Antoine Rocha

arocha4@ucsc.edu

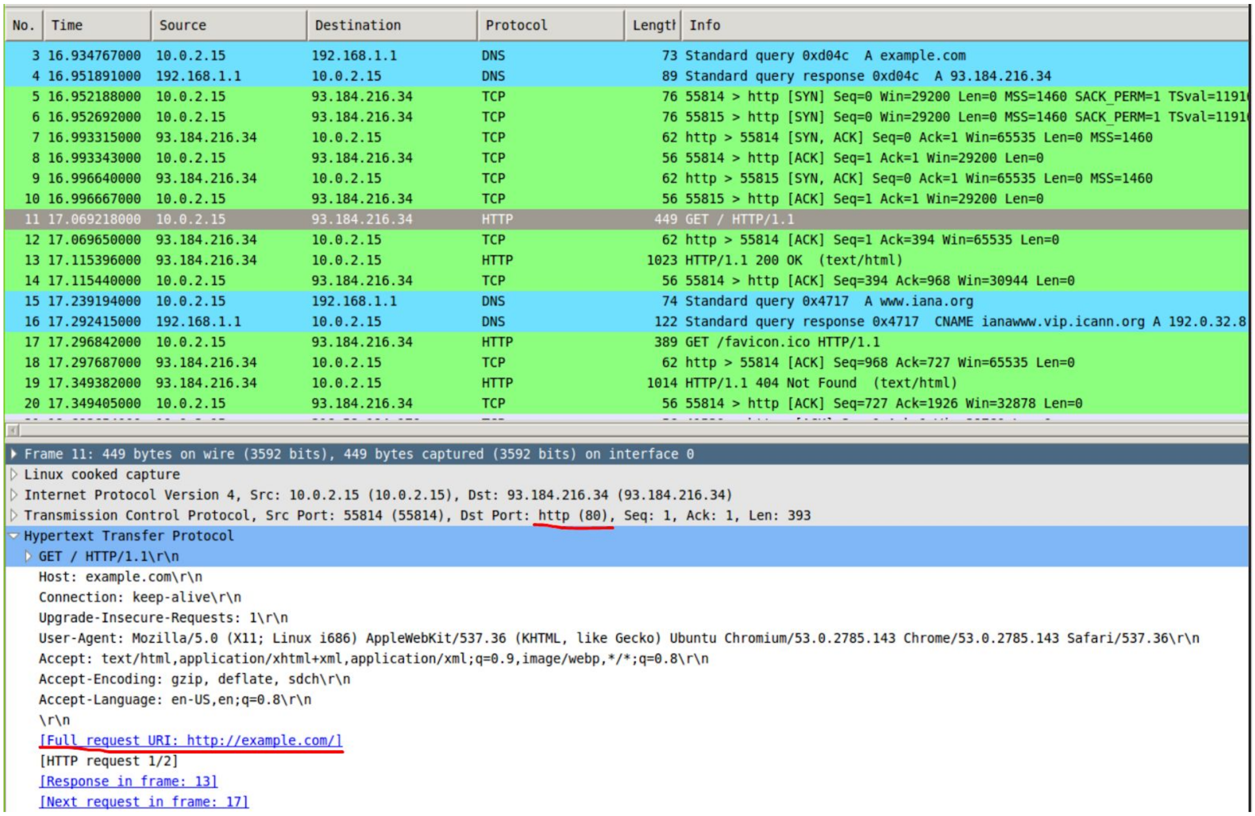
Ce 150 lab2

Lab 2

Part 1: HTTP

1. *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.*

   Once there is an established connection between the client and server after DNS obtained the IP address of the destination server a GET request is sent out to (well known port 80) unto the full request URI (http://example.com).

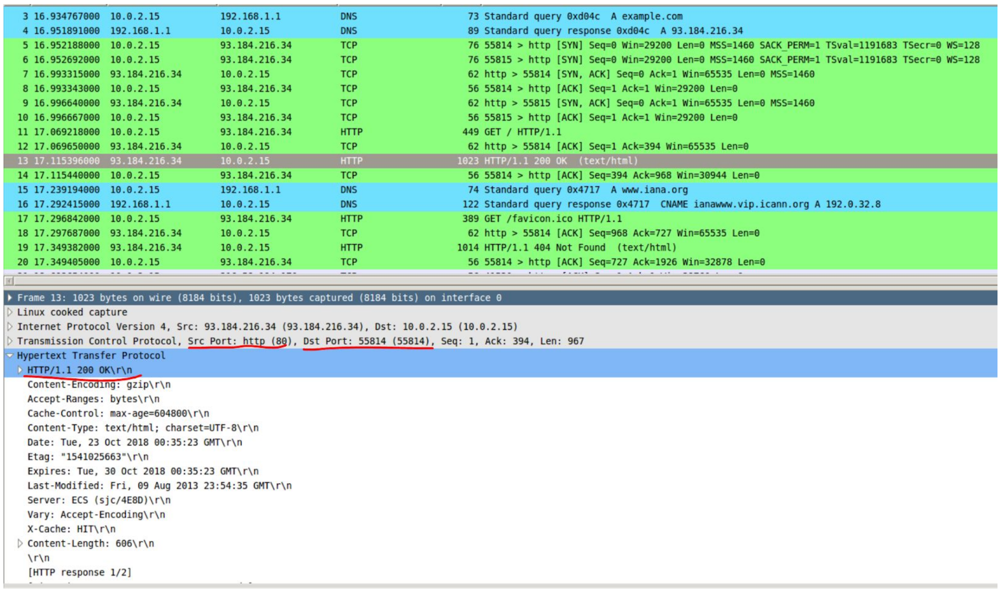


No.	Time	Source	Destination	Protocol	Lengtl	Info
3	16.934767000	10.0.2.15	192.168.1.1	DNS	73	Standard query 0xd04c A example.com
4	16.951891000	192.168.1.1	10.0.2.15	DNS	89	Standard query response 0xd04c A 93.184.216.34
5	16.952188000	10.0.2.15	93.184.216.34	TCP	76	55814 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1191
6	16.952692000	10.0.2.15	93.184.216.34	TCP	76	55815 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1191
7	16.993315000	93.184.216.34	10.0.2.15	TCP	62	http > 55814 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
8	16.993343000	10.0.2.15	93.184.216.34	TCP	56	55814 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
9	16.996640000	93.184.216.34	10.0.2.15	TCP	62	http > 55815 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
10	16.996667000	10.0.2.15	93.184.216.34	TCP	56	55815 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
11	17.069218000	10.0.2.15	93.184.216.34	HTTP	449	GET / HTTP/1.1
12	17.069650000	93.184.216.34	10.0.2.15	TCP	62	http > 55814 [ACK] Seq=1 Ack=394 Win=65535 Len=0
13	17.115396000	93.184.216.34	10.0.2.15	HTTP	1023	HTTP/1.1 200 OK (text/html)
14	17.115440000	10.0.2.15	93.184.216.34	TCP	56	55814 > http [ACK] Seq=394 Ack=968 Win=30944 Len=0
15	17.239194000	10.0.2.15	192.168.1.1	DNS	74	Standard query 0x4717 A www.iana.org
16	17.292415000	192.168.1.1	10.0.2.15	DNS	122	Standard query response 0x4717 CNAME ianawww.vip.icann.org A 192.0.32.8
17	17.296842000	10.0.2.15	93.184.216.34	HTTP	389	GET /favicon.ico HTTP/1.1
18	17.297687000	93.184.216.34	10.0.2.15	TCP	62	http > 55814 [ACK] Seq=968 Ack=727 Win=65535 Len=0
19	17.349382000	93.184.216.34	10.0.2.15	HTTP	1014	HTTP/1.1 404 Not Found (text/html)
20	17.349405000	10.0.2.15	93.184.216.34	TCP	56	55814 > http [ACK] Seq=727 Ack=1926 Win=32878 Len=0

```
▸ Frame 11: 449 bytes on wire (3592 bits), 449 bytes captured (3592 bits) on interface 0
▷ Linux cooked capture
▷ Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 93.184.216.34 (93.184.216.34)
▷ Transmission Control Protocol, Src Port: 55814 (55814), Dst Port: http (80), Seq: 1, Ack: 1, Len: 393
▽ Hypertext Transfer Protocol
  ▷ GET / HTTP/1.1\r\n
    Host: example.com\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://example.com/]
    [HTTP request 1/2]
    [Response in frame: 13]
    [Next request in frame: 17]
```

2. *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.*

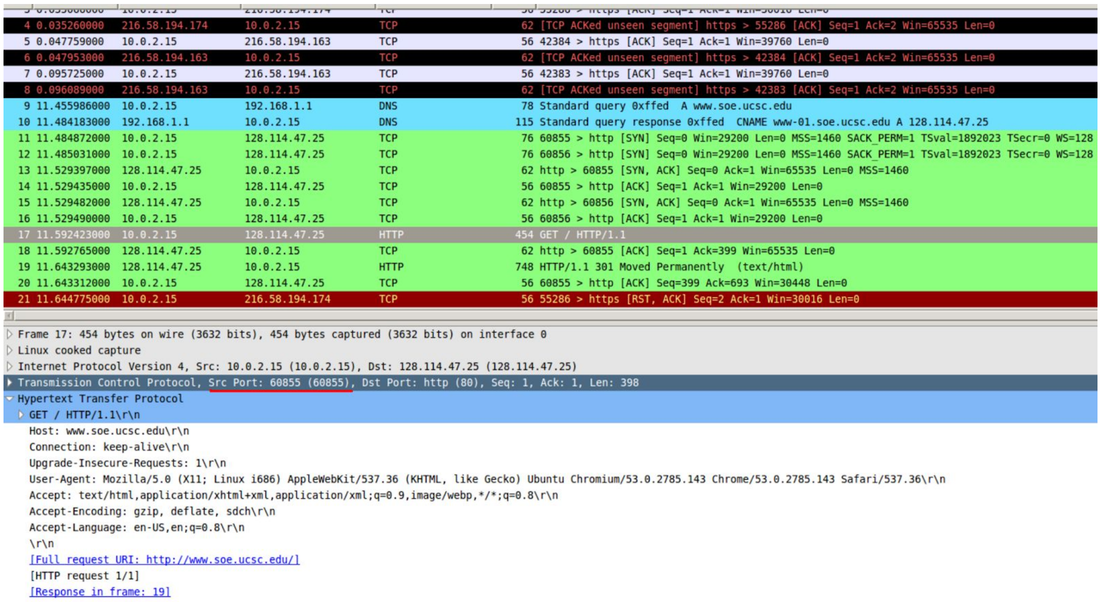
   Once the connection has been established the server's response to the clients GET URI(http://example.com)) request is HTTP status code 200 OKis a success acknowledgement of the GET URI request. The content type of the response is a text/html.



```
  3 16.934767000  10.0.2.15       192.168.1.1     DNS          73 Standard query 0xd04c  A example.com
  4 16.951891000  192.168.1.1     10.0.2.15       DNS          89 Standard query response 0xd04c  A 93.184.216.34
  5 16.952188000  10.0.2.15       93.184.216.34   TCP          76 55814 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1191683 TSecr=0 WS=128
  6 16.952692000  10.0.2.15       93.184.216.34   TCP          76 55815 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1191683 TSecr=0 WS=128
  7 16.993315000  93.184.216.34   10.0.2.15       TCP          62 http > 55814 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
  8 16.993343000  10.0.2.15       93.184.216.34   TCP          56 55814 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
  9 16.996640000  93.184.216.34   10.0.2.15       TCP          62 http > 55815 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
 10 16.996667000  10.0.2.15       93.184.216.34   TCP          56 55815 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
 11 17.069218000  10.0.2.15       93.184.216.34   HTTP        449 GET / HTTP/1.1
 12 17.069650000  93.184.216.34   10.0.2.15       TCP          62 http > 55814 [ACK] Seq=1 Ack=394 Win=65535 Len=0
 13 17.115396000  93.184.216.34   10.0.2.15       HTTP       1023 HTTP/1.1 200 OK  (text/html)
 14 17.115440000  10.0.2.15       93.184.216.34   TCP          56 55814 > http [ACK] Seq=394 Ack=968 Win=30944 Len=0
 15 17.239194000  10.0.2.15       192.168.1.1     DNS          74 Standard query 0x4717  A www.iana.org
 16 17.292415000  192.168.1.1     10.0.2.15       DNS         122 Standard query response 0x4717  CNAME ianawww.vip.icann.org A 192.0.32.8
 17 17.296842000  10.0.2.15       93.184.216.34   HTTP        389 GET /favicon.ico HTTP/1.1
 18 17.297687000  93.184.216.34   10.0.2.15       TCP          62 http > 55814 [ACK] Seq=968 Ack=727 Win=65535 Len=0
 19 17.349382000  93.184.216.34   10.0.2.15       HTTP       1014 HTTP/1.1 404 Not Found  (text/html)
 20 17.349405000  10.0.2.15       93.184.216.34   TCP          56 55814 > http [ACK] Seq=727 Ack=1926 Win=32878 Len=0

▸ Frame 13: 1023 bytes on wire (8184 bits), 1023 bytes captured (8184 bits) on interface 0
▷ Linux cooked capture
▷ Internet Protocol Version 4, Src: 93.184.216.34 (93.184.216.34), Dst: 10.0.2.15 (10.0.2.15)
▷ Transmission Control Protocol, Src Port: http (80), Dst Port: 55814 (55814), Seq: 1, Ack: 394, Len: 967
▿ Hypertext Transfer Protocol
  ▷ HTTP/1.1 200 OK\r\n
    Content-Encoding: gzip\r\n
    Accept-Ranges: bytes\r\n
    Cache-Control: max-age=604800\r\n
    Content-Type: text/html; charset=UTF-8\r\n
    Date: Tue, 23 Oct 2018 00:35:23 GMT\r\n
    Etag: "1541025663"\r\n
    Expires: Tue, 30 Oct 2018 00:35:23 GMT\r\n
    Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT\r\n
    Server: ECS (sjc/4E8D)\r\n
    Vary: Accept-Encoding\r\n
    X-Cache: HIT\r\n
  ▷ Content-Length: 606\r\n
    \r\n
    [HTTP response 1/2]
```

3. *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.*

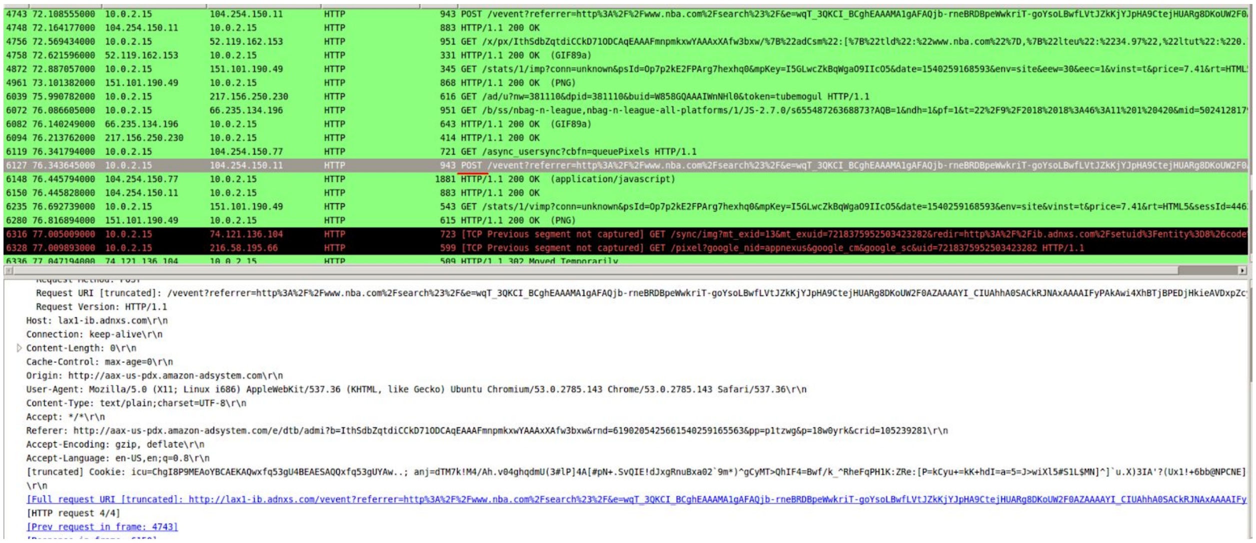
   I noticed that that everything was identical form the previous request except the port numbers. The previous port clients port for example.com was 55814 while for the website in a new tab the port is 60855.



```
  4 0.035260000   216.58.194.174   10.0.2.15        TCP          62 [TCP ACKed unseen segment] https > 55286 [ACK] Seq=1 Ack=2 Win=65535 Len=0
  5 0.047759000   10.0.2.15        216.58.194.163   TCP          56 42384 > https [ACK] Seq=1 Ack=1 Win=39760 Len=0
  6 0.047953000   216.58.194.163   10.0.2.15        TCP          62 [TCP ACKed unseen segment] https > 42384 [ACK] Seq=1 Ack=2 Win=65535 Len=0
  7 0.095725000   10.0.2.15        216.58.194.163   TCP          56 42383 > https [ACK] Seq=1 Ack=1 Win=39760 Len=0
  8 0.096089000   216.58.194.163   10.0.2.15        TCP          62 [TCP ACKed unseen segment] https > 42383 [ACK] Seq=1 Ack=2 Win=65535 Len=0
  9 11.455986000  10.0.2.15        192.168.1.1      DNS          78 Standard query 0xffed  A www.soe.ucsc.edu
 10 11.484183000  192.168.1.1      10.0.2.15        DNS         115 Standard query response 0xffed  CNAME www-01.soe.ucsc.edu A 128.114.47.25
 11 11.484872000  10.0.2.15        128.114.47.25    TCP          76 60855 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1892023 TSecr=0 WS=128
 12 11.485031000  10.0.2.15        128.114.47.25    TCP          76 60856 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1892023 TSecr=0 WS=128
 13 11.529397000  128.114.47.25    10.0.2.15        TCP          62 http > 60855 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
 14 11.529435000  10.0.2.15        128.114.47.25    TCP          56 60855 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
 15 11.529482000  128.114.47.25    10.0.2.15        TCP          62 http > 60856 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
 16 11.529490000  10.0.2.15        128.114.47.25    TCP          56 60856 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
 17 11.592423000  10.0.2.15        128.114.47.25    HTTP        454 GET / HTTP/1.1
 18 11.592765000  128.114.47.25    10.0.2.15        TCP          62 http > 60855 [ACK] Seq=1 Ack=399 Win=65535 Len=0
 19 11.643293000  128.114.47.25    10.0.2.15        HTTP        748 HTTP/1.1 301 Moved Permanently  (text/html)
 20 11.643312000  10.0.2.15        128.114.47.25    TCP          56 60855 > http [ACK] Seq=399 Ack=693 Win=30448 Len=0
 21 11.644775000  10.0.2.15        216.58.194.174   TCP          56 55286 > https [RST, ACK] Seq=2 Ack=1 Win=30016 Len=0

▷ Frame 17: 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: 60855 (60855), Dst Port: http (80), Seq: 1, Ack: 1, Len: 398
▿ Hypertext Transfer Protocol
  ▷ GET / HTTP/1.1\r\n
    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: 19]
```

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

   www.nba.com is not secured and does not use https. So I went to the search bar and typed the warriors and used the HTTP filter within wireshark and found a Post status code.



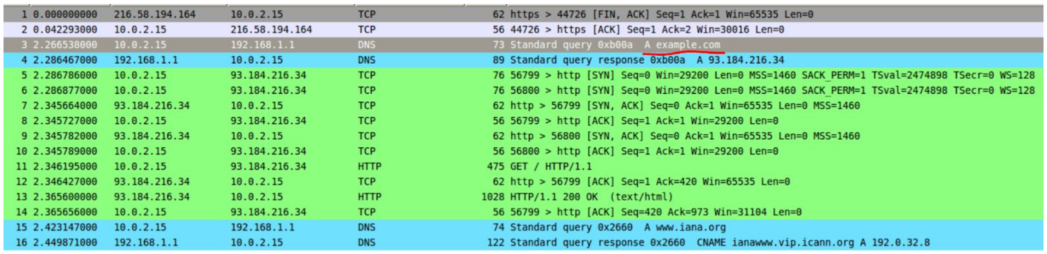
```
4743 72.108555000 10.0.2.15       104.254.150.11   HTTP       943 POST /vevent?referrer=http%3A%2F%2Fwww.nba.com%2Fsearch%23%2F&e=wqT_3QKCI_BCghEAAAMA1gAFAQjb-rneBRDBpeWwkriT-goYsoLBwfLVtJZkKjYJpHA9CtejHUARg8DKoUW2F0
4748 72.164177000 104.254.150.11  10.0.2.15        HTTP       883 HTTP/1.1 200 OK
4756 72.569434000 10.0.2.15       52.119.162.153   HTTP       951 GET /x/px/IthSdbZqtdiCCkD71ODCAqEAAAFmnpmkxwYAAAxXAfw3bxw/%7B%22adCsm%22:[%7B%22tld%22:%22www.nba.com%22%7D,%7B%22lteu%22:%2234.97%22,%22ltut%22:%220.
4758 72.621596000 52.119.162.153  10.0.2.15        HTTP       331 HTTP/1.1 200 OK  (GIF89a)
4872 72.887057000 10.0.2.15       151.101.190.49   HTTP       345 GET /stats/1/imp?conn=unknown&psId=Op7p2kE2FPArg7hexhq0&mpKey=I5GLwcZkBqWgaO9IIcO5&date=1540259168593&env=site&eew=30&eec=1&vinst=t&price=7.41&rt=HTML
4961 73.101382000 151.101.190.49  10.0.2.15        HTTP       868 HTTP/1.1 200 OK  (PNG)
6039 75.990782000 10.0.2.15       217.156.250.230  HTTP       616 GET /ad/u?nw=381110&dpid=381110&buid=W858GQAAAIWnNHl0&token=tubemogul HTTP/1.1
6072 76.086605000 10.0.2.15       66.235.134.196   HTTP       951 GET /b/ss/nbag-n-league,nbag-n-league-all-platforms/1/JS-2.7.0/s65548726368873?AQB=1&ndh=1&pf=1&t=22%2F9%2F2018%2018%3A46%3A11%201%20420&mid=502412817
6082 76.140249000 66.235.134.196  10.0.2.15        HTTP       643 HTTP/1.1 200 OK  (GIF89a)
6094 76.213762000 217.156.250.230 10.0.2.15        HTTP       414 HTTP/1.1 200 OK
6119 76.341794000 10.0.2.15       104.254.150.77   HTTP       721 GET /async_usersync?cbfn=queuePixels HTTP/1.1
6127 76.343645000 10.0.2.15       104.254.150.11   HTTP       943 POST /vevent?referrer=http%3A%2F%2Fwww.nba.com%2Fsearch%23%2F&e=wqT_3QKCI_BCghEAAAMA1gAFAQjb-rneBRDBpeWwkriT-goYsoLBwfLVtJZkKjYJpHA9CtejHUARg8DKoUW2F0
6148 76.445794000 104.254.150.77  10.0.2.15        HTTP      1881 HTTP/1.1 200 OK  (application/javascript)
6150 76.445828000 104.254.150.11  10.0.2.15        HTTP       883 HTTP/1.1 200 OK
6235 76.692739000 10.0.2.15       151.101.190.49   HTTP       543 GET /stats/1/vimp?conn=unknown&psId=Op7p2kE2FPArg7hexhq0&mpKey=I5GLwcZkBqWgaO9IIcO5&date=1540259168593&env=site&vinst=t&price=7.41&rt=HTML5&sessId=446
6280 76.816894000 151.101.190.49  10.0.2.15        HTTP       615 HTTP/1.1 200 OK  (PNG)
6316 77.005009000 10.0.2.15       74.121.136.104   HTTP       723 [TCP Previous segment not captured] GET /sync/img?mt_exid=13&mt_exuid=7218375952503423282&redir=http%3A%2F%2Fib.adnxs.com%2Fsetuid%3Fentity%3D8%26code
6328 77.009893000 10.0.2.15       216.58.195.66    HTTP       599 [TCP Previous segment not captured] GET /pixel?google_nid=appnexus&google_cm&google_sc&uid=7218375952503423282 HTTP/1.1
6336 77.047194000 74.121.136.104  10.0.2.15        HTTP       509 HTTP/1.1 302 Moved Temporarily
```

```
      Request Method: POST
      Request URI [truncated]: /vevent?referrer=http%3A%2F%2Fwww.nba.com%2Fsearch%23%2F&e=wqT_3QKCI_BCghEAAAMA1gAFAQjb-rneBRDBpeWwkriT-goYsoLBwfLVtJZkKjYJpHA9CtejHUARg8DKoUW2F0AZAAAAYI_CIUAhhA0SACkRJNAxAAAAIFyPAkAwi4XhBTjBPEDjHkieAVDxpZc
      Request Version: HTTP/1.1
    Host: lax1-ib.adnxs.com\r\n
    Connection: keep-alive\r\n
  ▷ Content-Length: 0\r\n
    Cache-Control: max-age=0\r\n
    Origin: http://aax-us-pdx.amazon-adsystem.com\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
    Content-Type: text/plain;charset=UTF-8\r\n
    Accept: */*\r\n
    Referer: http://aax-us-pdx.amazon-adsystem.com/e/dtb/admi?b=IthSdbZqtdiCCkD71ODCAqEAAAFmnpmkxwYAAAxXAfw3bxw&rnd=6190205425661540259165563&pp=p1tzwg&p=18w0yrk&crid=105239281\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.8\r\n
    [truncated] Cookie: icu=ChgI8P9MEAoYBCAEKAQwxfq53gU4BEAESAQQxfq53gUYAw..; anj=dTM7k!M4/Ah.v04ghqdmU(3#lP]4A[#pN+.SvQIE!dJxgRnuBxa02`9m*)^gCyMT>QhIF4=Bwf/k_^RheFqPH1K:ZRe:[P=kCyu+=kK+hdI=a=5=J>wiXl5#S1L$MN]^]`u.X)3IA'?(Ux1!+6bb@NPCNE]
    \r\n
    [Full request URI [truncated]: http://lax1-ib.adnxs.com/vevent?referrer=http%3A%2F%2Fwww.nba.com%2Fsearch%23%2F&e=wqT_3QKCI_BCghEAAAMA1gAFAQjb-rneBRDBpeWwkriT-goYsoLBwfLVtJZkKjYJpHA9CtejHUARg8DKoUW2F0AZAAAAYI_CIUAhhA0SACkRJNAxAAAAIFy
    [HTTP request 4/4]
    [Prev request in frame: 4743]
    [Response in frame: 6150]
```

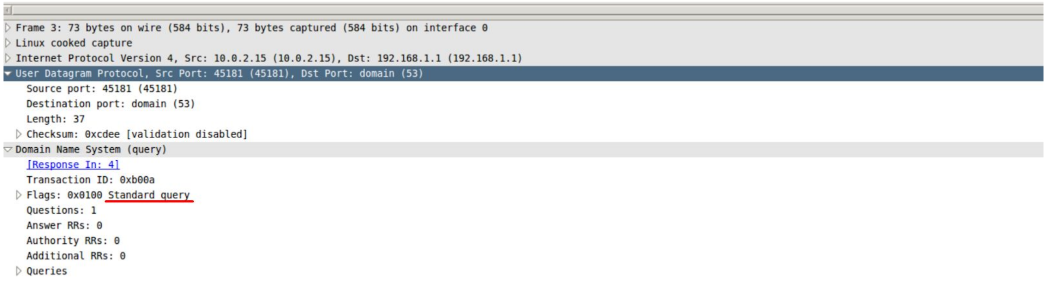
Part 2: DNS

5. *Open Chromium and navigate to www.example.com. 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.*

   Yes the steps to load the webpage are that the client requests the DNS for the destination ip address of example.com and the DNS finds a server that does have the IP address(via hops if the domain name isn't within the first server) then a connection is established (via three way handshake). Thus part 1 problem 1 takes over. I believe these are the correct packets. Hypothetically if I continued to refresh the webpage the server would save the webpage contents via cache and the wireshark segment would look different.



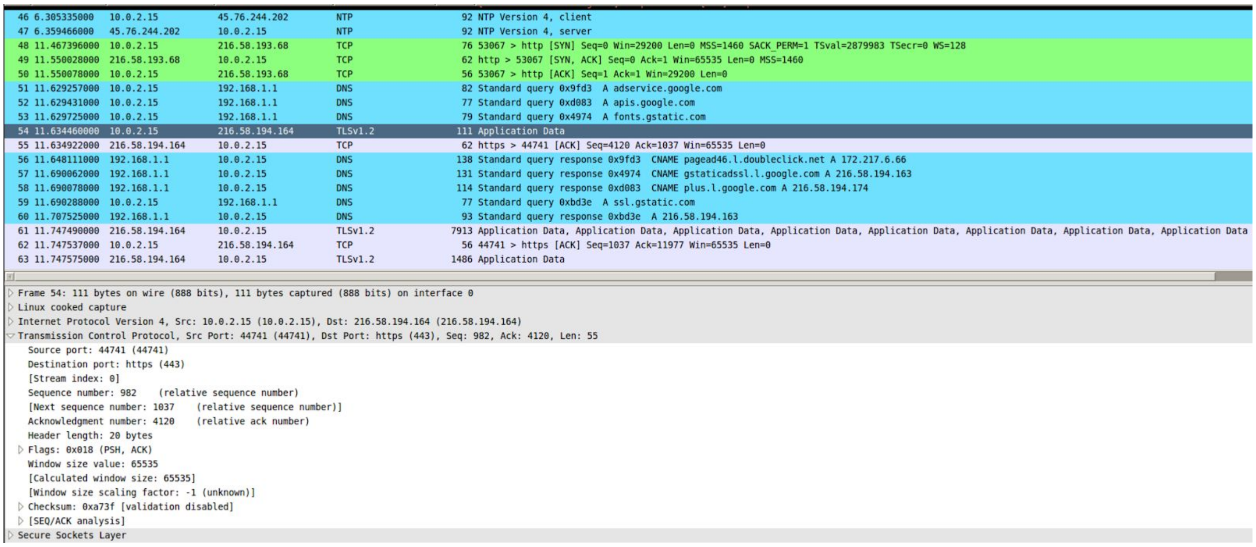
```
 1 0.000000000  216.58.194.164  10.0.2.15       TCP          62 https > 44726 [FIN, ACK] Seq=1 Ack=1 Win=65535 Len=0
 2 0.042293000  10.0.2.15       216.58.194.164  TCP          56 44726 > https [ACK] Seq=1 Ack=2 Win=30016 Len=0
 3 2.266538000  10.0.2.15       192.168.1.1     DNS          73 Standard query 0xb00a  A example.com
 4 2.286467000  192.168.1.1     10.0.2.15       DNS          89 Standard query response 0xb00a  A 93.184.216.34
 5 2.286786000  10.0.2.15       93.184.216.34   TCP          76 56799 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2474898 TSecr=0 WS=128
 6 2.286877000  10.0.2.15       93.184.216.34   TCP          76 56800 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2474898 TSecr=0 WS=128
 7 2.345664000  93.184.216.34   10.0.2.15       TCP          62 http > 56799 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
 8 2.345727000  10.0.2.15       93.184.216.34   TCP          56 56799 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
 9 2.345782000  93.184.216.34   10.0.2.15       TCP          62 http > 56800 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
10 2.345789000  10.0.2.15       93.184.216.34   TCP          56 56800 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
11 2.346195000  10.0.2.15       93.184.216.34   HTTP        475 GET / HTTP/1.1
12 2.346427000  93.184.216.34   10.0.2.15       TCP          62 http > 56799 [ACK] Seq=1 Ack=420 Win=65535 Len=0
13 2.365600000  93.184.216.34   10.0.2.15       HTTP       1028 HTTP/1.1 200 OK  (text/html)
14 2.365656000  10.0.2.15       93.184.216.34   TCP          56 56799 > http [ACK] Seq=420 Ack=973 Win=31104 Len=0
15 2.423147000  10.0.2.15       192.168.1.1     DNS          74 Standard query 0x2660  A www.iana.org
16 2.449871000  192.168.1.1     10.0.2.15       DNS         122 Standard query response 0x2660  CNAME ianawww.vip.icann.org A 192.0.32.8
```



```
▷ Frame 3: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface 0
▷ Linux cooked capture
▷ Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 192.168.1.1 (192.168.1.1)
▼ User Datagram Protocol, Src Port: 45181 (45181), Dst Port: domain (53)
    Source port: 45181 (45181)
    Destination port: domain (53)
    Length: 37
  ▷ Checksum: 0xcdee [validation disabled]
▽ Domain Name System (query)
    [Response In: 4]
    Transaction ID: 0xb00a
  ▷ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▷ Queries
```

6. *In Chromium, navigate to http://216.58.193.68. 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.*

   I believe these are the correct packets because a connection is established and we queried a DNS with an IP address and the info given on the right hand side shows the alias for google.com followed by application data.

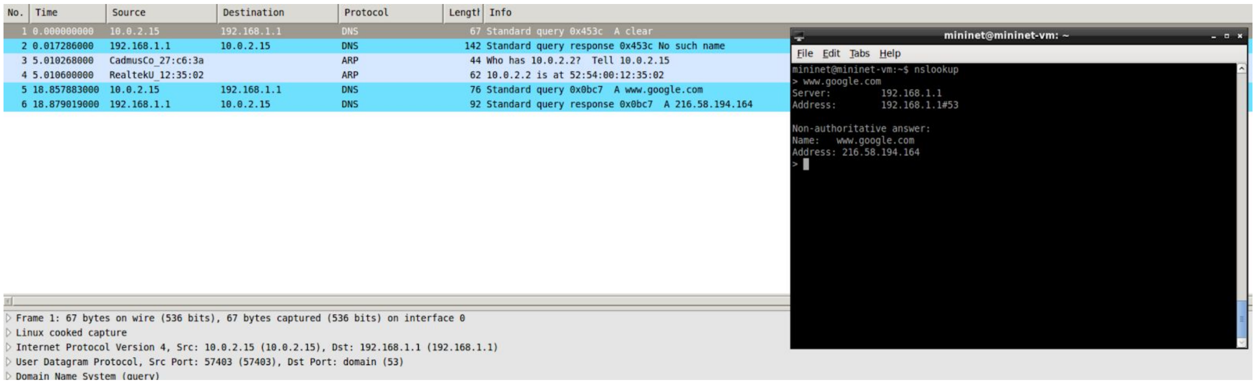


```
46 6.305335000   10.0.2.15       45.76.244.202    NTP        92 NTP Version 4, client
47 6.359466000   45.76.244.202   10.0.2.15        NTP        92 NTP Version 4, server
48 11.467396000  10.0.2.15       216.58.193.68    TCP        76 53067 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2879983 TSecr=0 WS=128
49 11.550028000  216.58.193.68   10.0.2.15        TCP        62 http > 53067 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
50 11.550078000  10.0.2.15       216.58.193.68    TCP        56 53067 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
51 11.629257000  10.0.2.15       192.168.1.1      DNS        82 Standard query 0x9fd3  A adservice.google.com
52 11.629431000  10.0.2.15       192.168.1.1      DNS        77 Standard query 0xd083  A apis.google.com
53 11.629725000  10.0.2.15       192.168.1.1      DNS        79 Standard query 0x4974  A fonts.gstatic.com
54 11.634460000  10.0.2.15       216.58.194.164   TLSv1.2   111 Application Data
55 11.634922000  216.58.194.164  10.0.2.15        TCP        62 https > 44741 [ACK] Seq=4120 Ack=1037 Win=65535 Len=0
56 11.648111000  192.168.1.1     10.0.2.15        DNS       138 Standard query response 0x9fd3  CNAME pagead46.l.doubleclick.net A 172.217.6.66
57 11.690062000  192.168.1.1     10.0.2.15        DNS       131 Standard query response 0x4974  CNAME gstaticadssl.l.google.com A 216.58.194.163
58 11.690078000  192.168.1.1     10.0.2.15        DNS       114 Standard query response 0xd083  CNAME plus.l.google.com A 216.58.194.174
59 11.690288000  10.0.2.15       192.168.1.1      DNS        77 Standard query 0xbd3e  A ssl.gstatic.com
60 11.707525000  192.168.1.1     10.0.2.15        DNS        93 Standard query response 0xbd3e  A 216.58.194.163
61 11.747490000  216.58.194.164  10.0.2.15        TLSv1.2  7913 Application Data, Application Data, Application Data, Application Data, Application Data, Application Data, Application Data, Application Data
62 11.747537000  10.0.2.15       216.58.194.164   TCP        56 44741 > https [ACK] Seq=1037 Ack=11977 Win=65535 Len=0
63 11.747575000  216.58.194.164  10.0.2.15        TLSv1.2  1486 Application Data
```

```
▷ Frame 54: 111 bytes on wire (888 bits), 111 bytes captured (888 bits) on interface 0
▷ Linux cooked capture
▷ Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 216.58.194.164 (216.58.194.164)
▽ Transmission Control Protocol, Src Port: 44741 (44741), Dst Port: https (443), Seq: 982, Ack: 4120, Len: 55
    Source port: 44741 (44741)
    Destination port: https (443)
    [Stream index: 0]
    Sequence number: 982    (relative sequence number)
    [Next sequence number: 1037    (relative sequence number)]
    Acknowledgment number: 4120    (relative ack number)
    Header length: 20 bytes
  ▷ Flags: 0x018 (PSH, ACK)
    Window size value: 65535
    [Calculated window size: 65535]
    [Window size scaling factor: -1 (unknown)]
  ▷ Checksum: 0xa73f [validation disabled]
  ▷ [SEQ/ACK analysis]
▷ Secure Sockets Layer
```

7. *Open a terminal window. Using nslookup, find the A records for www.google.com. 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 I was given is 216.58.194.164



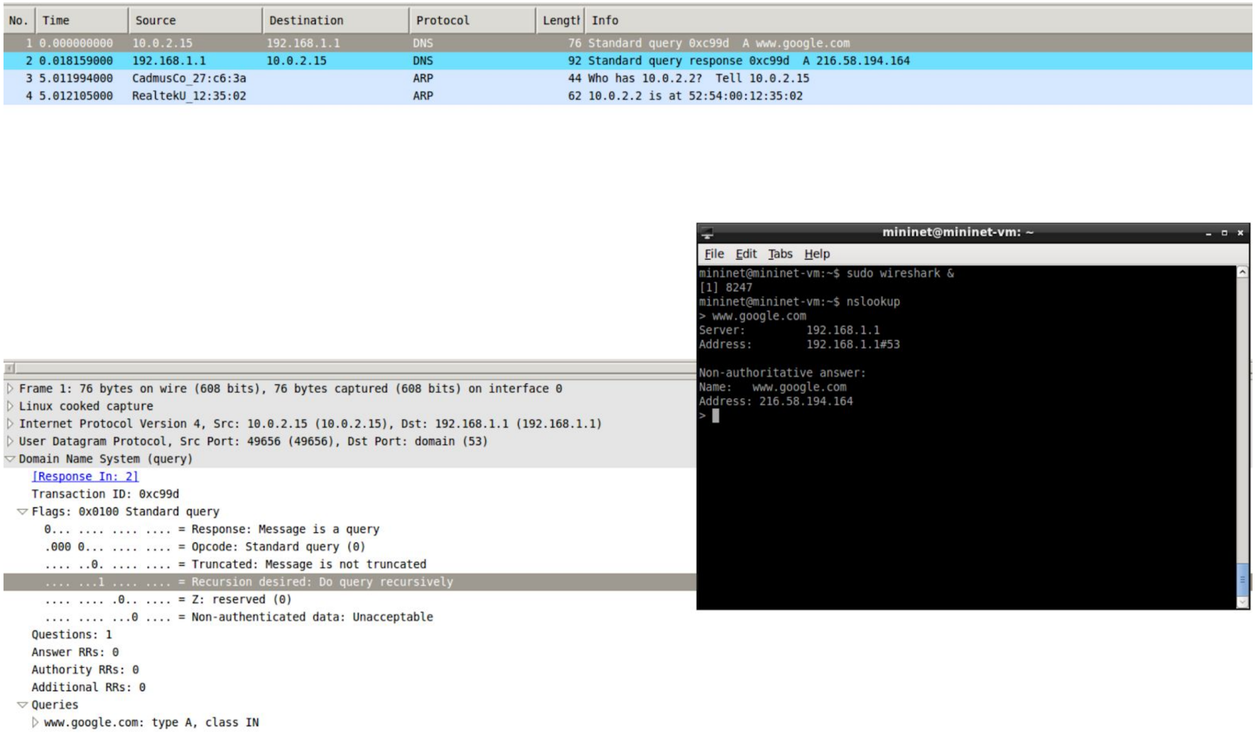
```
No. Time          Source            Destination   Protocol  Length Info
  1 0.000000000   10.0.2.15         192.168.1.1   DNS           67 Standard query 0x453c  A clear
  2 0.017286000   192.168.1.1       10.0.2.15     DNS          142 Standard query response 0x453c No such name
  3 5.010268000   CadmusCo_27:c6:3a               ARP           44 Who has 10.0.2.2?  Tell 10.0.2.15
  4 5.010600000   RealtekU_12:35:02               ARP           62 10.0.2.2 is at 52:54:00:12:35:02
  5 18.857883000  10.0.2.15         192.168.1.1   DNS           76 Standard query 0x0bc7  A www.google.com
  6 18.879019000  192.168.1.1       10.0.2.15     DNS           92 Standard query response 0x0bc7  A 216.58.194.164
```

```
▷ Frame 1: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface 0
▷ Linux cooked capture
▷ Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 192.168.1.1 (192.168.1.1)
▷ User Datagram Protocol, Src Port: 57403 (57403), Dst Port: domain (53)
▷ Domain Name System (query)
```

```
mininet@mininet-vm:~$ nslookup
> www.google.com
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
Name:   www.google.com
Address: 216.58.194.164
> 
```

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

   Yes the computer wanted to complete the request recursively. A (Non authoritative answer:) response given is by definition is a recursive query domain. A non-recursive queries are queries that our server is authoritative for. Given the response of a non-authoritative answer: our request is done recursively.



```
No.  Time          Source              Destination       Protocol    Length  Info
   1 0.000000000   10.0.2.15           192.168.1.1       DNS             76  Standard query 0xc99d  A www.google.com
   2 0.018159000   192.168.1.1         10.0.2.15         DNS             92  Standard query response 0xc99d  A 216.58.194.164
   3 5.011994000   CadmusCo_27:c6:3a                     ARP             44  Who has 10.0.2.2?  Tell 10.0.2.15
   4 5.012105000   RealtekU_12:35:02                     ARP             62  10.0.2.2 is at 52:54:00:12:35:02
```

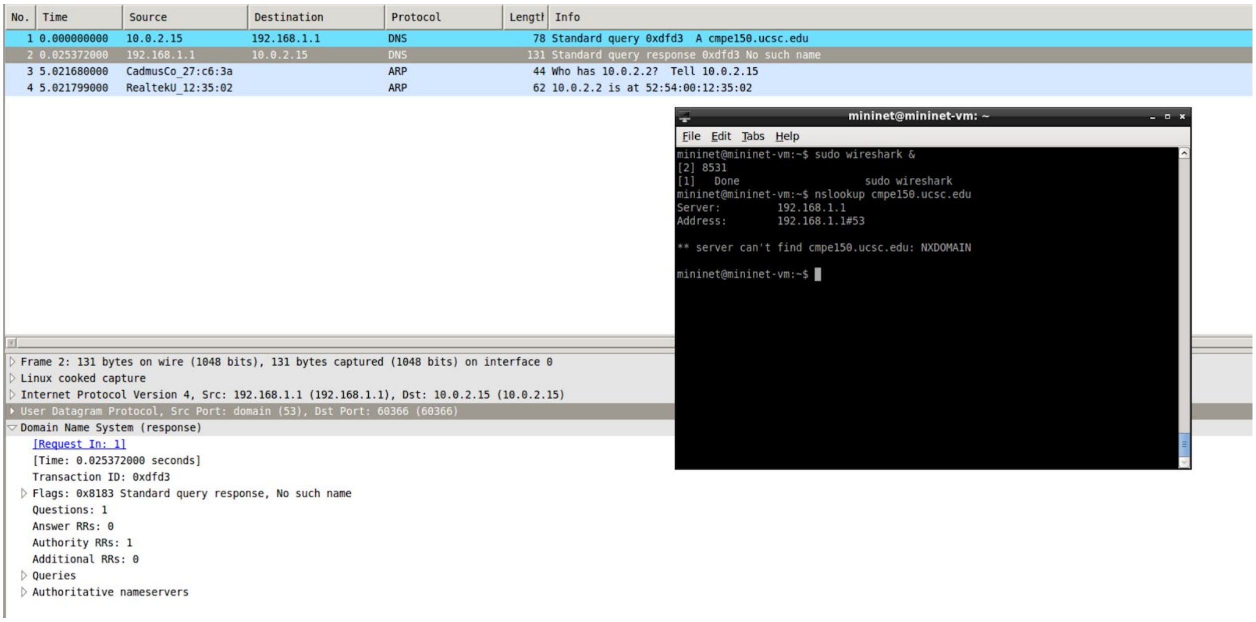
```
▷ Frame 1: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface 0
▷ Linux cooked capture
▷ Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 192.168.1.1 (192.168.1.1)
▷ User Datagram Protocol, Src Port: 49656 (49656), Dst Port: domain (53)
▽ Domain Name System (query)
    [Response In: 2]
    Transaction ID: 0xc99d
  ▽ Flags: 0x0100 Standard query
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▽ Queries
    ▷ www.google.com: type A, class IN
```

mininet@mininet-vm: ~

```
mininet@mininet-vm:~$ sudo wireshark &
[1] 8247
mininet@mininet-vm:~$ nslookup
> www.google.com
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
Name:   www.google.com
Address: 216.58.194.164
> 
```

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

The request was not resolved given that the message at the bottom said"server cant find cmpe150.ucsc.edu: NXDOMAIN" but the IP address given is 192.168.1.1#53 which is the our VM's DNS.



```
No.  Time          Source              Destination       Protocol    Length  Info
   1 0.000000000   10.0.2.15           192.168.1.1       DNS             78  Standard query 0xdfd3  A cmpe150.ucsc.edu
   2 0.025372000   192.168.1.1         10.0.2.15         DNS            131  Standard query response 0xdfd3 No such name
   3 5.021680000   CadmusCo_27:c6:3a                     ARP             44  Who has 10.0.2.2?  Tell 10.0.2.15
   4 5.021799000   RealtekU_12:35:02                     ARP             62  10.0.2.2 is at 52:54:00:12:35:02
```

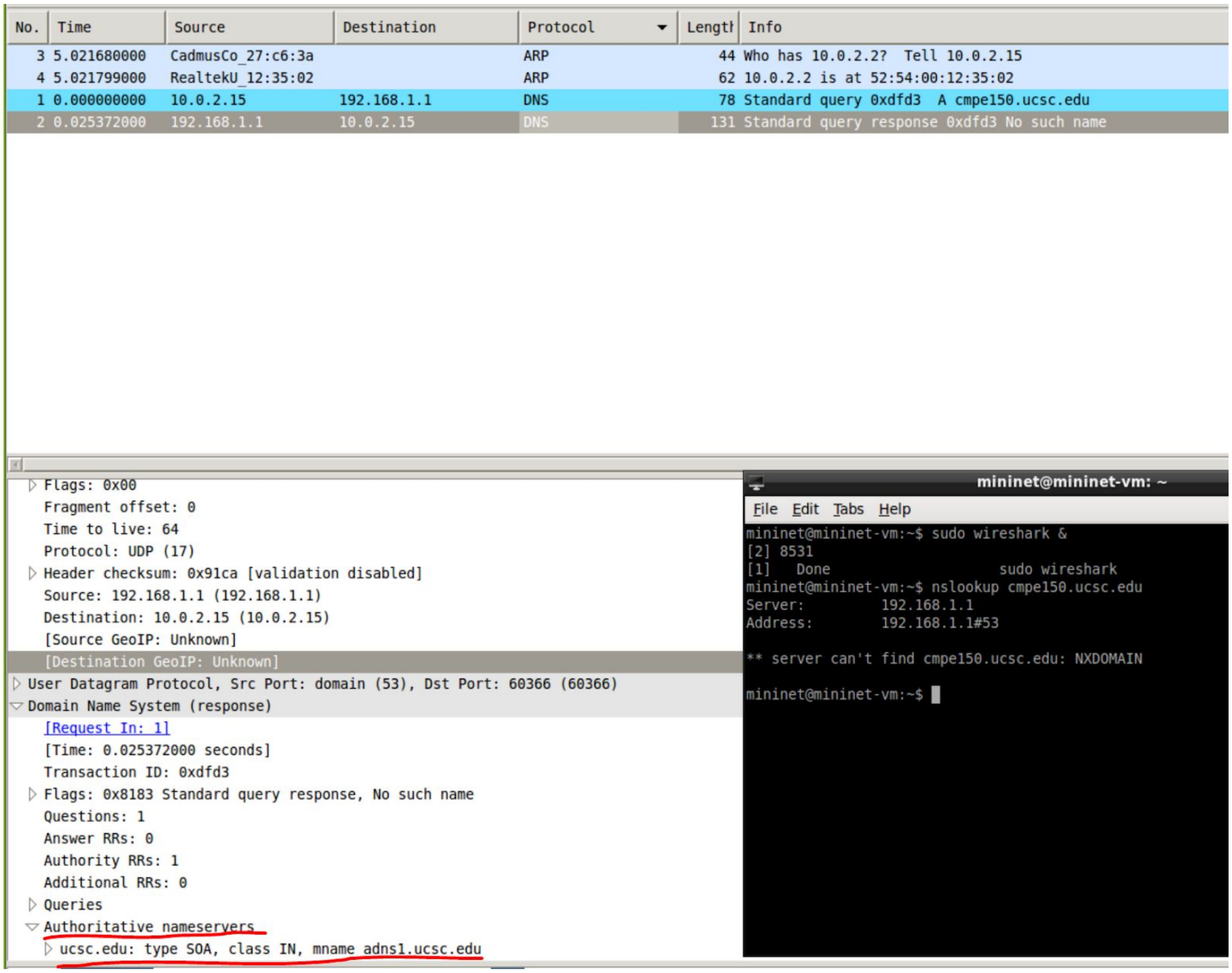
```
▷ Frame 2: 131 bytes on wire (1048 bits), 131 bytes captured (1048 bits) on interface 0
▷ Linux cooked capture
▷ Internet Protocol Version 4, Src: 192.168.1.1 (192.168.1.1), Dst: 10.0.2.15 (10.0.2.15)
▸ User Datagram Protocol, Src Port: domain (53), Dst Port: 60366 (60366)
▽ Domain Name System (response)
    [Request In: 1]
    [Time: 0.025372000 seconds]
    Transaction ID: 0xdfd3
  ▷ Flags: 0x8183 Standard query response, No such name
    Questions: 1
    Answer RRs: 0
    Authority RRs: 1
    Additional RRs: 0
  ▷ Queries
  ▷ Authoritative nameservers
```

mininet@mininet-vm: ~

```
mininet@mininet-vm:~$ sudo wireshark &
[2] 8531
[1]   Done                    sudo wireshark
mininet@mininet-vm:~$ nslookup cmpe150.ucsc.edu
Server:         192.168.1.1
Address:        192.168.1.1#53

** server can't find cmpe150.ucsc.edu: NXDOMAIN

mininet@mininet-vm:~$ 
```

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

The Authoritative name server for the ucsc.edu domain is adns1.ucsc.edu shown through wireshark Domain Name System(response) tab



```
No. Time         Source             Destination  Protocol  Lengtl Info
  3 5.021680000  CadmusCo_27:c6:3a               ARP          44 Who has 10.0.2.2?  Tell 10.0.2.15
  4 5.021799000  RealtekU_12:35:02               ARP          62 10.0.2.2 is at 52:54:00:12:35:02
  1 0.000000000  10.0.2.15          192.168.1.1  DNS          78 Standard query 0xdfd3  A cmpe150.ucsc.edu
  2 0.025372000  192.168.1.1        10.0.2.15    DNS         131 Standard query response 0xdfd3 No such name
```

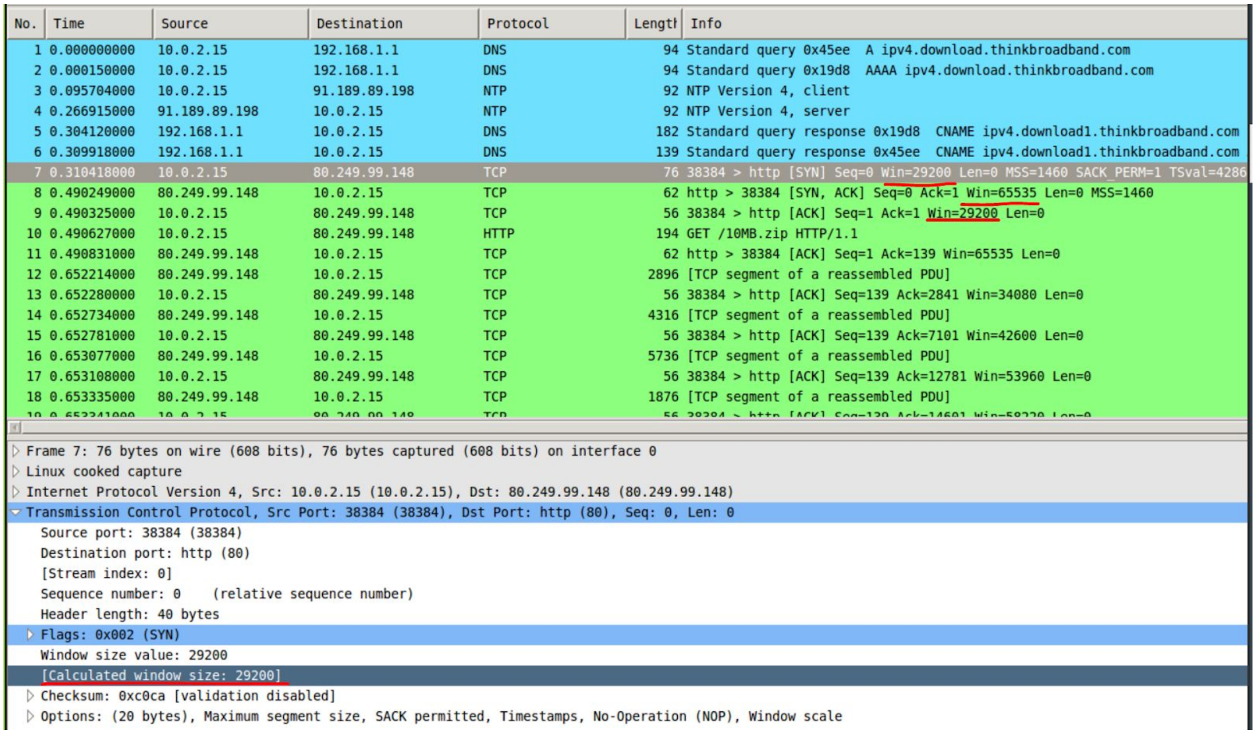
```
  ▷ Flags: 0x00
    Fragment offset: 0
    Time to live: 64
    Protocol: UDP (17)
  ▷ Header checksum: 0x91ca [validation disabled]
    Source: 192.168.1.1 (192.168.1.1)
    Destination: 10.0.2.15 (10.0.2.15)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
▷ User Datagram Protocol, Src Port: domain (53), Dst Port: 60366 (60366)
▽ Domain Name System (response)
    [Request In: 1]
    [Time: 0.025372000 seconds]
    Transaction ID: 0xdfd3
  ▷ Flags: 0x8183 Standard query response, No such name
    Questions: 1
    Answer RRs: 0
    Authority RRs: 1
    Additional RRs: 0
  ▷ Queries
  ▽ Authoritative nameservers
    ▷ ucsc.edu: type SOA, class IN, mname adns1.ucsc.edu
```

```
mininet@mininet-vm:~$ sudo wireshark &
[2] 8531
[1]   Done                    sudo wireshark
mininet@mininet-vm:~$ nslookup cmpe150.ucsc.edu
Server:         192.168.1.1
Address:        192.168.1.1#53

** server can't find cmpe150.ucsc.edu: NXDOMAIN

mininet@mininet-vm:~$ 
```

Part 3: TCP

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

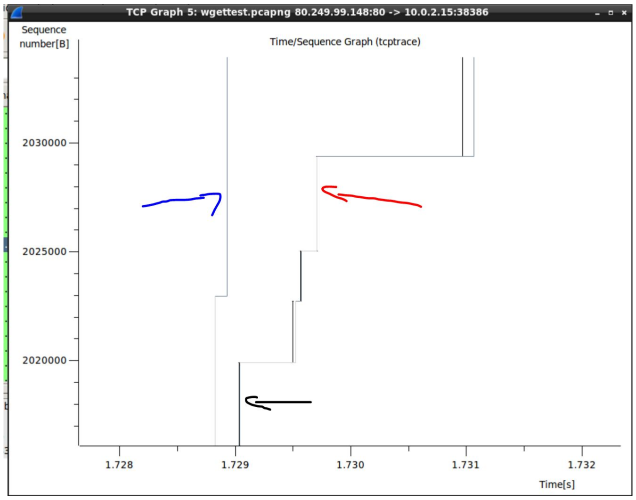
The three-way handshake connection shown below displays my computers advertised window size of 29200 and the servers advertised window size of 65535.



```
No.  Time         Source         Destination    Protocol  Lengtl Info
   1 0.000000000  10.0.2.15      192.168.1.1    DNS          94 Standard query 0x45ee  A ipv4.download.thinkbroadband.com
   2 0.000150000  10.0.2.15      192.168.1.1    DNS          94 Standard query 0x19d8  AAAA ipv4.download.thinkbroadband.com
   3 0.095704000  10.0.2.15      91.189.89.198  NTP          92 NTP Version 4, client
   4 0.266915000  91.189.89.198  10.0.2.15      NTP          92 NTP Version 4, server
   5 0.304120000  192.168.1.1    10.0.2.15      DNS         182 Standard query response 0x19d8  CNAME ipv4.download1.thinkbroadband.com
   6 0.309918000  192.168.1.1    10.0.2.15      DNS         139 Standard query response 0x45ee  CNAME ipv4.download1.thinkbroadband.com
   7 0.310418000  10.0.2.15      80.249.99.148  TCP          76 38384 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=4286
   8 0.490249000  80.249.99.148  10.0.2.15      TCP          62 http > 38384 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
   9 0.490325000  10.0.2.15      80.249.99.148  TCP          56 38384 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
  10 0.490627000  10.0.2.15      80.249.99.148  HTTP        194 GET /10MB.zip HTTP/1.1
  11 0.490831000  80.249.99.148  10.0.2.15      TCP          62 http > 38384 [ACK] Seq=1 Ack=139 Win=65535 Len=0
  12 0.652214000  80.249.99.148  10.0.2.15      TCP        2896 [TCP segment of a reassembled PDU]
  13 0.652280000  10.0.2.15      80.249.99.148  TCP          56 38384 > http [ACK] Seq=139 Ack=2841 Win=34080 Len=0
  14 0.652734000  80.249.99.148  10.0.2.15      TCP        4316 [TCP segment of a reassembled PDU]
  15 0.652781000  10.0.2.15      80.249.99.148  TCP          56 38384 > http [ACK] Seq=139 Ack=7101 Win=42600 Len=0
  16 0.653077000  80.249.99.148  10.0.2.15      TCP        5736 [TCP segment of a reassembled PDU]
  17 0.653108000  10.0.2.15      80.249.99.148  TCP          56 38384 > http [ACK] Seq=139 Ack=12781 Win=53960 Len=0
  18 0.653335000  80.249.99.148  10.0.2.15      TCP        1876 [TCP segment of a reassembled PDU]
  19 0.653341000  10.0.2.15      80.249.99.148  TCP          56 38384 > http [ACK] Seq=139 Ack=14601 Win=58320 Len=0

▷ Frame 7: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface 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: 38384 (38384), Dst Port: http (80), Seq: 0, Len: 0
    Source port: 38384 (38384)
    Destination port: http (80)
    [Stream index: 0]
    Sequence number: 0    (relative sequence number)
    Header length: 40 bytes
  ▷ Flags: 0x002 (SYN)
    Window size value: 29200
    [Calculated window size: 29200]
  ▷ Checksum: 0xc0ca [validation disabled]
  ▷ Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
```

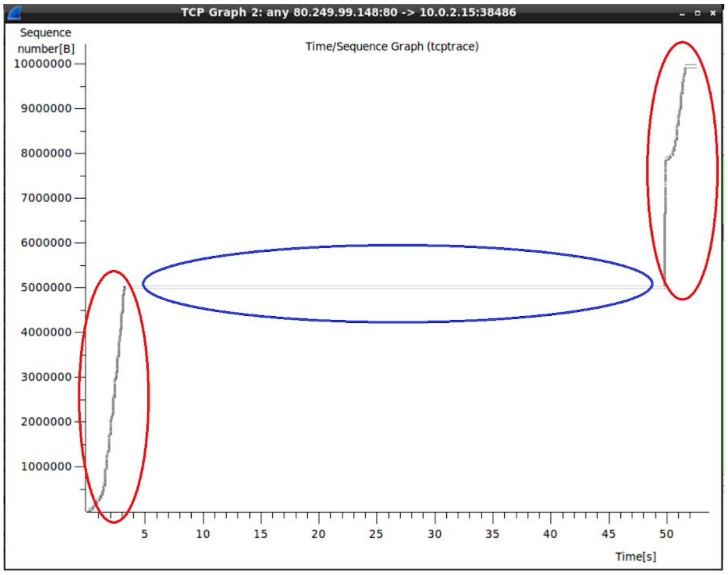
12. *Find a packet from the download whose source address is the server's address and the destination address is your computer's address. Create a tcptrace graph with this packet selected. Take a screenshot of the graph and explain what it is showing. Look into the Wireshark documentation if you need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

The segment in pointed by the Black arrow represents the segments sent.
The segment pointed by the Blue arrow tracks the receive window advertised from the other computer.
The segment pointed by the Red arrow keeps track of the ACK values received from the other endpoint.



13. *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.*

The red circles shown in the graph are areas where 0% loss shown
The blue circle in the graph are areas where 100% loss is shown
TCP slow start is shown in the lower left red circle from 1 - 2 seconds
congestion avoidance is shown in the top right red circle in the middle of the plateau


TCP Graph 2: any 80.249.99.148:80 -> 10.0.2.15:38486
Time/Sequence Graph (tcptrace)
Sequence number[B]
10000000
9000000
8000000
7000000
6000000
5000000
4000000
3000000
2000000
1000000
5
10
15
20
25
30
35
40
45
50
Time[s]