ALab 2

### Pre-Lab Questions

1. Choose 5 HTTP status codes and describe each one.
   1. “100” - Continue, This status code describes that the Initial part of the request been received

by the server and that the client can continue with the request. It is used when the server needs

to inform the client to continue sending the rest of the request.

“201” - Created, This status code means that the request has been fulfilled and resulted in one

or more new resources being made. The newly created resource(s) are typically included

in the body of the response, and the URI(s) to these resources are returned as the

entity of the response.

“204” - No Content, This status code means that the server has successfully processed

the request but is not returning any content in the response body. It's often used

when the client doesn't need any new information from the server but the server

may need to update some information on the current page.

“302” - Found, This status code, means that the requested resource is under a temporarily

under a different URI. The client should continue to use the Request-URI for more requests.

This response is only cacheable if it is asked by a Cache-Control or an Expire header field.

The temporary URI is given by the location field in the response.

“410” - Gone, This status code is a error response code which is that the requested

resource is no longer available on the server and has been permanently removed.

Unlike a 404 Not Found response, which indicates that the resource may be available in the future, 410 Gone implies that the resource is gone for good.

1. List the 8 HTTP 1.1 methods and explain what they do

* OPTIONS - The OPTIONS method describes the communication options available on the

request/response chain asked by the Request-URI. It returns the HTTP methods supported

by the server for a specific URL, allowing the client to understand what methods are

allowed for the resource.

* GET - The GET method requests data from a specified resource. It gets information from

the server without changing it. This is used when you are retrieving data, such as looking

at a web page.

* HEAD - The HEAD method is similar to the GET but in HEAD the server does not return a

message-body in the response. It only request only the headers of the response, not the

body.

* POST - The POST method is used to submit data to be processed to a specified resource. It

sends data to the server to create or update the resource. These request are used when

submitting form data or uploading files.

* PUT - The PUT method sends data to the server to create or update a resource at a specific

URL. It replaces the current representation of the target resource with the

request payload.

* DELETE - The DELETE method requests that the server deletes the resource specified

by the URL. It removes the resource identified by the URL from the server.

* TRACE - The TRACE method echoes the received request so that a client can see

what changes or additions have been made by the servers. It is mainly used for diagnostic

purposes.

* CONNECT - The CONNECT method converts the request connection to a transparent

TCP/IP Tunnel. To facilitate HTTPS through an HTTP proxy. This is used when a client

wants to establish a secure connection to a server through a proxy server.

1. 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?

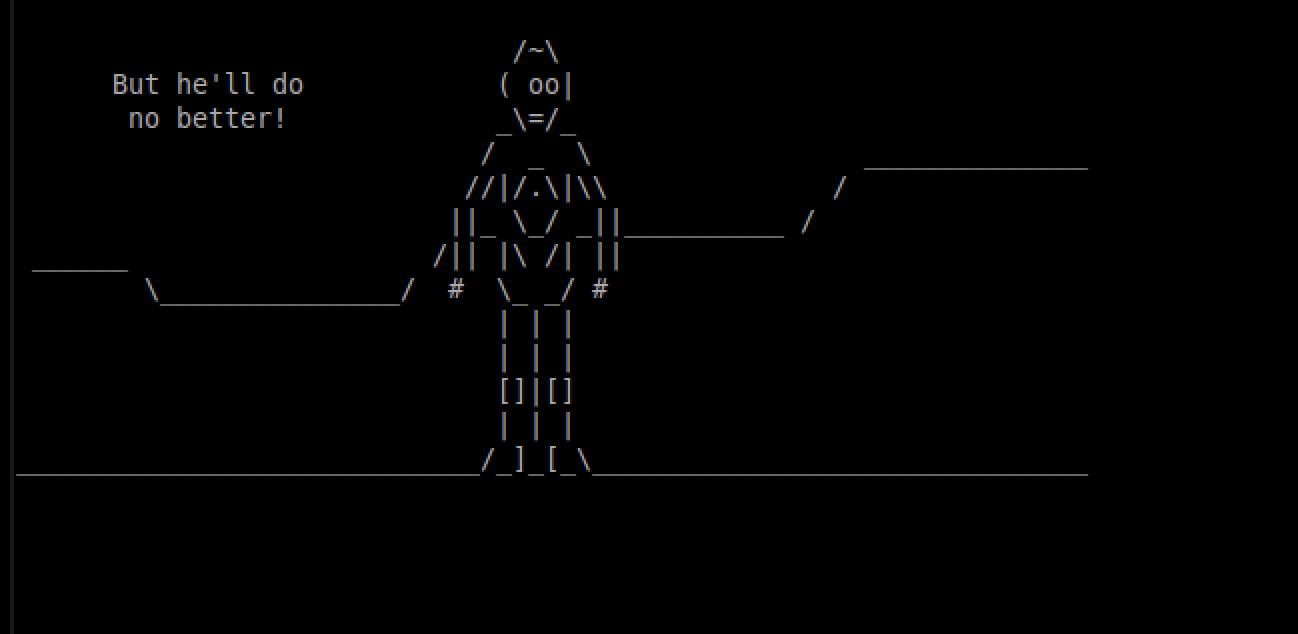
* 1. Using wget on example.com to view the last modified data of the webpage, the last HTTP  
      the return Status given was ‘200 OK’. I used the ‘wget -S example.com’ to see the return

status given.

1. Look up the telnet command. Use telnet to connect to [www.telehack.com](http://www.telehack.com), Then type

Starwars, What does this telnet server do?

1. Using “telnet telehack.com” allows me to connect to telehack.com. After typing Starwars, it

starts playing Star wars in the common prompt in a ASCII version.

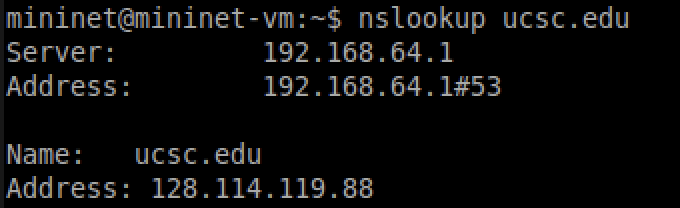
1. In your own words describe what a DNS resource record (RR) is. Now using the command line

tool nslookup find the MX resource record of ucsc.edu. What does this resource record mean?

1. A DNS resource record (RR) has information about about a resources in a zone like hosts,

that the zone contains. Using the command “nslookup ucsc.edu” shows the Server IP,

Address IP and port number along with the name and the address. This resource record stores

the data about the domain names and IP addresses. 

1. What does the command nslookup -type=ns . do ? Explain its output. (Note: the . is part of the

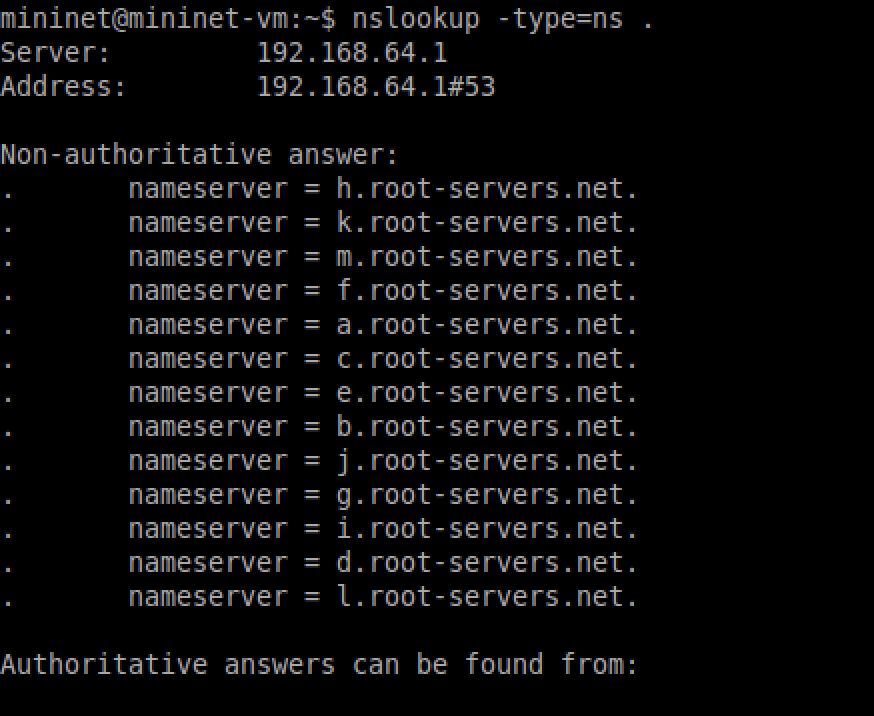
command!)

1. The command nslookup -type=ns . is used to search DNS servers for the authoritative

name servers of the root domain given by “.” . In the output it first gave the Server IP

(192.168.64.1) and the Address and the port number (192.168.64.1 #53). Then it gives the

name servers for the root domain (.). Each letter from a to m corresponds with a different

root server. 

1. How can multiple application services running on a single machine with a single

IP address be uniquely identified?

1. A single machine with a single IP address can running multiple application services are

uniquely identified by each application using a different port number. For example,

Spotify could have a port number of 8090 while Apple music has a port number of 9090.

1. What is the purpose of the window mechanism in TCP?
   1. The purpose of window mechanism in TCP is to ensure how many packets are sent at

at a time that depends on the size of the window and each packet. It allows to control the flow

Of packets between two networks. It also allows us to know that each packet is received at

the other side.

1. What is an MTU? What happens when a packet is larger than the MTU?
   1. An MTU is the Maximum transport unit, or in other words the size of the packet. If the

packet size is larger than the MTU then the packets will be divided into smaller packets

and these packets are put back together by the receiver into the original size.

Resources:

<https://www.ietf.org/rfc/rfc2616.txt> (1,2)

​​<https://man7.org/linux/man-pages/man1/wget.1.html> (3)

<https://phoenixnap.com/kb/nslookup-command> (5)

<https://linux.die.net/man/1/nslookup> (6)

<http://www.tcpipguide.com/free/> (7,8)

<https://www.cloudflare.com/learning/network-layer/what-is-mtu/> (9)

### Lab Questions:

1. 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

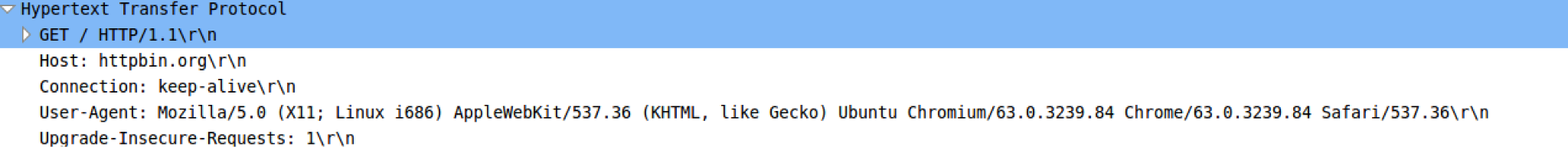
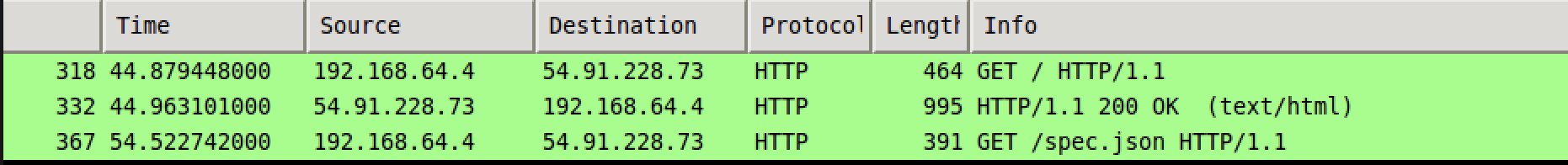
this request?

1. The initial request that the compute made was the request made in the first line of this

screenshot. The HTTP method the computer used to make the request was the GET method,

the source was 192.168.64.4 which is my computer and the destination is 54.91.228.73

which is the webpage.



1. Find the HTTP packet that corresponds to the initial response the server made to your

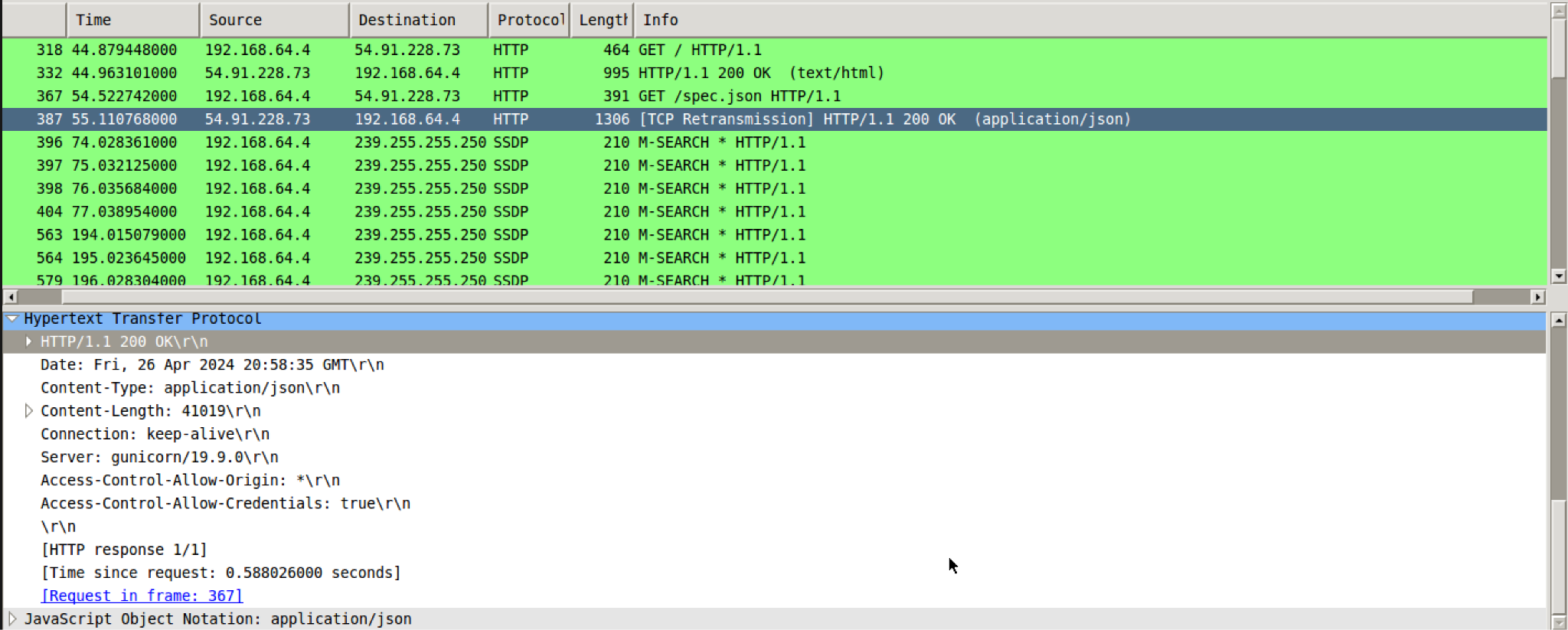
request. Take a screenshot of this packet. What HTTP status code did the server return?

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

1. The server returned a “200 OK” status code, in this code the source was “54.91.228.73”,

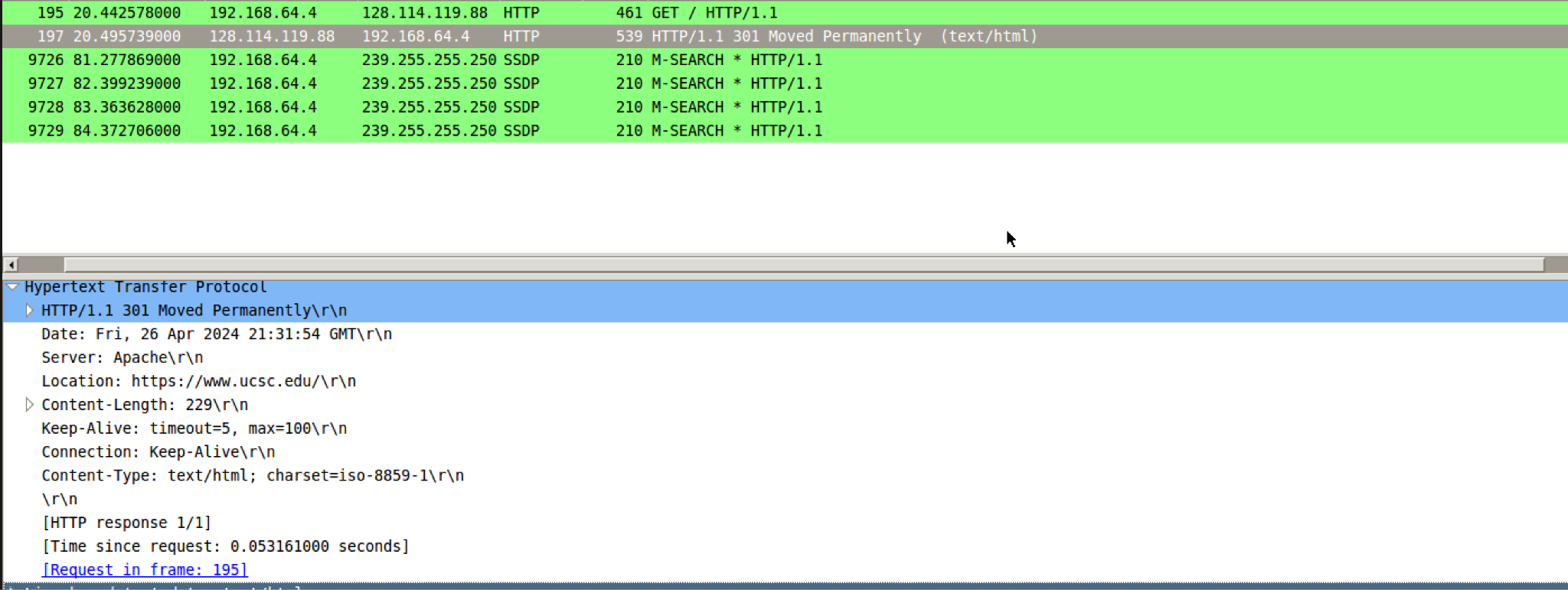
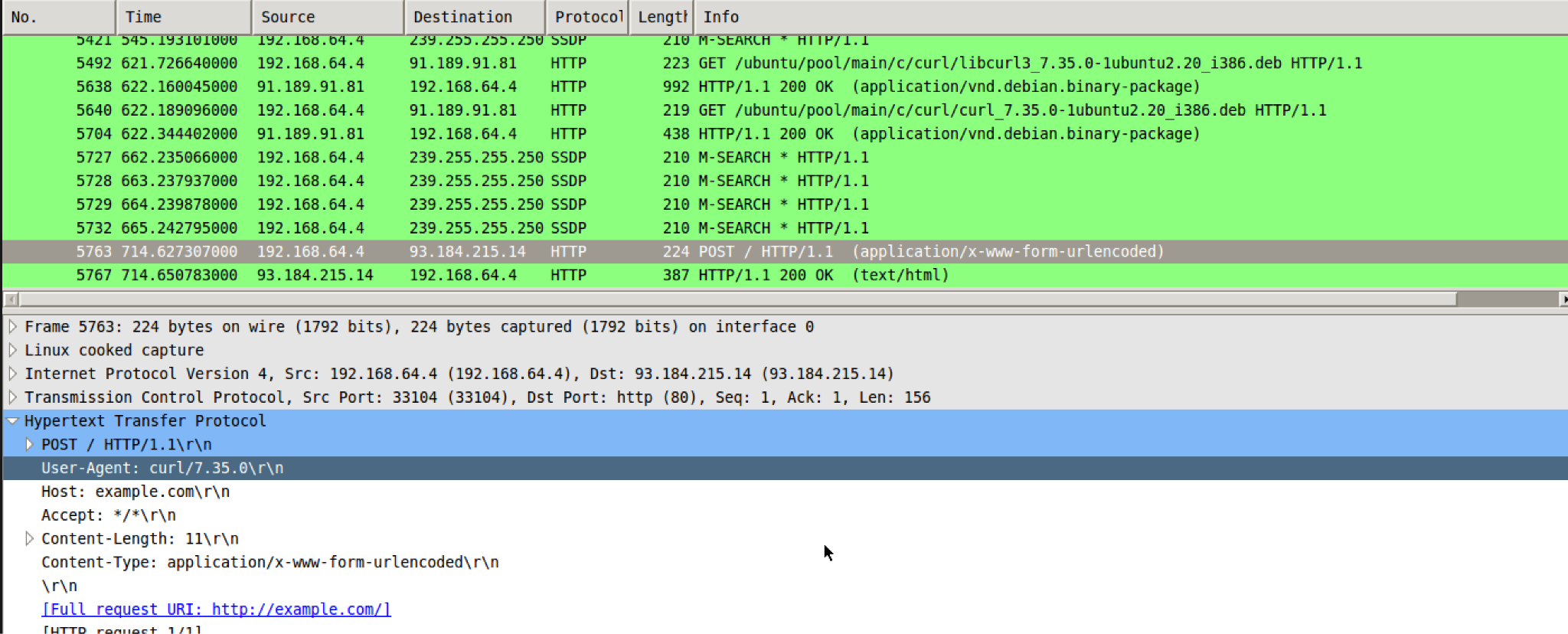
which is the webpage and the destination is 192.168.64.4, which is my computer. This means

that this packet was coming from the web page and going to my computer. The content

the type of the response the server is sending back is “application/json\r\n”. 

1. 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.

1. The computer requested to open “<http://ucsc.edu>” however the server of that URL was moved to a different server so it responded with a “301 Moved Permanently” response status code. Going more into depth, looking at the Hypertext Transfer Protocol, it shows that the server was moved to “<https://ucsc.edu>” and not “<http://ucsc.edu>” like we requested, which is why we got the response code of “301 Moved Permanently”
2. Take a screenshot of your packet, and explain what you did to create it.
   1. In order to create a packet other than GET, I used the “curl - - data ‘hello’ http://example.com” terminal. This command creates a POST HTTP command because of the “--data” option. The “--data” option is used to send data in a POST request. When using “--data” it automatically sets the HTTP method to POST.
3. 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](http://www.example.com)?

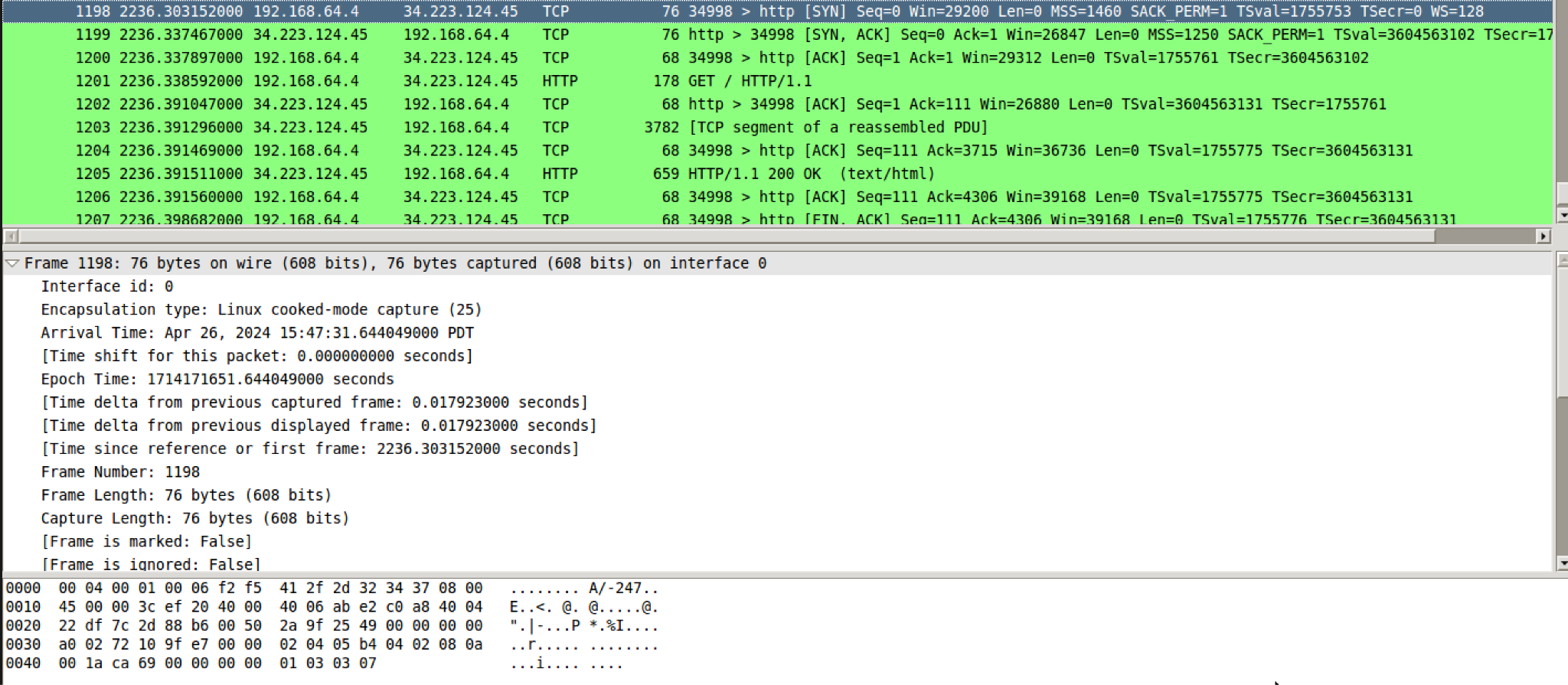
1. I believe that all the packets from No. 1198 to No. 1201 were the steps that the computer

took before the webpage was loaded. I think that these are the correct packets because

the destination of the those packets is the same destination (34.223.124.45) that the

GET HTTP command used in order to get to the “http://neverssl.com” webpage. Making

the IP address of http://neverssl.com, 34.223.124.45.



1. 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.

1. Using the command “wget --no-dns-cache <http://34.223.124.45/>” downloads the same

content, without sending DNS requests. Using “wget --no-dns-cache <http://34.223.124.45/>”,

does not initiate any DNS request because its directly using the IP address provided

instead of a domain name. Also the “--no-dns-cache” flag prevents wget from caching any

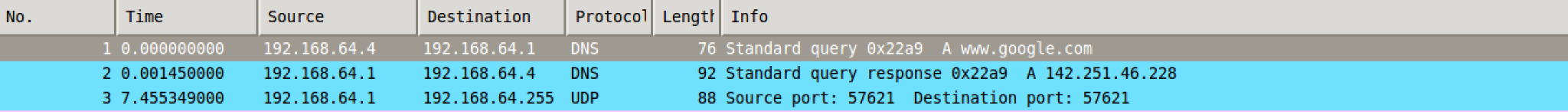
DNS, since our DNS was not able to flush.

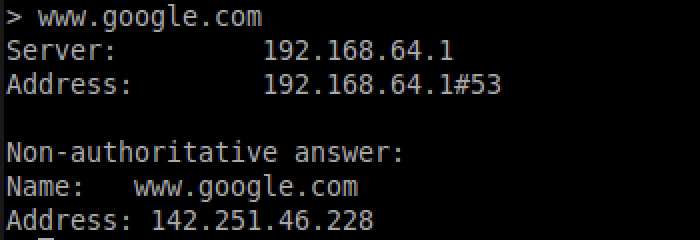
1. Take a screenshot of the packets corresponding to your request, and the response

from the server. If the request was resolved, what is the IP address you were given

for [www.google.com](http://www.google.com)?

1. The IP address I was given for “[www.google.com](http://www.google.com)” was 142.251.46.228.





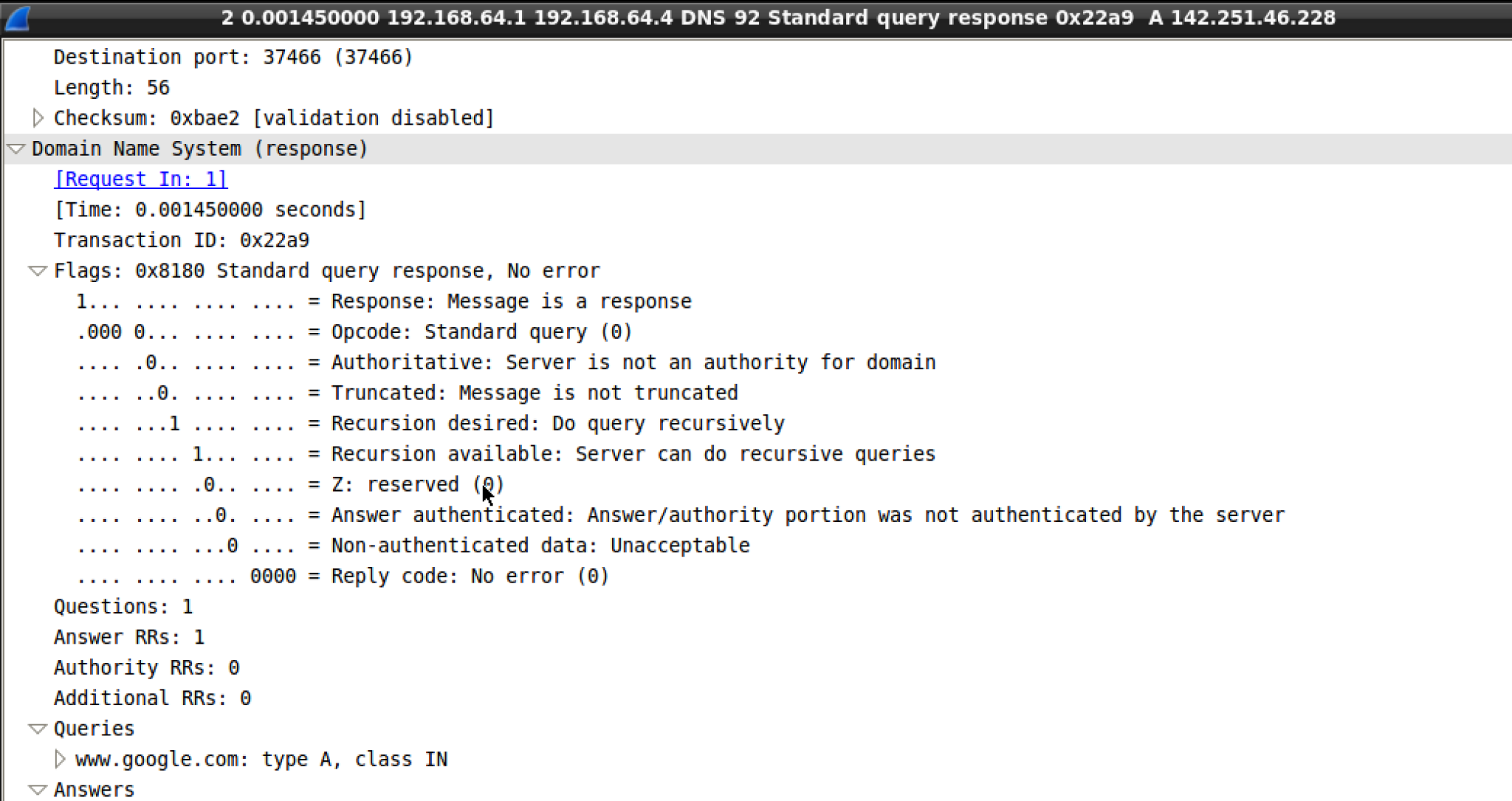
1. Did your computer want to complete the request recursively? How do you know?

Take a screenshot proving your answer.

1. Yes, my computer wanted to complete the request recursively. Opening the packet

you can find that under Domain Name system and Flags, that Recursion is desired,

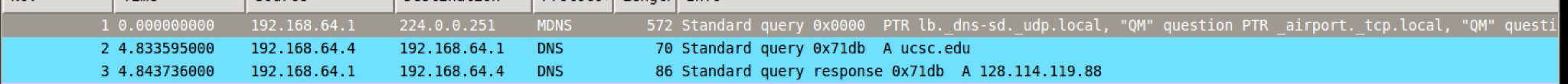
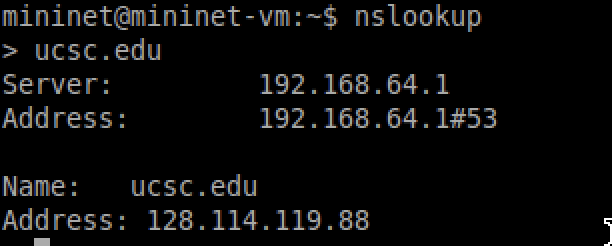
and it's going to query recursively.



1. 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?

1. The IP address I was given for UCSC.edu was 128.114.119.88. 
2. What is the authoritative name server for the ucsc.edu domain? How do you know?

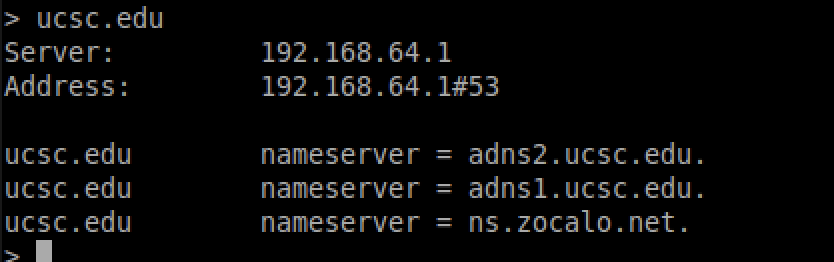
Take a screenshot proving your answer.

1. The authoritative name servers for the ucsc.edu domain are “adns2.ucsc.edu.”,

“adns1.ucsc.edu.”, and “ns.zocalo.net.”. By typing “set type=ns”, it sets the query

type to only Name servers. So when typing in a domain name, it will only return the

authoritative name servers of the domain.



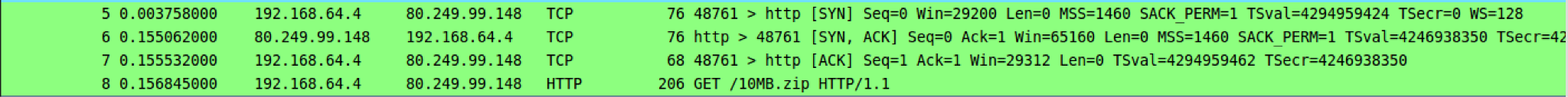
1. Find the packets corresponding with the SYN, SYN-ACK, and ACK that initiated the

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?

1. The initial window size that the computer advertised to the server was 29200, and the initial

Window size that the server advertised was 65160. 

1. Find a packet from the download with a source of the server and a destination of your

computer. Create a tcptrace graph with this packet selecrated. Take a screenshot of the

graph and explain what it is showing. Look into the Wireshark documentation if you

I need assistance making this graph.

1. This is tcptrace graph that goes from the server to my computer, which shows the sequence

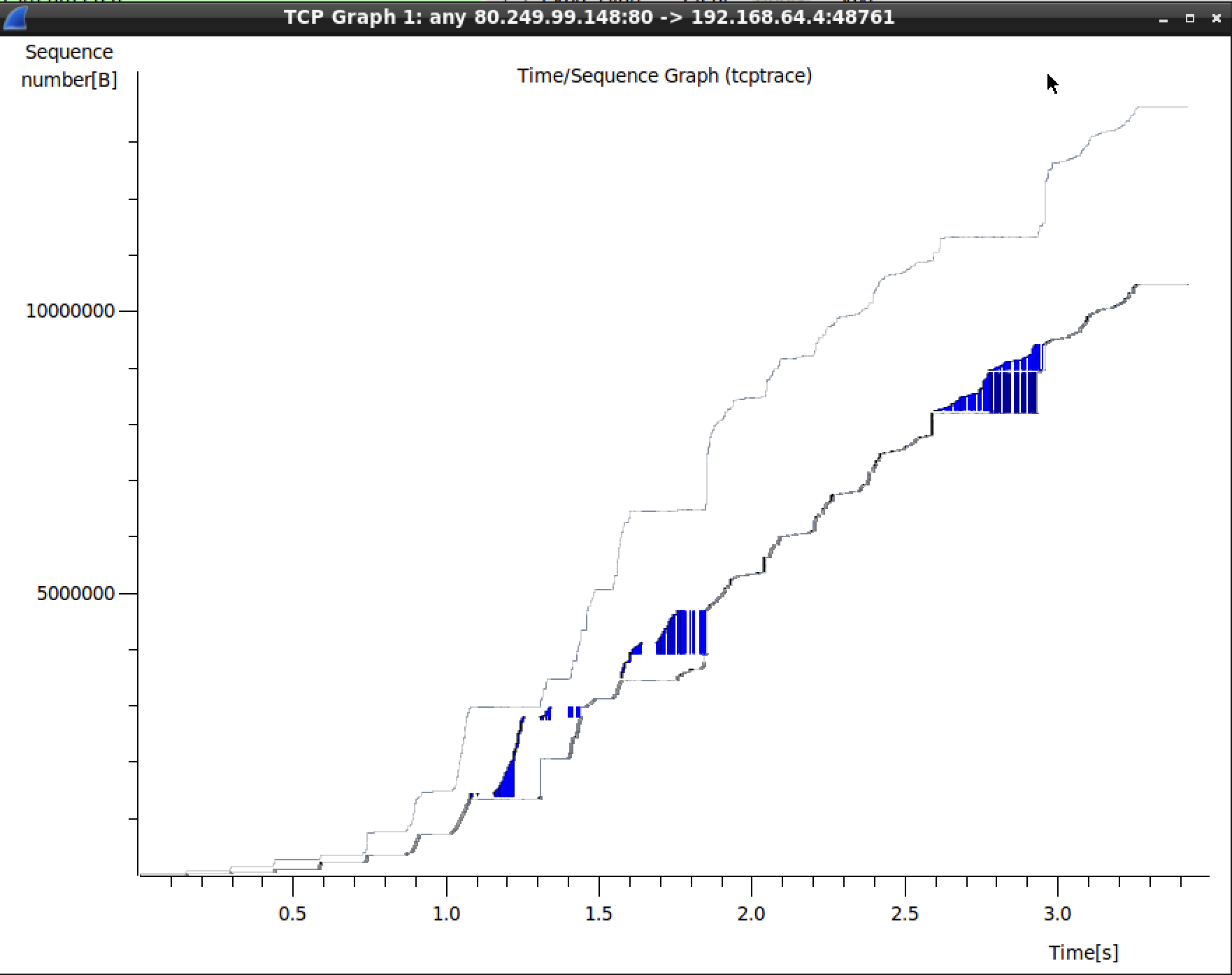
of numbers of TCP over time. This graph shows the blue lines which are the sequence

and the vertical lines are the TCP segments. Like looking at time 2.5s to 3.0 s you can see

the sequence numbers which is the counter of the bytes transmitted. So with each byte sent,

the sequence number is incremented. So by the end of 3 seconds, there around 10000000 bytes

sent.



1. 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.

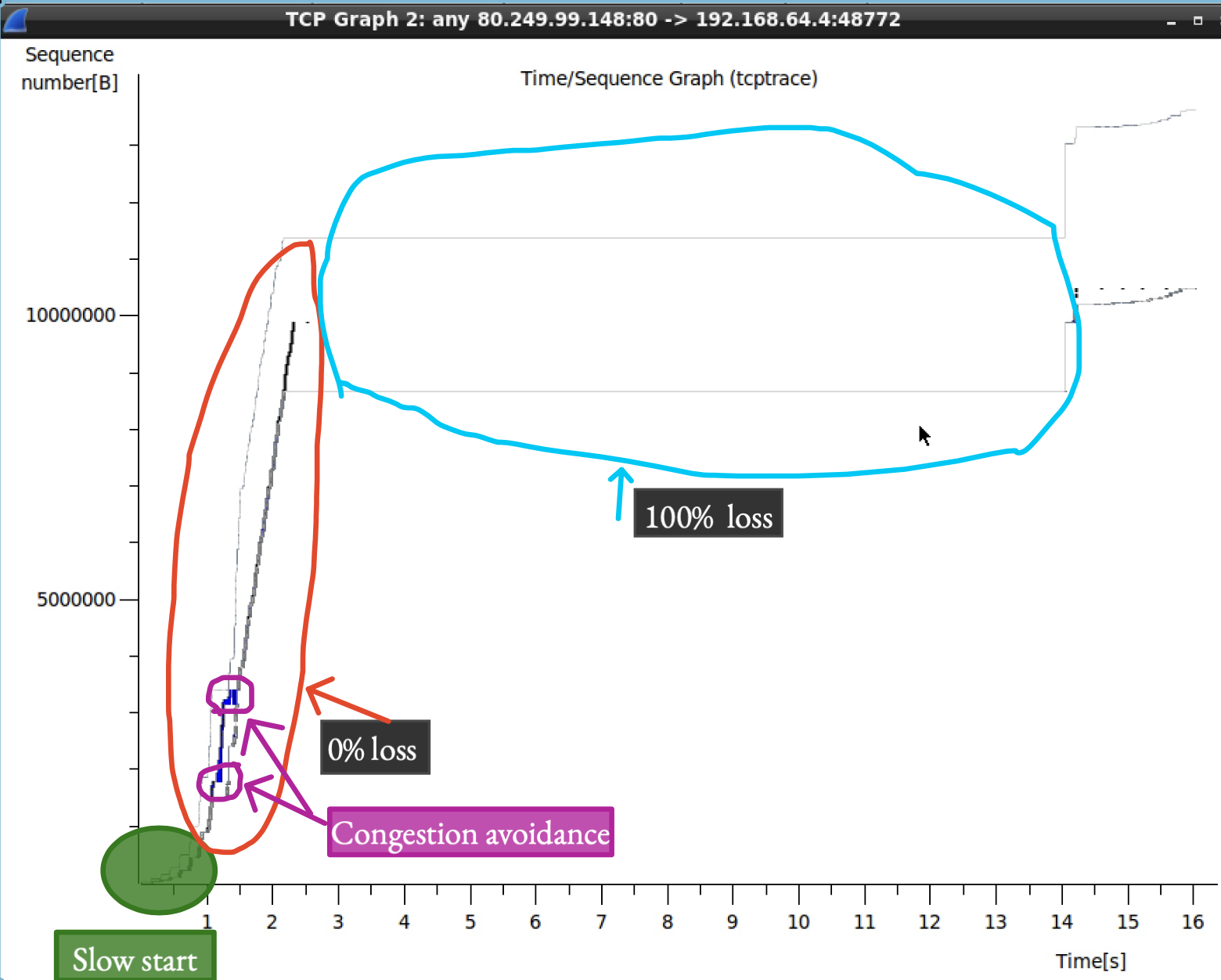
1. This is a TCPtrace graph where a packet is sent from the server to my computer with

where there is TCP loss and not in a loss. In the first couple seconds, where there is

0% loss, so it is able to download without any loss, but after a while I put in the command

To be 100% loss. In the 100% loss section, there is no downloading progress which is why

it is just flat, since there is no downloading going on.



Resource:

<https://www.linode.com/docs/guides/how-to-use-nslookup-command/> (10)

<https://www.packetsafari.com/blog/2021/10/31/wireshark-tcp-graphs/> (12,13)