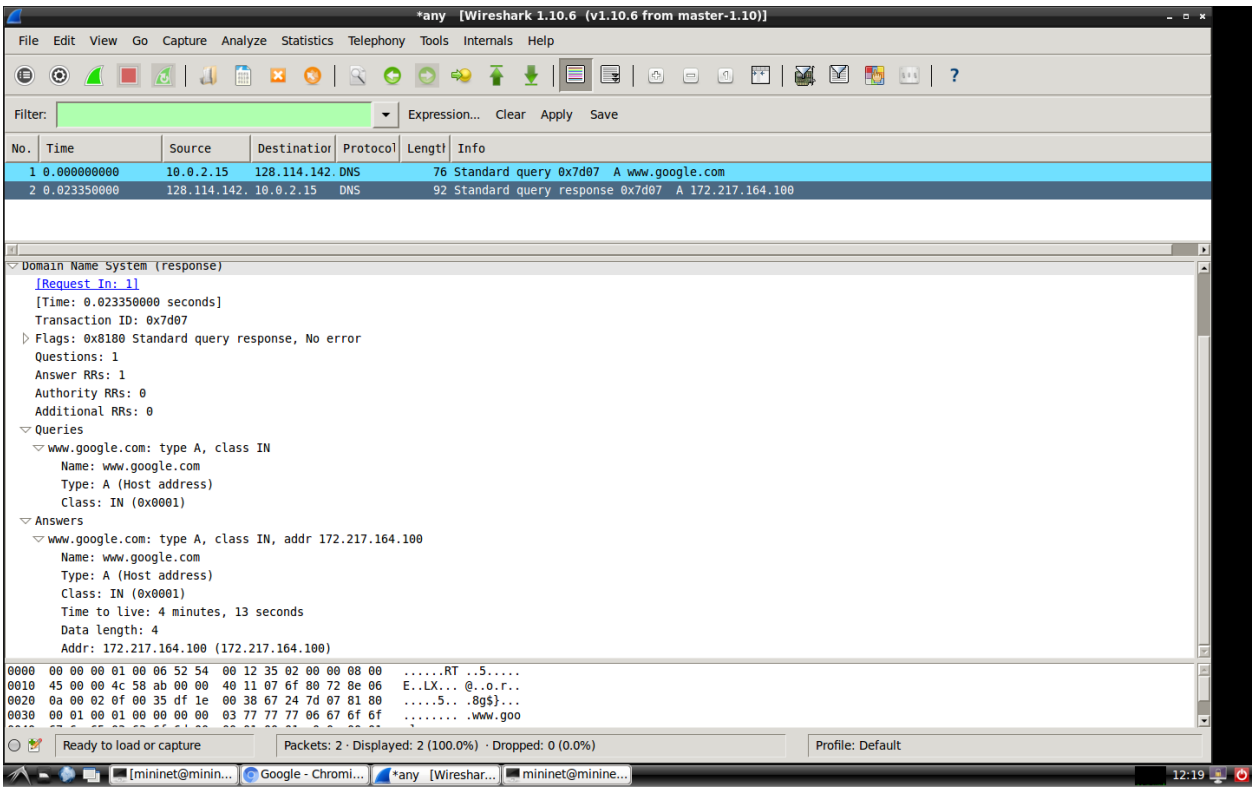
There were no steps taken before the webpage was loaded. The computer did not have to take any extra DNS steps, because we gave it the IP address, and as a result, there was no need for a DNS query to translate the domain name into an IP address.



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?

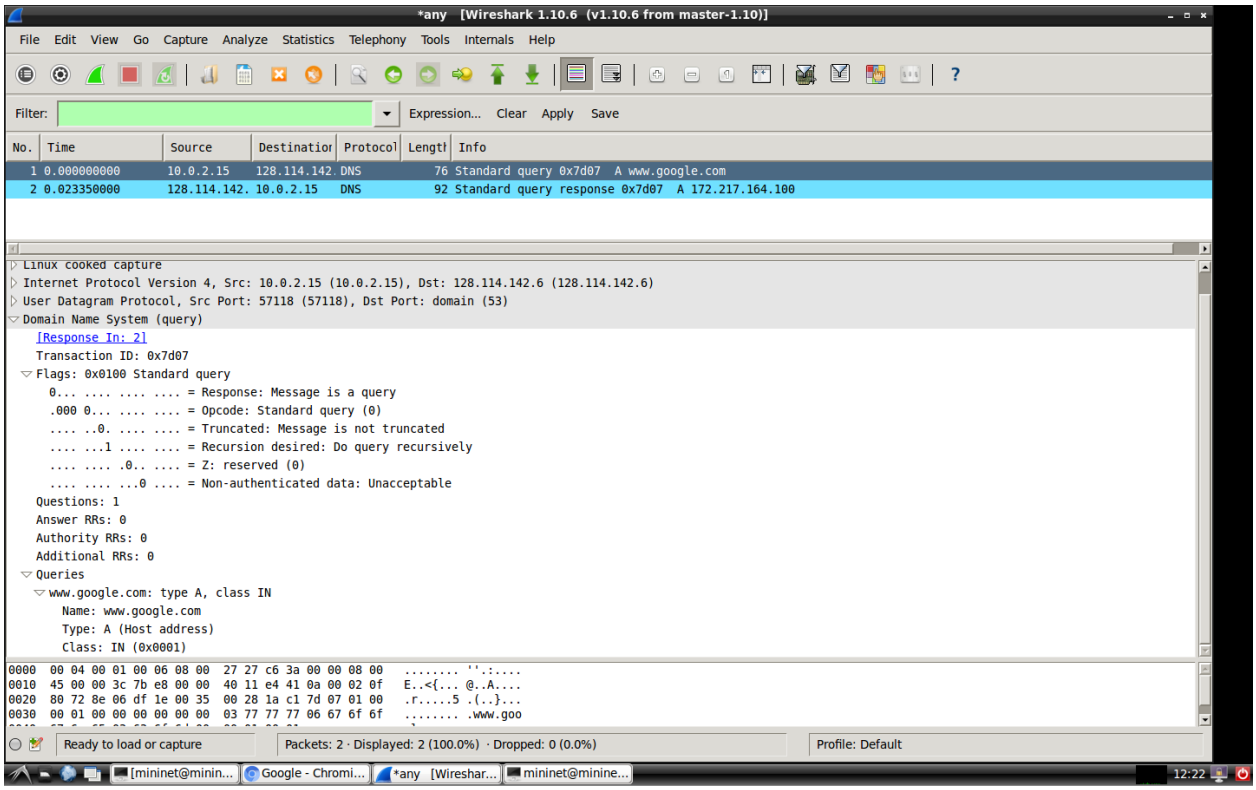
The request was resolved, and the IP address that I was given for www.google.com was 172.217.164.100.

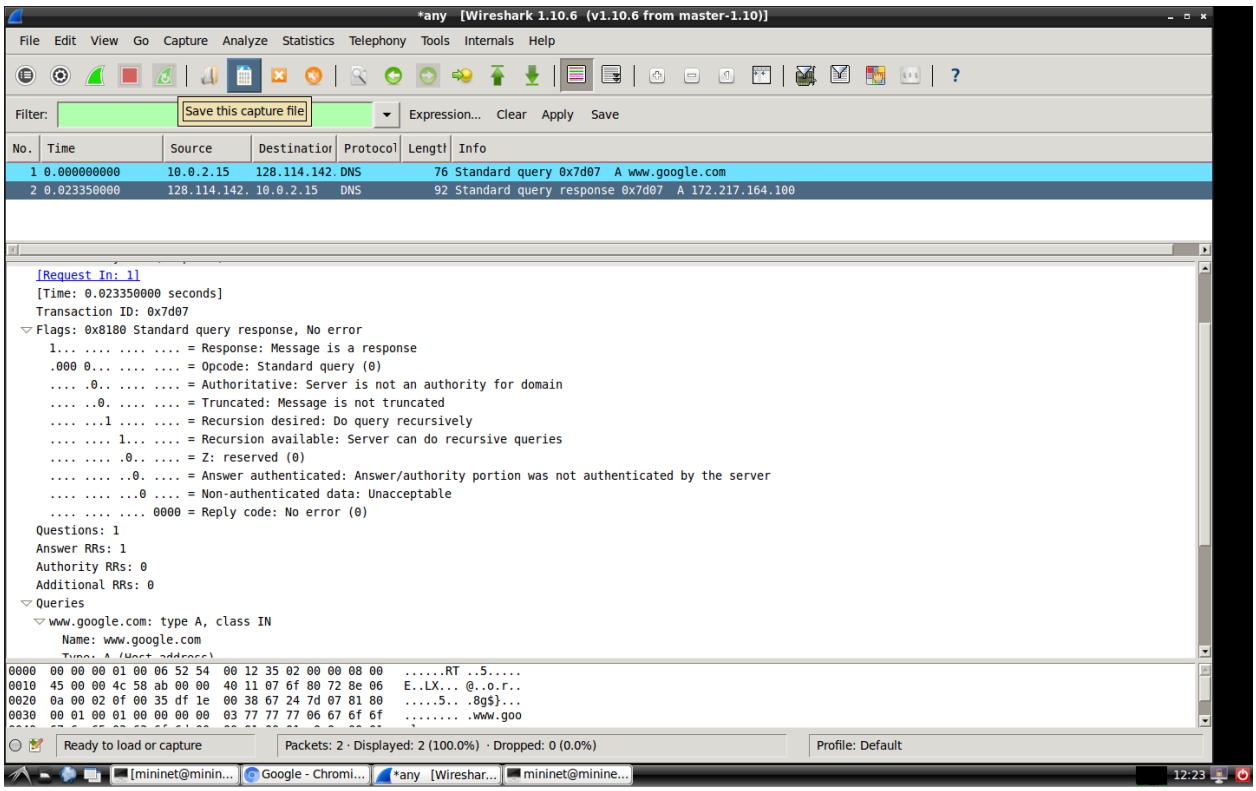



*any [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply Save
No. Time Source Destinatior Protocol Lengtl Info
1 0.000000000 10.0.2.15 128.114.142. DNS 76 Standard query 0x7d07 A www.google.com
2 0.023350000 128.114.142. 10.0.2.15 DNS 92 Standard query response 0x7d07 A 172.217.164.100
Domain Name System (response)
[Request In: 1]
[Time: 0.023350000 seconds]
Transaction ID: 0x7d07
Flags: 0x8180 Standard query response, No error
Questions: 1
Answer RRs: 1
Authority RRs: 0
Additional RRs: 0
Queries
www.google.com: type A, class IN
Name: www.google.com
Type: A (Host address)
Class: IN (0x0001)
Answers
www.google.com: type A, class IN, addr 172.217.164.100
Name: www.google.com
Type: A (Host address)
Class: IN (0x0001)
Time to live: 4 minutes, 13 seconds
Data length: 4
Addr: 172.217.164.100 (172.217.164.100)
0000 00 00 00 01 00 06 52 54 00 12 35 02 00 00 08 00 ......RT ..5.....
0010 45 00 00 4c 58 ab 00 00 40 11 07 6f 80 72 8e 06 E..LX... @..o.r..
0020 0a 00 02 0f 00 35 df 1e 00 38 67 24 7d 07 81 80 .....5.. .8g$}...
0030 00 01 00 01 00 00 00 00 03 77 77 77 06 67 6f 6f .... .www.goo
Ready to load or capture
Packets: 2 · Displayed: 2 (100.0%) · Dropped: 0 (0.0%)
Profile: Default
[mininet@minin...
Google - Chromi...
*any [Wireshar...
mininet@minine...
12:19

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

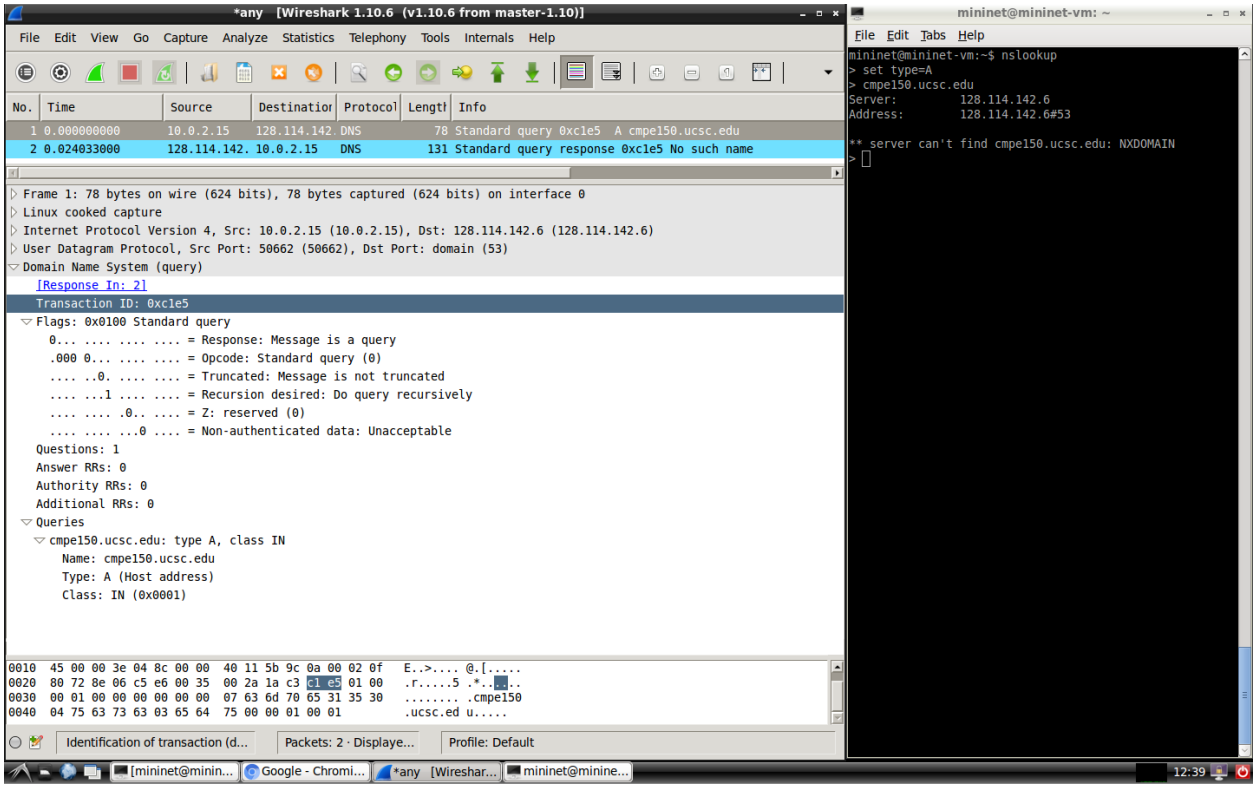
The computer wanted to complete the request recursively. We know this by the flags that are set. Both the query and the response have the “Recursion desired” flag set to 1, which means that the computer wanted to complete the request recursively. The response even has the “Recursion available” flag also set to 1.

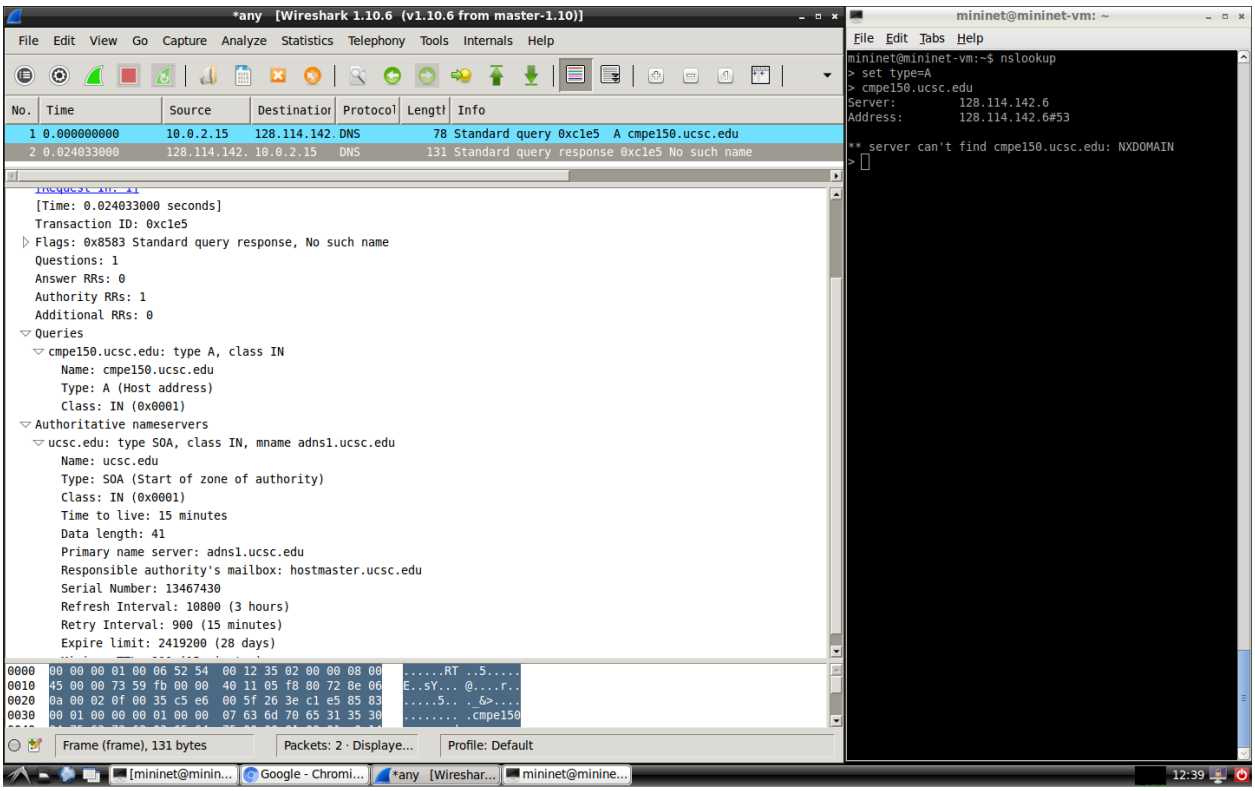



*any [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Save this capture file
Expression... Clear Apply Save
No. Time Source Destination Protocol Length Info
1 0.000000000 10.0.2.15 128.114.142. DNS 76 Standard query 0x7d07 A www.google.com
2 0.023350000 128.114.142. 10.0.2.15 DNS 92 Standard query response 0x7d07 A 172.217.164.100
[Request In: 1]
[Time: 0.023350000 seconds]
Transaction ID: 0x7d07
Flags: 0x8180 Standard query response, No error
1... .... = Response: Message is a response
.000 0... .... = Opcode: Standard query (0)
.... .0.. .... = Authoritative: Server is not an authority for domain
.... ..0. .... = Truncated: Message is not truncated
....1 .... = Recursion desired: Do query recursively
.... 1... .... = Recursion available: Server can do recursive queries
.... .0.. .... = Z: reserved (0)
....0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
....0 .... = Non-authenticated data: Unacceptable
.... 0000 = Reply code: No error (0)
Questions: 1
Answer RRs: 1
Authority RRs: 0
Additional RRs: 0
Queries
www.google.com: type A, class IN
Name: www.google.com
0000 00 00 00 01 00 06 52 54 00 12 35 02 00 00 08 00 ......RT ..5.....
0010 45 00 00 4c 58 ab 00 00 40 11 07 6f 80 72 8e 06 E..LX... @..o.r..
0020 0a 00 02 0f 00 35 df 1e 00 38 67 24 7d 07 81 80 .....5.. .8g$}...
0030 00 01 00 01 00 00 00 00 03 77 77 77 06 67 6f 6f .... .www.goo
Ready to load or capture
Packets: 2 · Displayed: 2 (100.0%) · Dropped: 0 (0.0%)
Profile: Default
[mininet@minin...
Google - Chromi...
*any [Wireshar...
mininet@minine...
12:23

Using nslookup, find the A records for cmpe150.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 cmpe150.ucsc.edu?

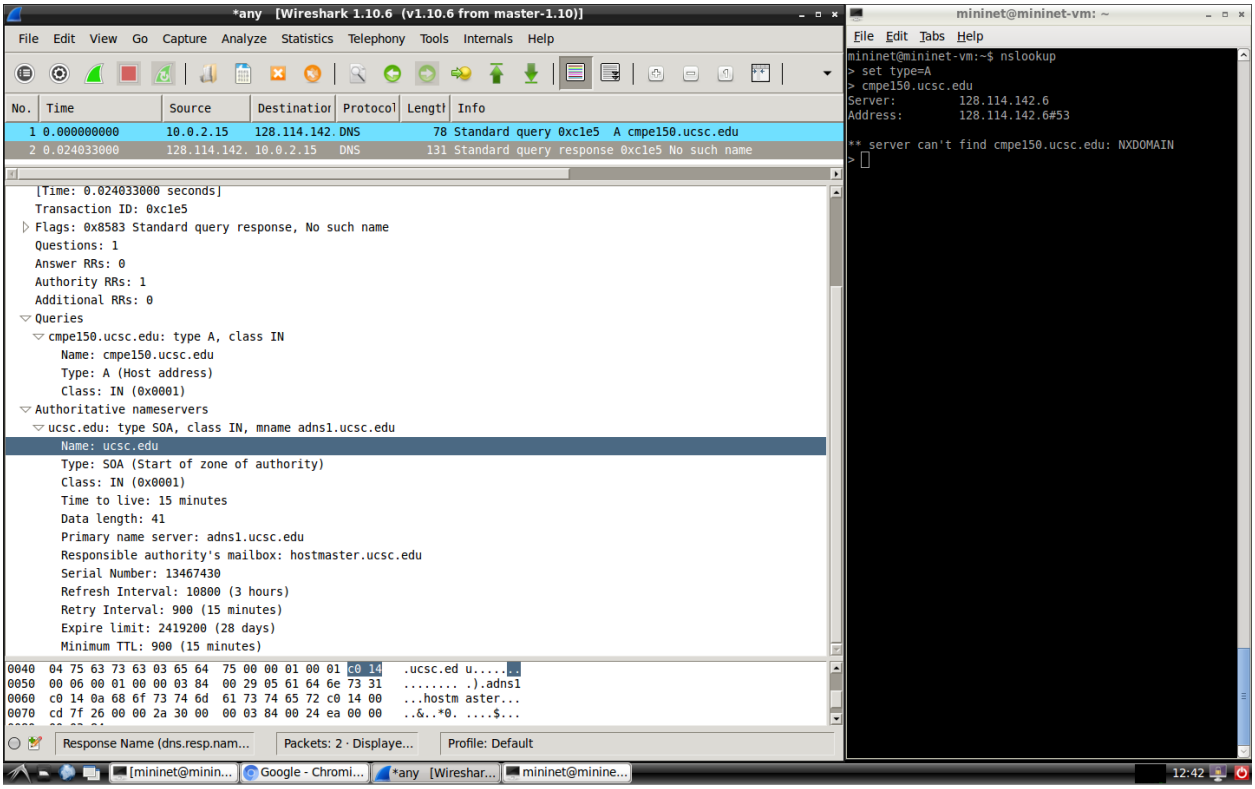
I was not given an IP address for cmpe150.ucsc.edu. After doing the nslookup command, there was an error message returned stating that we could not find the domain.




*any [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
No. Time Source Destination Protocol Length Info
1 0.000000000 10.0.2.15 128.114.142. DNS 78 Standard query 0xc1e5 A cmpe150.ucsc.edu
2 0.024033000 128.114.142. 10.0.2.15 DNS 131 Standard query response 0xc1e5 No such name
[Time: 0.024033000 seconds]
Transaction ID: 0xc1e5
Flags: 0x8583 Standard query response, No such name
Questions: 1
Answer RRs: 0
Authority RRs: 1
Additional RRs: 0
Queries
cmpe150.ucsc.edu: type A, class IN
Name: cmpe150.ucsc.edu
Type: A (Host address)
Class: IN (0x0001)
Authoritative nameservers
ucsc.edu: type SOA, class IN, mname adns1.ucsc.edu
Name: ucsc.edu
Type: SOA (Start of zone of authority)
Class: IN (0x0001)
Time to live: 15 minutes
Data length: 41
Primary name server: adns1.ucsc.edu
Responsible authority's mailbox: hostmaster.ucsc.edu
Serial Number: 13467430
Refresh Interval: 10800 (3 hours)
Retry Interval: 900 (15 minutes)
Expire limit: 2419200 (28 days)
0000 00 00 00 01 00 06 52 54 00 12 35 02 00 00 08 00 ......RT ..5.....
0010 45 00 00 73 59 fb 00 00 40 11 05 f8 80 72 8e 06 E..sY... @....r..
0020 0a 00 02 0f 00 35 c5 e6 00 5f 26 3e c1 e5 85 83 .....5.. ._&>....
0030 00 01 00 00 00 01 00 00 07 63 6d 70 65 31 35 30 .... .cmpe150
Frame (frame), 131 bytes
Packets: 2 · Displaye...
Profile: Default
[mininet@minin...
Google - Chromi...
*any [Wireshar...
mininet@minine...
12:39
mininet@mininet-vm: ~
File Edit Tabs Help
mininet@mininet-vm:~$ nslookup
> set type=A
> cmpe150.ucsc.edu
Server: 128.114.142.6
Address: 128.114.142.6#53
** server can't find cmpe150.ucsc.edu: NXDOMAIN
>

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

An authoritative name server is a server in which we can be 100% certain that the objects we are receiving are from the desired source. In other words, we are getting whatever we need directly from the source. The authoritative name servers for the ucsc.edu domain is: ucsc.edu.
We know this by looking at the DNS query response information.



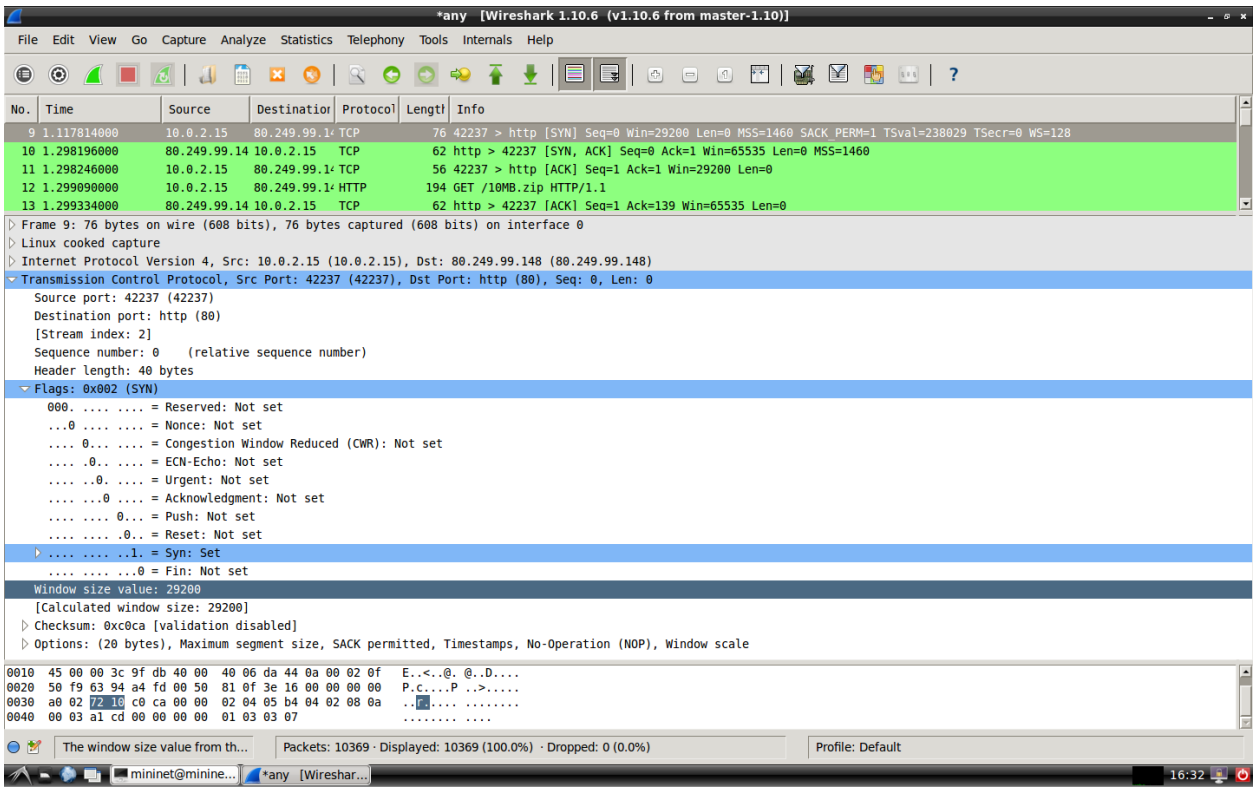
Part 3: TCP

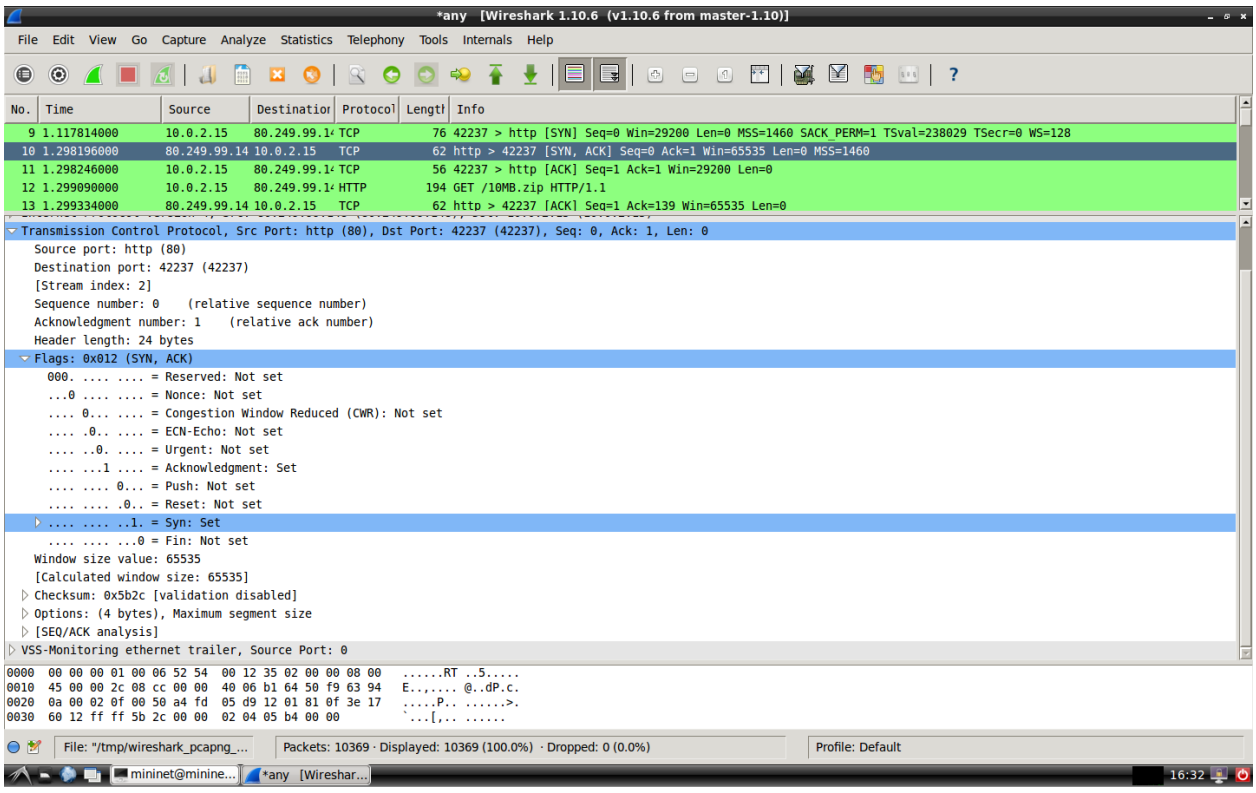
In this section, we will observe how the TCP protocol operates. We will do this by using the Mininet VM. Begin by opening Wireshark and listening on the 'any' interface.

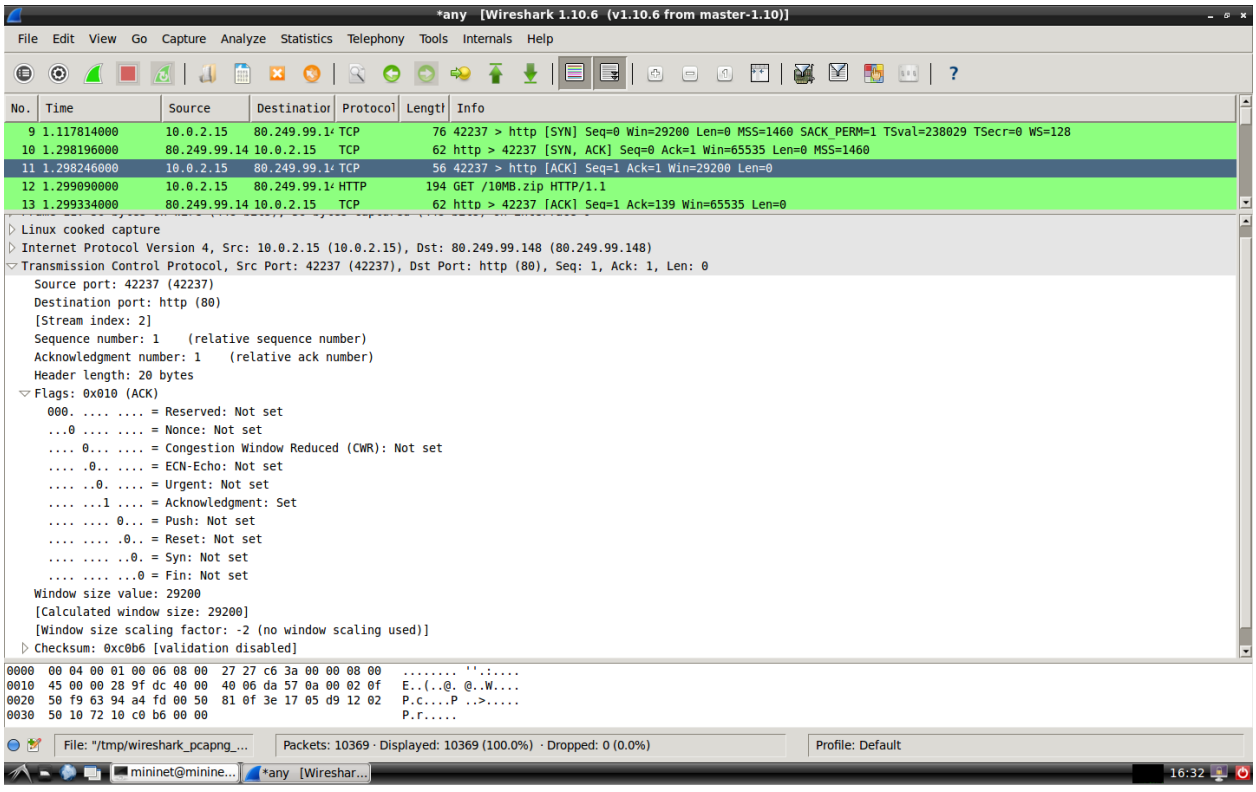
Open a terminal window. Using wget, download the file http://ipv4.download.thinkbroadband.com/10MB.zip

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?

According to the screenshots below, the initial window size that my computer advertised to the server was 29200 bits. The initial window size that the server advertised to my computer was 65535 bits. Since my window size is less than the server's window size, my window size is the bottleneck in the connection.

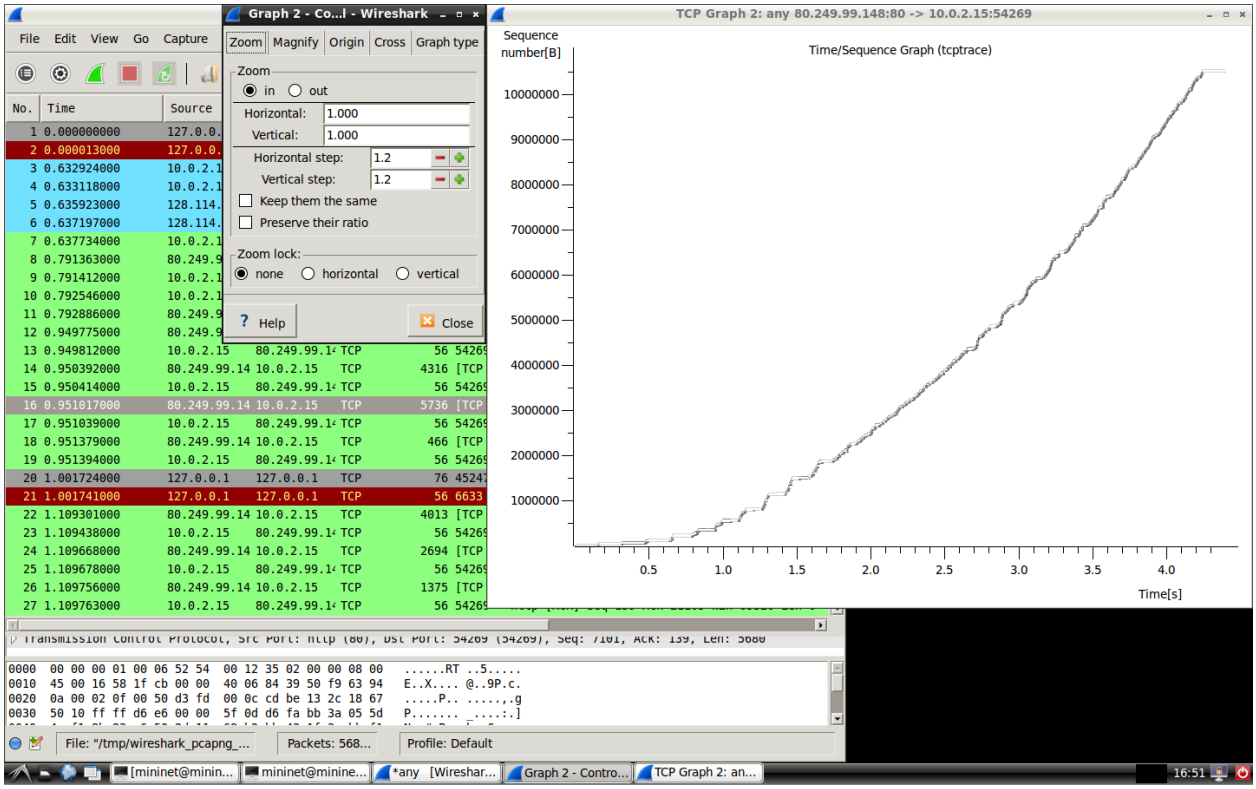



*any [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
No. Time Source Destination Protocol Length Info
9 1.117814000 10.0.2.15 80.249.99.14 TCP 76 42237 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=238029 TSecr=0 WS=128
10 1.298196000 80.249.99.14 10.0.2.15 TCP 62 http > 42237 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
11 1.298246000 10.0.2.15 80.249.99.14 TCP 56 42237 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
12 1.299090000 10.0.2.15 80.249.99.14 HTTP 194 GET /10MB.zip HTTP/1.1
13 1.299334000 80.249.99.14 10.0.2.15 TCP 62 http > 42237 [ACK] Seq=1 Ack=139 Win=65535 Len=0
Transmission Control Protocol, Src Port: http (80), Dst Port: 42237 (42237), Seq: 0, Ack: 1, Len: 0
Source port: http (80)
Destination port: 42237 (42237)
[Stream index: 2]
Sequence number: 0 (relative sequence number)
Acknowledgment number: 1 (relative ack number)
Header length: 24 bytes
Flags: 0x012 (SYN, ACK)
000. .... = Reserved: Not set
...0 .... = Nonce: Not set
.... 0... .... = Congestion Window Reduced (CWR): Not set
.... .0.. .... = ECN-Echo: Not set
.... ..0. .... = Urgent: Not set
....1 .... = Acknowledgment: Set
.... 0... = Push: Not set
.... .0.. = Reset: Not set
....1. = Syn: Set
....0 = Fin: Not set
Window size value: 65535
[Calculated window size: 65535]
Checksum: 0x5b2c [validation disabled]
Options: (4 bytes), Maximum segment size
[SEQ/ACK analysis]
VSS-Monitoring ethernet trailer, Source Port: 0
0000 00 00 00 01 00 06 52 54 00 12 35 02 00 00 08 00 ......RT ..5.....
0010 45 00 00 2c 08 cc 00 00 40 06 b1 64 50 f9 63 94 E..,.... @..dP.c.
0020 0a 00 02 0f 00 50 a4 fd 05 d9 12 01 81 0f 3e 17 .....P.. ......>.
0030 60 12 ff ff 5b 2c 00 00 02 04 05 b4 00 00 `...[,.. ......
File: "/tmp/wireshark_pcapng_... Packets: 10369 · Displayed: 10369 (100.0%) · Dropped: 0 (0.0%) Profile: Default
mininet@minine... *any [Wireshar... 16:32


*any [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
No. Time Source Destination Protocol Length Info
9 1.117814000 10.0.2.15 80.249.99.14 TCP 76 42237 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=238029 TSecr=0 WS=128
10 1.298196000 80.249.99.14 10.0.2.15 TCP 62 http > 42237 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
11 1.298246000 10.0.2.15 80.249.99.14 TCP 56 42237 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
12 1.299090000 10.0.2.15 80.249.99.14 HTTP 194 GET /10MB.zip HTTP/1.1
13 1.299334000 80.249.99.14 10.0.2.15 TCP 62 http > 42237 [ACK] Seq=1 Ack=139 Win=65535 Len=0
Linux cooked capture
Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 80.249.99.148 (80.249.99.148)
Transmission Control Protocol, Src Port: 42237 (42237), Dst Port: http (80), Seq: 1, Ack: 1, Len: 0
Source port: 42237 (42237)
Destination port: http (80)
[Stream index: 2]
Sequence number: 1 (relative sequence number)
Acknowledgment number: 1 (relative ack number)
Header length: 20 bytes
Flags: 0x010 (ACK)
000. .... = Reserved: Not set
...0 .... = Nonce: Not set
.... 0... .... = Congestion Window Reduced (CWR): Not set
.... .0.. .... = ECN-Echo: Not set
.... ..0. .... = Urgent: Not set
....1 .... = Acknowledgment: Set
.... 0... = Push: Not set
.... .0.. = Reset: Not set
....0. = Syn: Not set
....0 = Fin: Not set
Window size value: 29200
[Calculated window size: 29200]
[Window size scaling factor: -2 (no window scaling used)]
Checksum: 0xc0b6 [validation disabled]
0000 00 04 00 01 00 06 08 00 27 27 c6 3a 00 00 08 00 .... ''.:....
0010 45 00 00 28 9f dc 40 00 40 06 da 57 0a 00 02 0f E..(..@. @..W....
0020 50 f9 63 94 a4 fd 00 50 81 0f 3e 17 05 d9 12 02 P.c....P ..>.....
0030 50 10 72 10 c0 b6 00 00 P.r.....
File: "/tmp/wireshark_pcapng_... Packets: 10369 · Displayed: 10369 (100.0%) · Dropped: 0 (0.0%) Profile: Default
mininet@minine... *any [Wireshar... 16:32

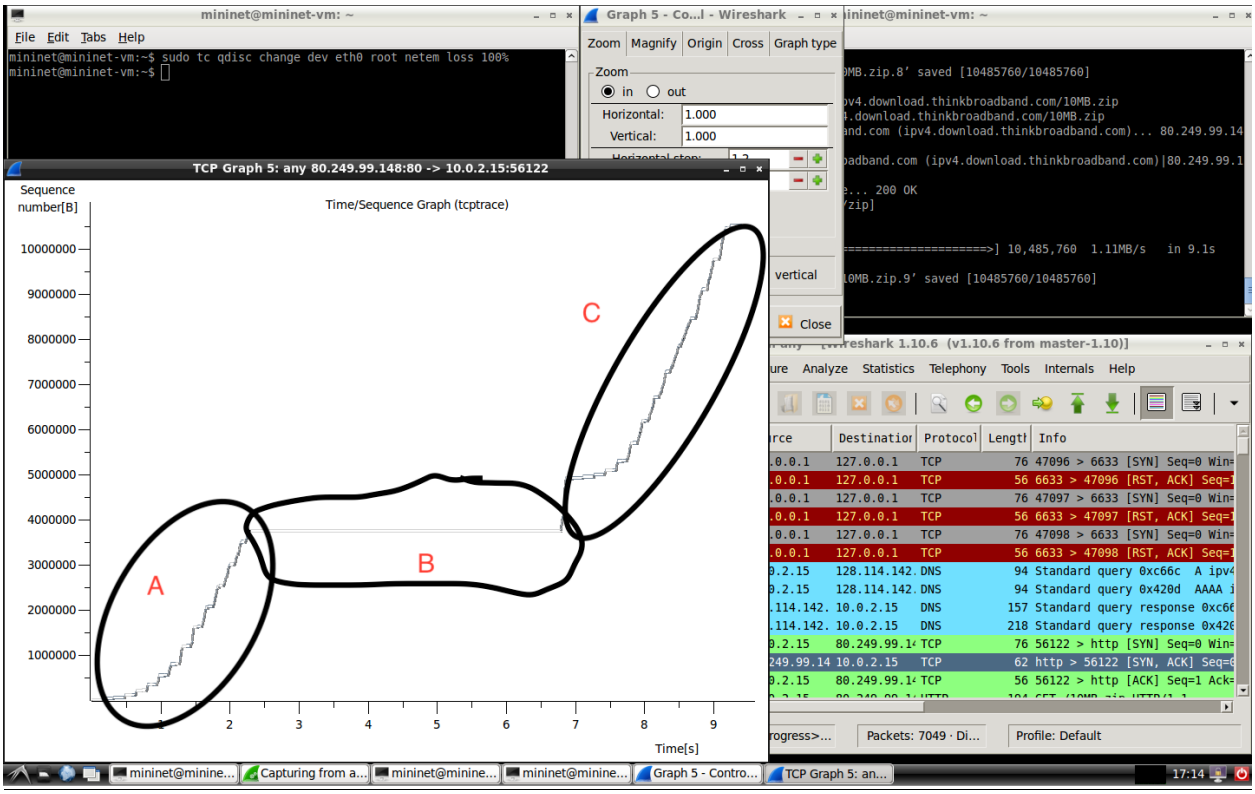
12. (10) Find a packet from the download whose source address is the server's address and the destination address is your computer's address. Using Wireshark, create a tcptrace graph with this packet selected. Take a screenshot of the graph and explain what it is showing. Look into the Wireshark documentation if you need assistance making this graph.

The graph shows that as time increases, we get more and more packets. The amount of packets that we get exponentially increases.



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.

Section A: This is the slow start and it is also where there is 0% loss.
Section B: This is where we get 100% loss and where there are no packets transmitted/received.
Section C: This is where we get back to 0% loss and we begin receiving packets again.
In my graph, I do not have congestion avoidance shown.



Works Cited

https://www.w3.org/Protocols/rfc2616/rfc2616-sec5.html

http://packetbomb.com/understanding-the-tcptrace-time-sequence-graph-in-wireshark/

https://wiki.linuxfoundation.org/networking/netem

https://support.managed.com/kb/a2312/how-to-use-nslookup-to-view-your-dns-records.aspx

https://virtualizationmaximus.wordpress.com/2013/01/09/authoritative-dns-server-name-server-dns-protocol-flags-authoritative-answer-and-total-authority-rrs-practical-simulation-insideout/

https://serverfault.com/questions/422288/what-is-the-difference-between-authoritative-nameserver-and-recursive-resolver

http://packetbomb.com/understanding-the-tcptrace-time-sequence-graph-in-wireshark/