Twisha Sharma
CSE 150

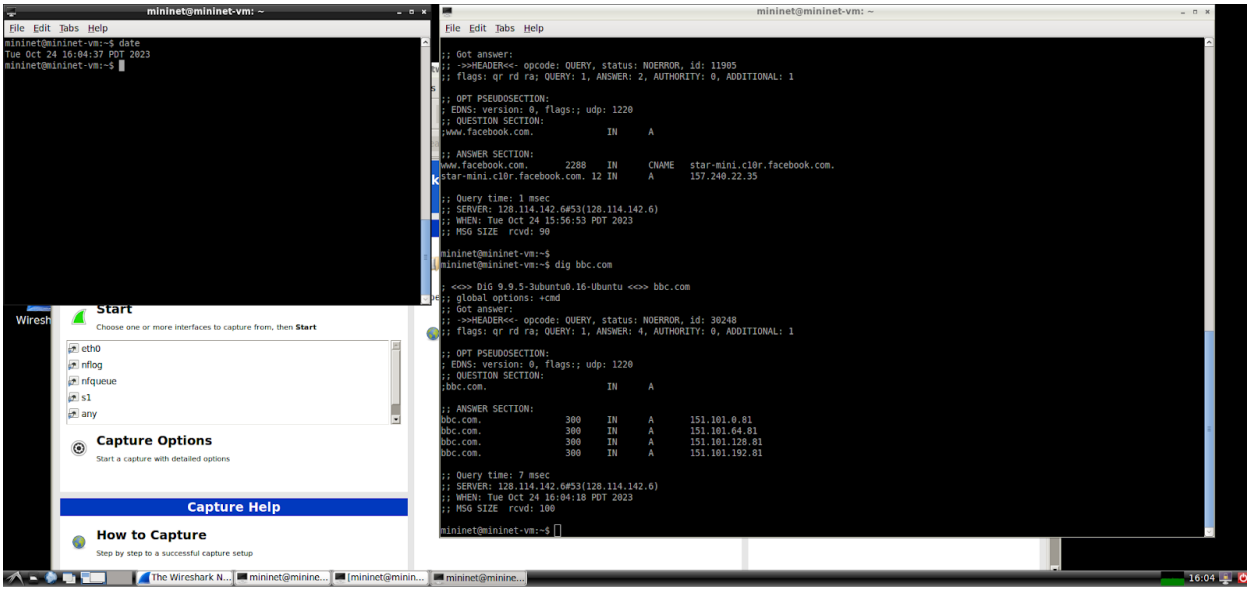
Lab 2

Notes:

- Average throughput = file size/ time taken to download file
- Look at the answer section when you run dig : that is all that DNS is; the later domain in the response answer section, references that a query was made for the domain run with the dig command and DNS responded with all the records it had for that website

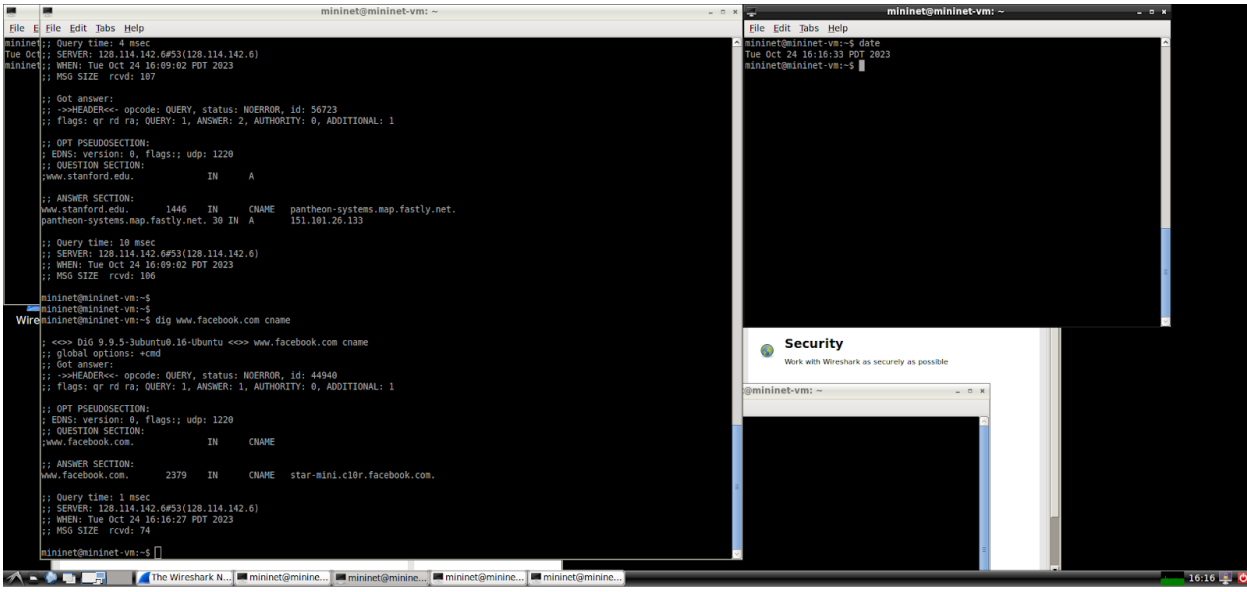
1. Resource records are the data or files that DNS as a database stores, this holds the information of the IP address of the host name that was given by the user. If the user were to give a host name, DNS would query its servers, either using caching, iteration or recursion, to get the IP address needed. When the user needs to ultimately load a webpage, the first step is to resolve the DNS and get the server's IP address. Then the TCP connection can be opened to send HTTP messages.

2.
    a. The resource records returned are all of type A, which means they hold the mapping for the path through the DNS server hierarchy, to the IP address information from the hostname.
    b. The TTL field is for caching purposes. DNS used caching to resolve this query and because of that, it needed to have TTL on its entries, because if a named host changed its IP address, it may not be known across the network, until the TTL has expired. This means that some cached entries may be out of date. TTL is necessary for caching, as the value indicates how long a DNS information should be cached by the resolving server The TTL value was 300.



3.
    a. Domain aliases are used on the internet to allow users to host a website on one domain but then have additional domain names, pointing to the same website.

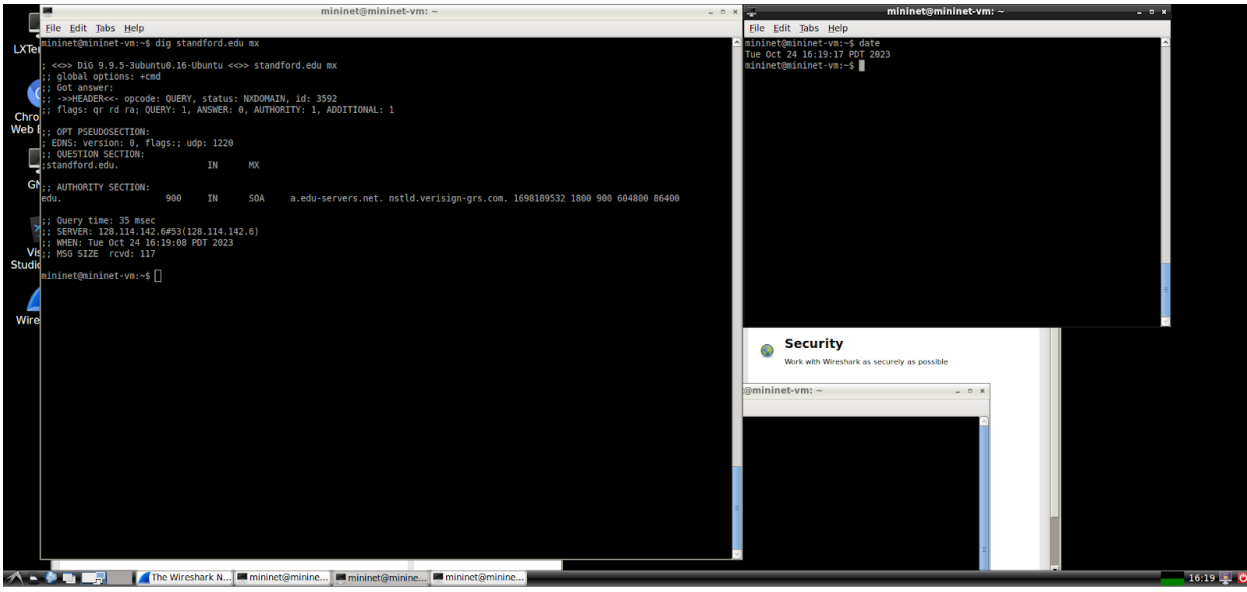
b. The command used to find the CNAME record for www.facebook.coom is dig www.facebook.com cname. The alias name is www.facebook.com and the canonical name star-mini.cl0r.facebook.com.



4.

a. The output shows information from the MX record that provides the value of the mail server associated with stanford.edu

b. The command is dig www.stanford.edu mx



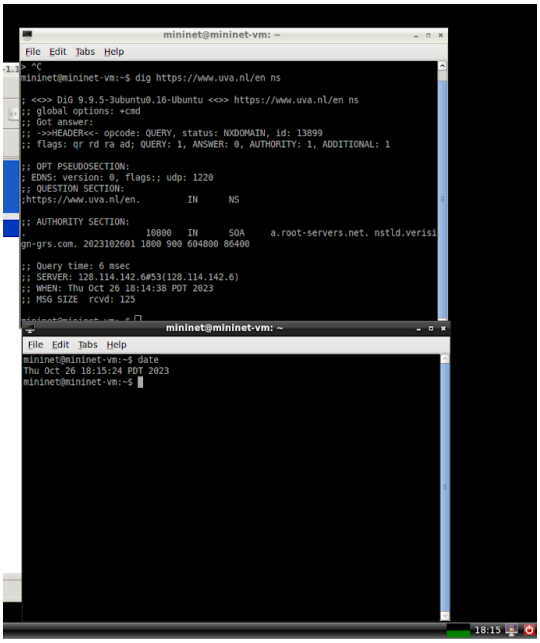
c. What was displayed was the name of the mail server associated with stanford.edu

d. What corresponding DNS queries are made for accessing the site www.stanford.edu compared to sending an email to joestudent@stanford.edu the record being accessed by DNS will be different. DNS uses type MXfor mail, and type A the webpage. The DNS

mechanism used in this example is MX record lookup - used to determine the mail servers that are responsible for getting the email from the domain. Additionally, the authoritative dns servers are used to provide the MX records which specific the mail seers to which email should be delivered.

5.

a. Authoritative servers are a trusted place in the DNS database, for providing records and responses to the DNS queries. Authoritative servers are very important for mapping the domain name to the IP address which means that the user can be connected to the web server. These servers also provide TTL which is necessary for caching as the value indicates how long a DNS information should be cached by the resolving server.

b. 

The name of the university I chose was the University of Amsterdam

c. Once the subdomains have been made, and authoritative servers have been set up specifically for those subdomains, the server will be responsible for handling the DNS queries related to the subdomain. After that it's also important to update the DNS records for the top domain server/parent server. This means specifying the IP addresses/hostnames of the authoritative name servers for each subdomain in the DNS records. Because of the individualized subdomains, departments will have control to make DNS changes/updates without affecting the whole university DNS infrastructure.

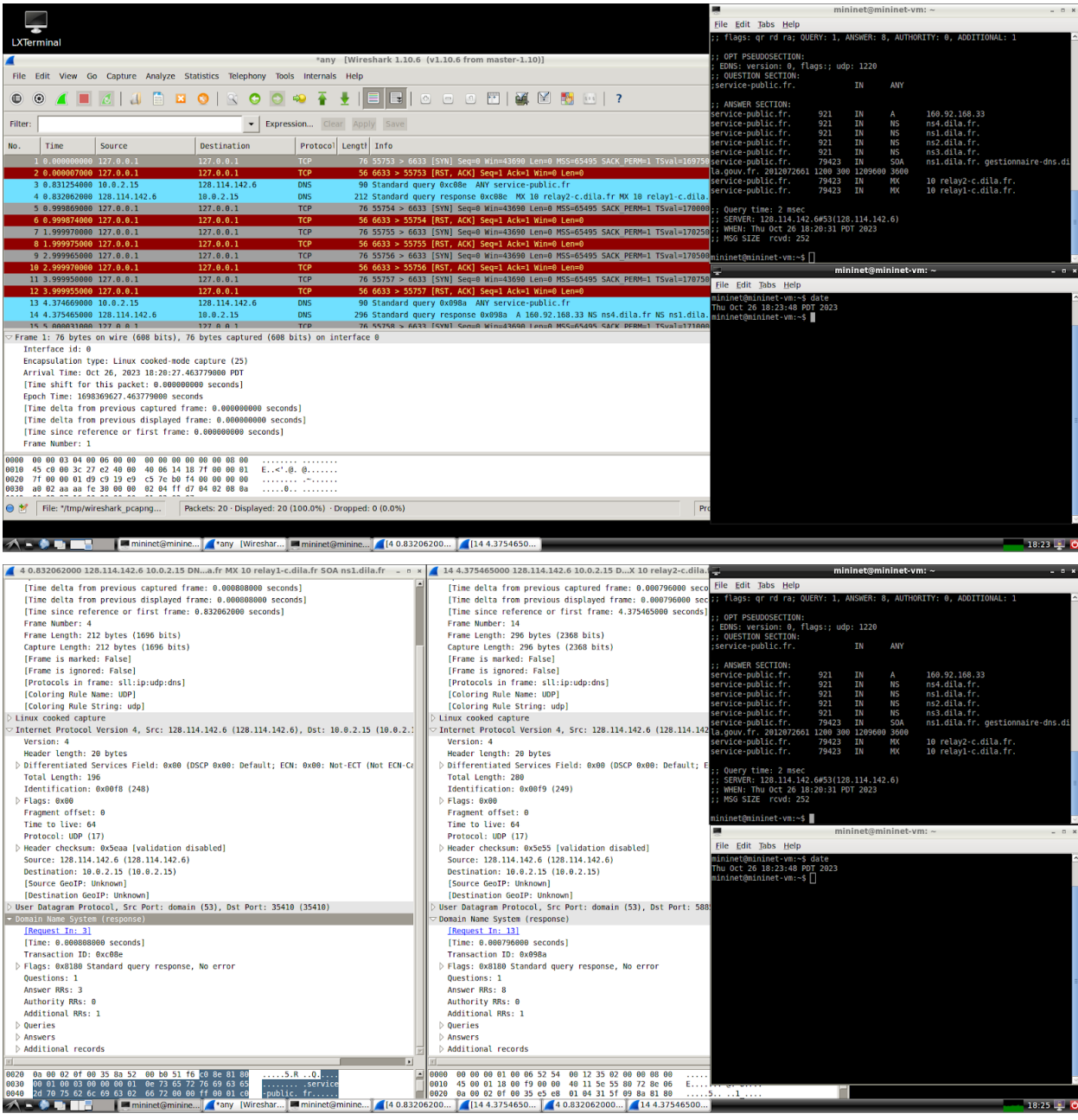
6.

a. “ANY” is used to get all the available DNS records for a given domain. You can gather information about a domain meaning all types of records, (A, AAAA, MX, CNAME). The ANY command allows a comprehensive view of all the DNS information associated with the domain.

b. Multiple A records are returned to enable load balancing and fault tolerance, both of which are essential for maintaining a high performance and availability for the site - something that is necessary with large websites like Amazon.

7.

a. No, the queries take different amounts of time to run. There is a slight difference in the timing with the first one taking slightly longer than the second.



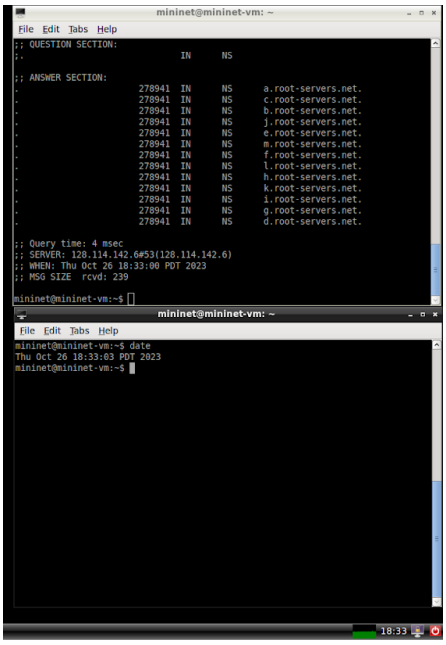
b. They can take different times to run due to factors such as network conditions, DNS caching, and the amount of load on the DNS servers. There can be congestion on the network, routing changes or network latency which can affect the network connections. With caching, the look time can be affected when the result of the query has been recently

cached; it could be done faster the subsequent time it is queried. The authoritative server load can affect the lookup time as well, because the servers can have heavy traffic or perhaps may be temporarily unavailable.

8.

a. Root name servers are the starting point of DNS queries, and they are responsible for the Top Level Domains. There are 13 of them around the world. When the DNS resolver receives a query for a domain name the first thing it will do is contact a root name server to determine the authoritative name server that is responsible for the top level domain server of the domain the query was made with. It's important to note that root name servers do not store information about the specific domain names, this is why the path must follow to the authoritative servers. The root server only refers to the authoritative server's information telling the resolver where to go next.

b. The command to find the root name servers is dig NS

c. 

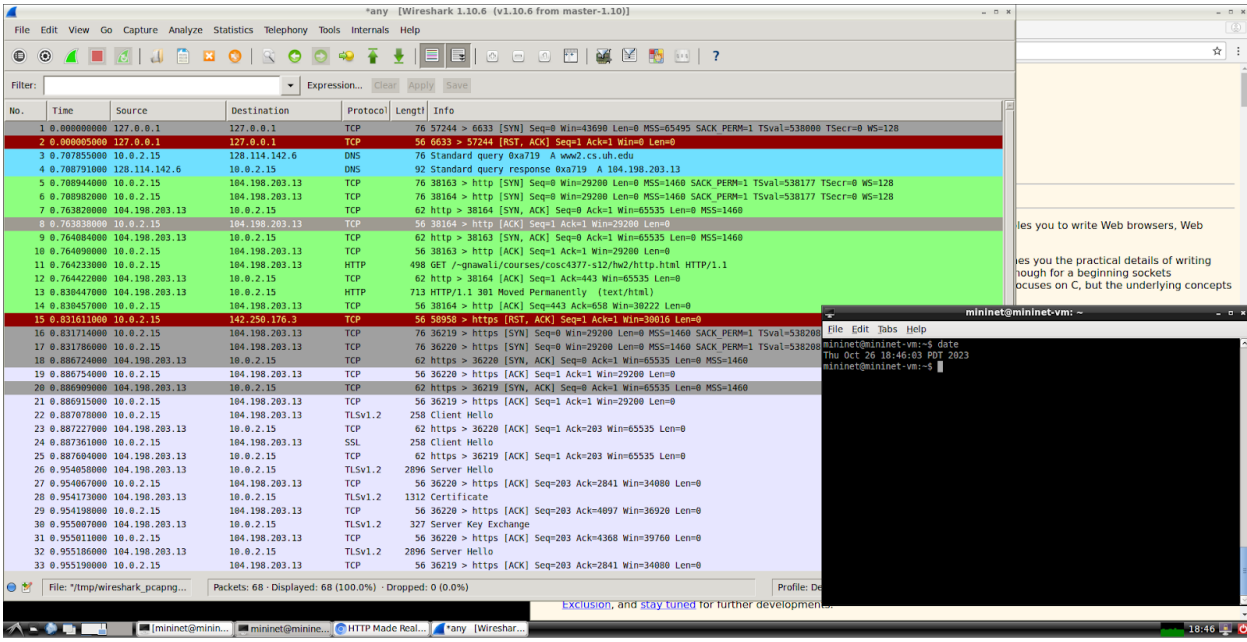
The results in the screenshot show the names of the root name servers (labeled a.root-server.net, etc, 13 in total). These root name servers are distributed worldwide, and they play a critical role in the DNS infrastructure.

9.

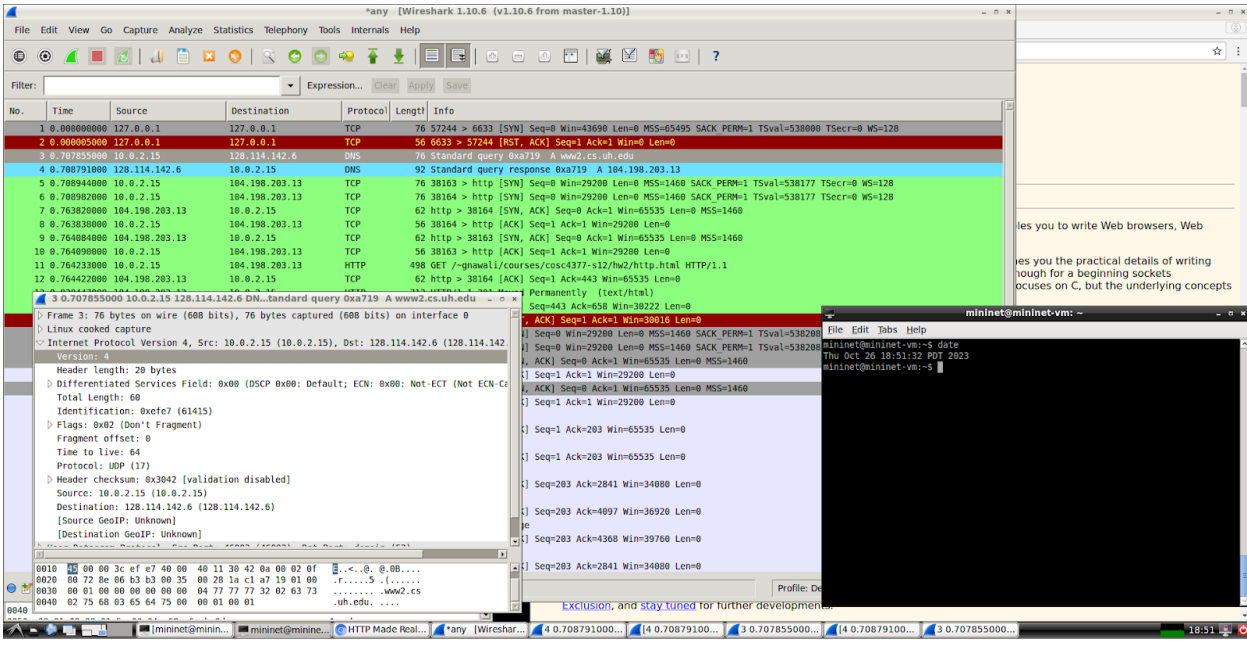
a. The command you use to run reverse DNS lookup on ip address 144.112.62.105 is: dig -x 144.112.62.105

b. The domain associated with the IP address 144.112.62.105 is: z.arin.net. dns-ops.arin.net. 20

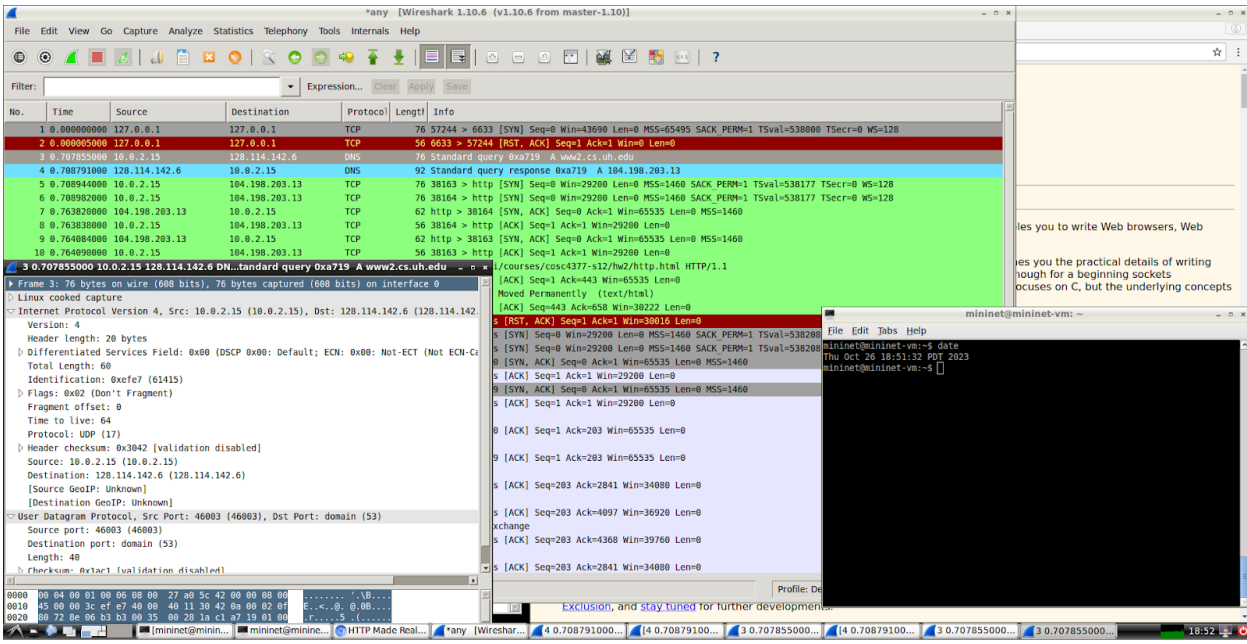
10.

a. 

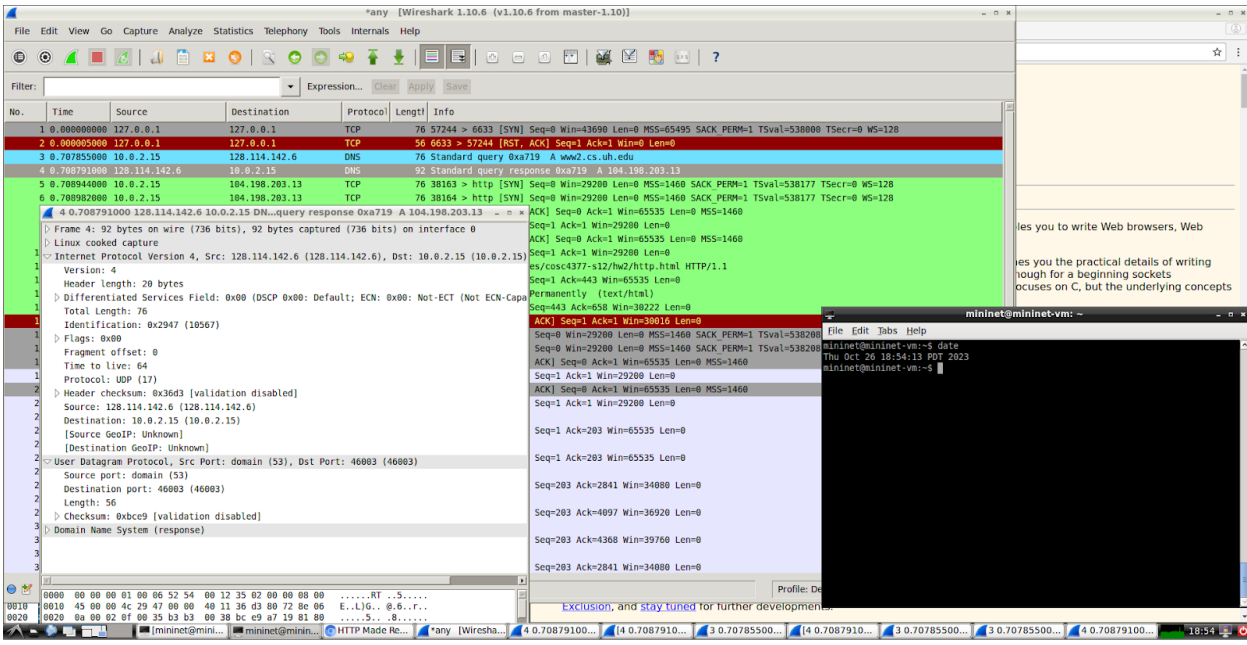
b. Typically, DNS queries and response messages use UDP because it is faster, and more suited for simple request, single response communication. However, DNS can use TCP in some cases when the response size exceeds 512 bytes.

c. 

d. The destination port for the DNS query is 53. This was expected as this is a very well known port for DNS.



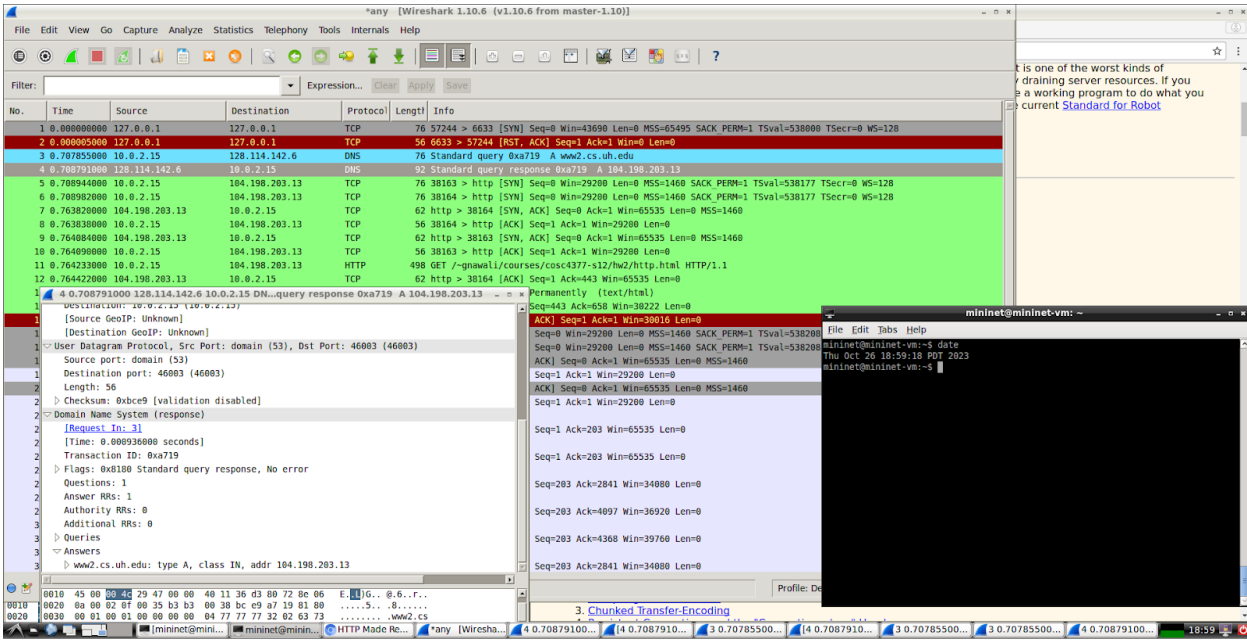
e. My host received the response on port 46003. This isn't a well known port in the standard port numbers, however; this is a port that was assigned or used for communication between the client (me) and the DNS server, during the DNS query and response process.



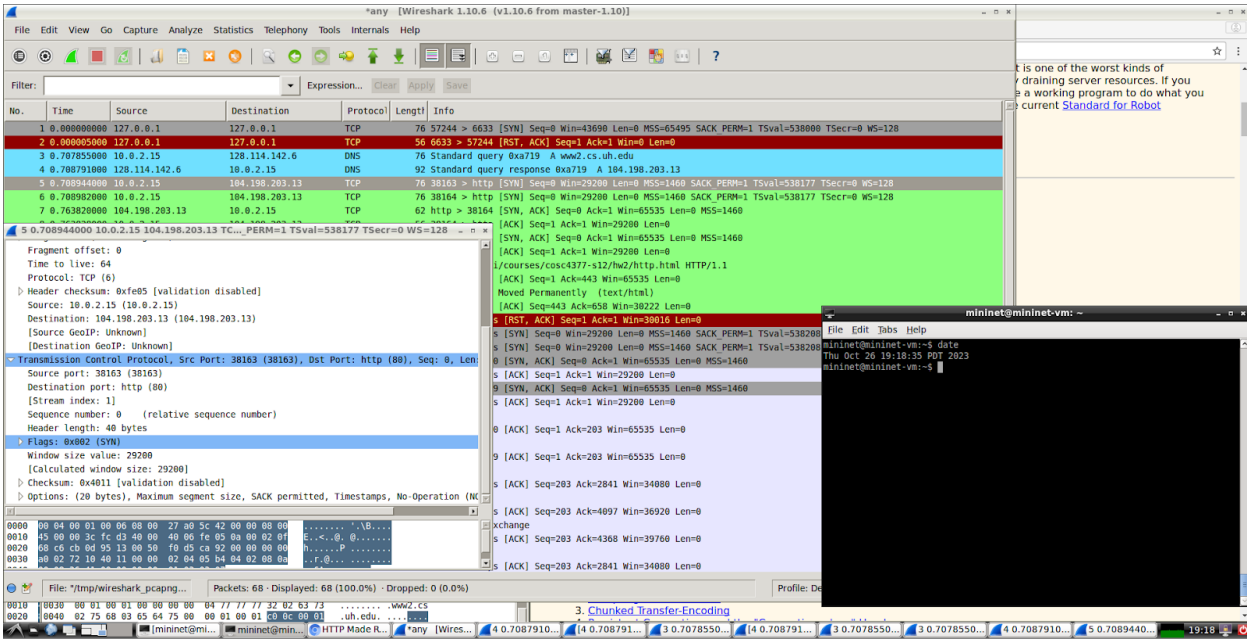
11.

a. There is only one answer that is provided and with it is the information or rather the RR record for the web page wireshark was run with. Answer: www2.cs.uh.edu: type A, class IN, addr 104.198.203.13. The Type A stands for the most common type of DNS records, Address, and is used to map the domain name to an IP address. Class IN stands for

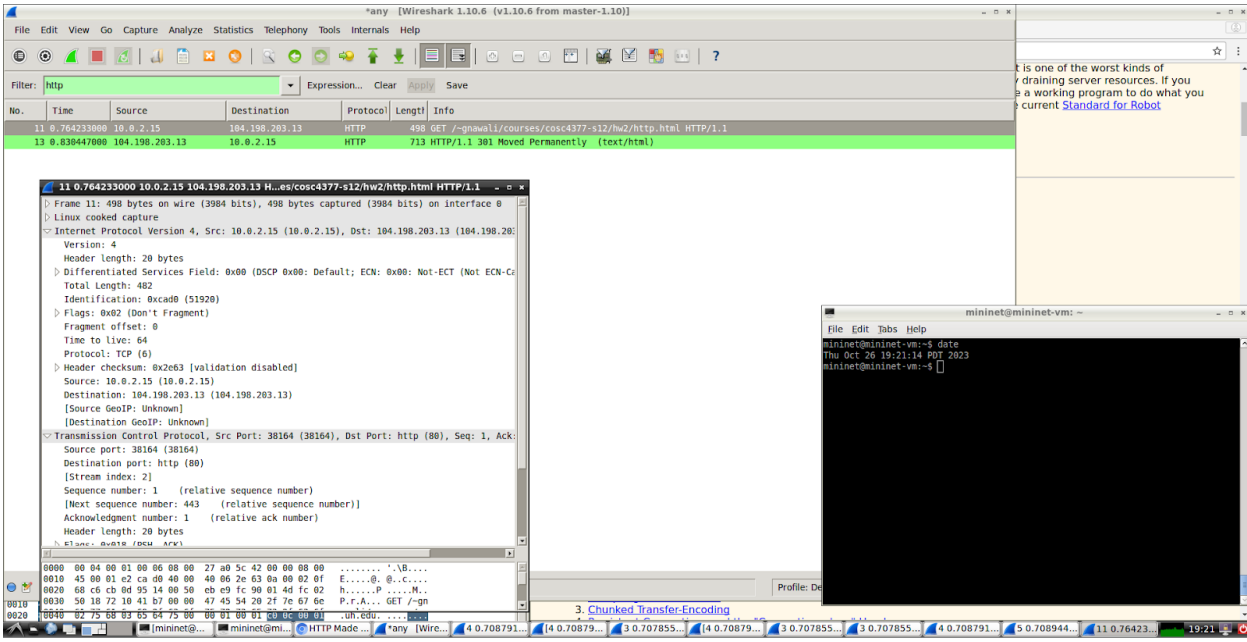
Internet and is the class of the DNS record. The Ip address is also included in the response message and this can be used by the client (me) to establish a connection to the web server.

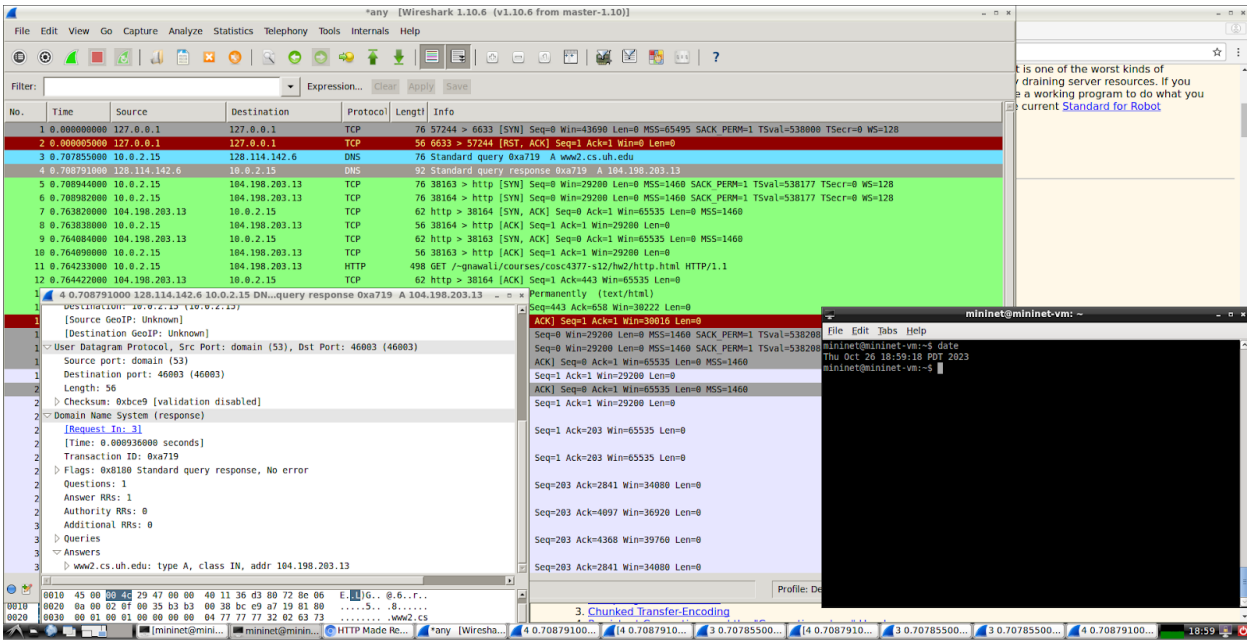


b. The very first TCP packet sent is in the frame of the screenshot below. You can see The source and destination IP addresses. Source: 10.0.2.15 (client - me); Destination : 104.198.203.13 (web server IP address). This is the packet that starts the TCP connection.



c. The destination IP address within the http GET packet is 104.198.203.13. This is the same IP address as the one returned by DNS.

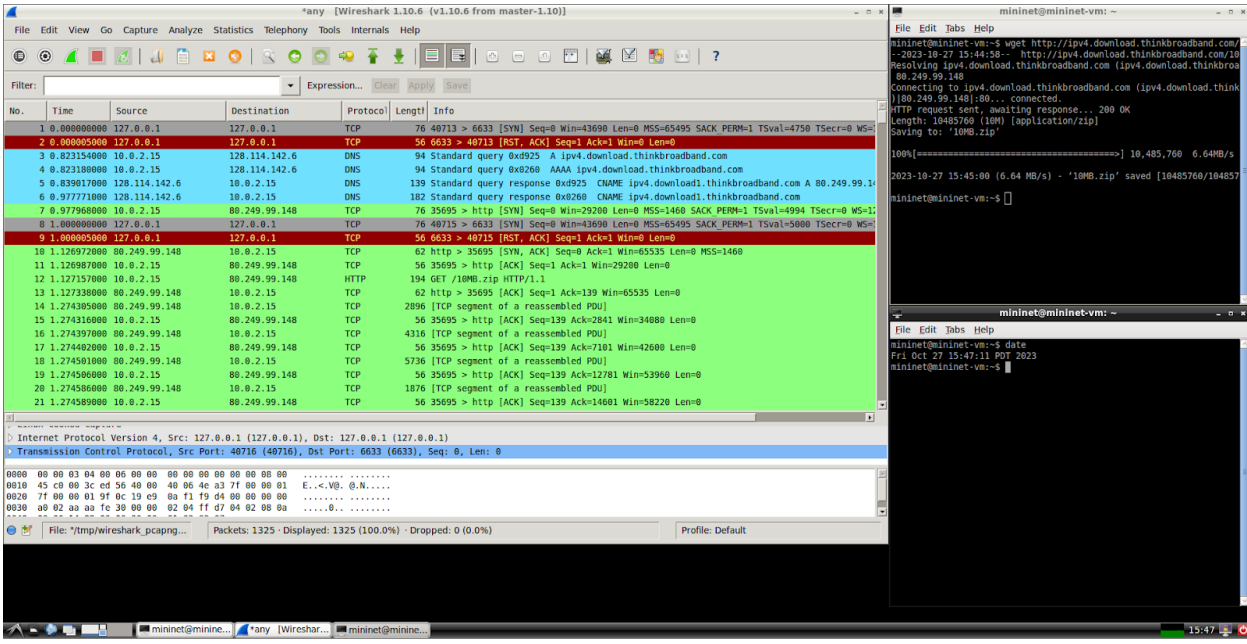




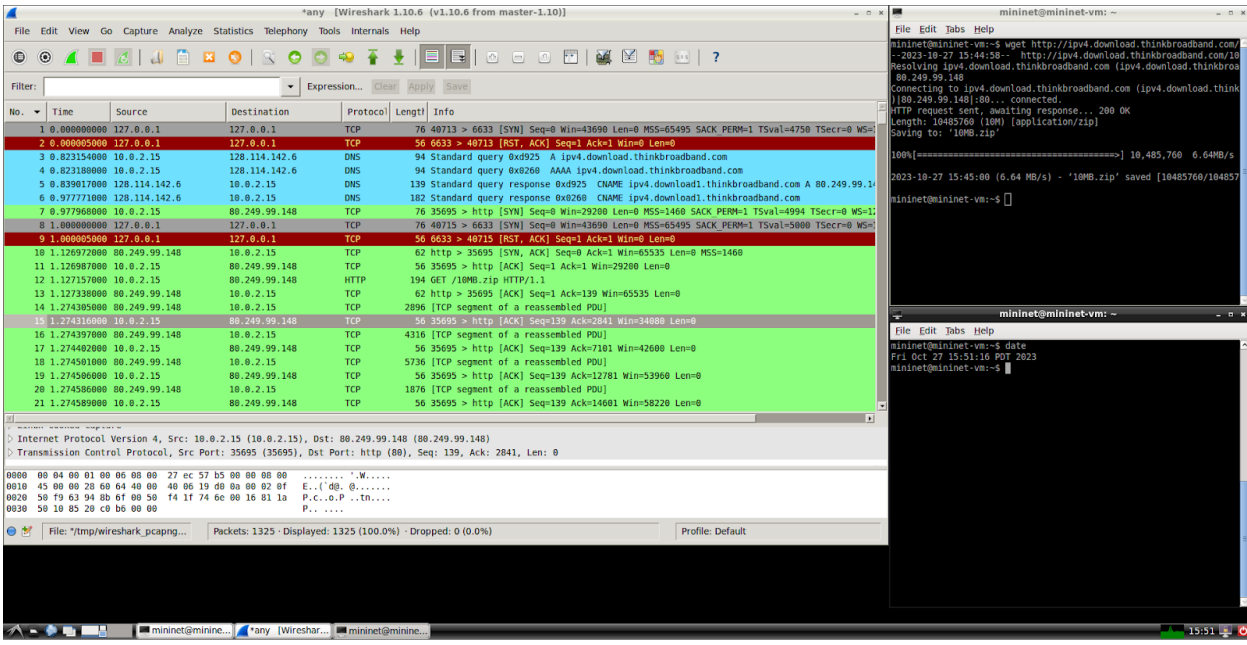
12.

a. The different protocols I see when running wget http://ipv4.download.thinkbroadband.com/10MB.zip with wireshark, are TCP, SSDP, DNS, and HTTP. TCP is the 3 way handshake for establishing a connection ; SSDP is for network traffic that is part of the network environment; DNS is for name resolution; HTTP is for the actual file transfer.

b. Frames 1, 7, 8, 10, 11, 12

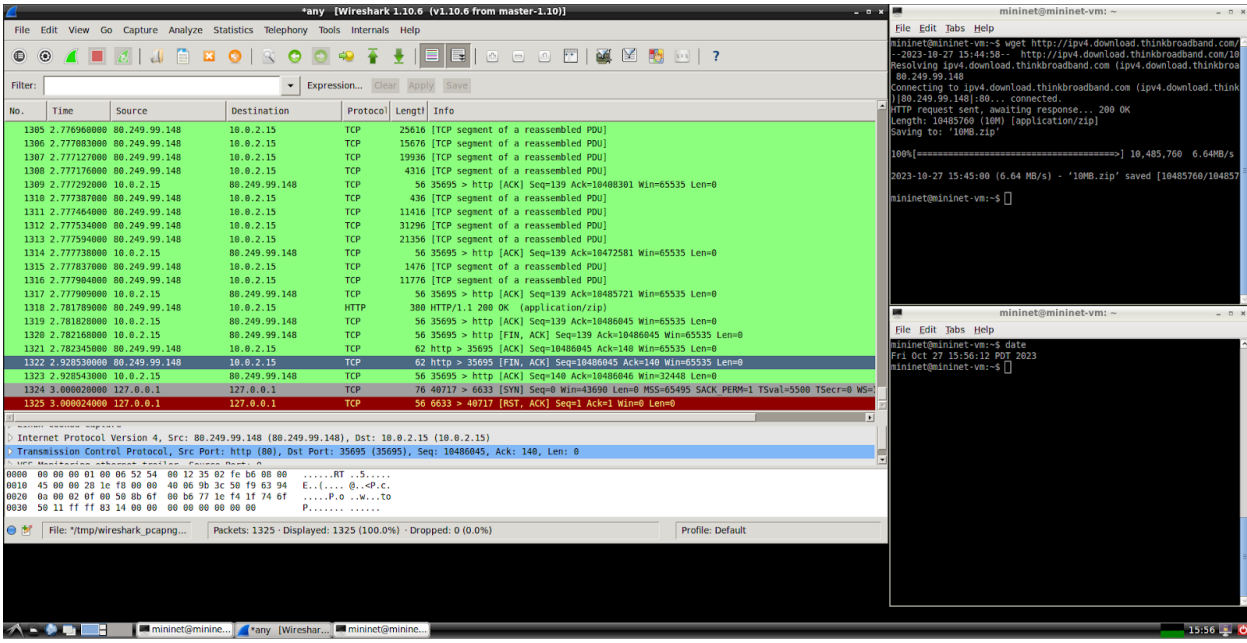


c. The first http request for the file is on Frame 12. This frame shows the http GET request which indicates the start of the file download process.



Frame 1322 is the frame that shows the last packet of the file transfer. This is seen in the

[FIN,ACK] finish acknowledged.



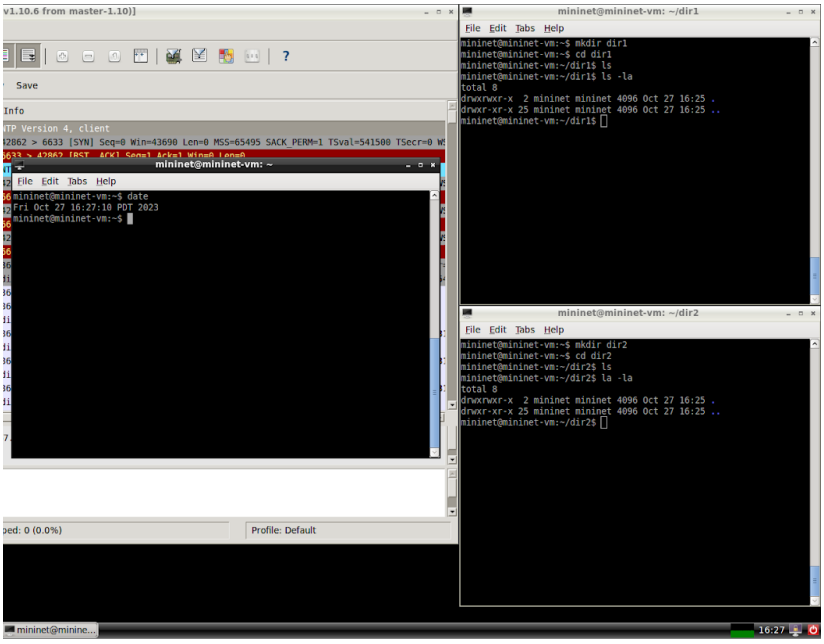
Time of first packet: 1.127157000
Time of last packet: 2.928530000
Therefore the download time is : 2.928530000 - 1.127157000 = 1.801373000

d. Average throughput = Total data downloaded / Download Time
File Size (see screenshot wget output) = 10 MB = 10*8 bits/byte = 80 Megabits
Average throughput = 80 Megabits / 1.801373000 seconds = 44.41 Mbps
Wget reports the average throughput at 6.64 MB/s. When converting the average throughput calculated by the numbers seen in wireshark (44.41 Mbps/8 = 5.55 MB/s (1MB/s = 8Mbps)) you can see that 5.55 MB/s is different from the 6.64 MB/s that wget reports. This can be due to network conditions, and the time that it takes to process and save the file locally.

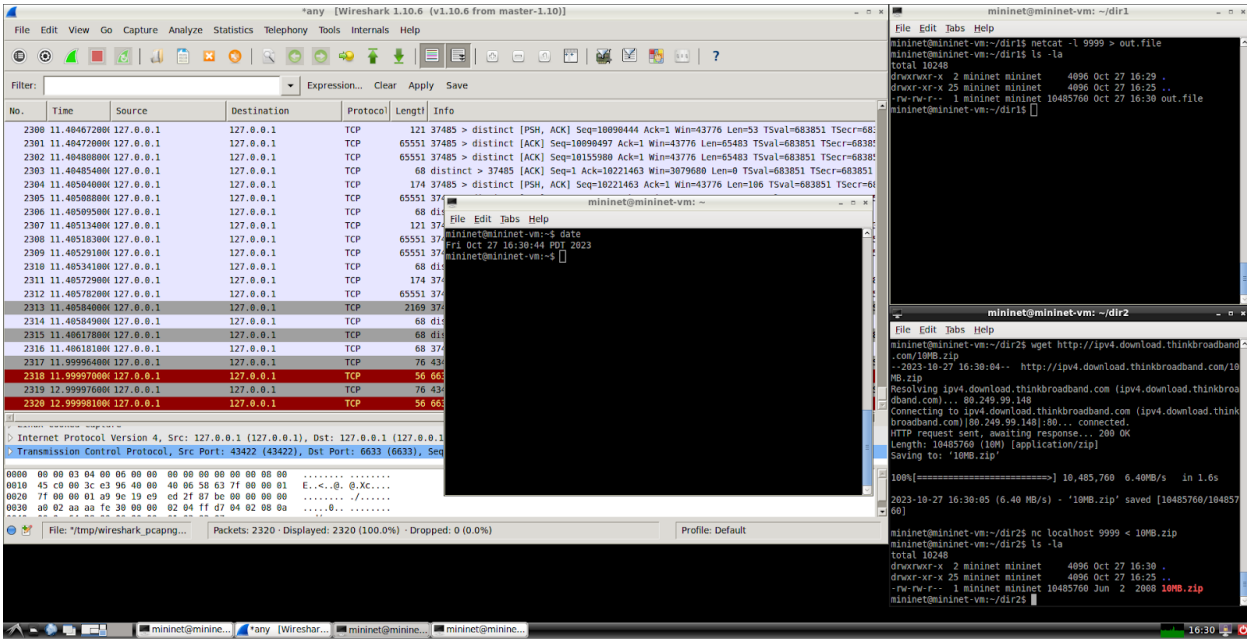
13.

a. Netcat (nc) is a tool used for transferring files, establishing network connections, or port scanning. It works on the transport and application layers, and can function as both the client and the server.

b. The screenshot below shows the terminal directories before the file was copied over



The next screenshot (below) shows the terminal directories after the file was copied over



c. Data transfer is the process of moving data from one place to another. This involves the transmission of information - texts, files, images, etc, from a source to a destination. In the transfer process there is a client (which is used to refer to the device sending the data) and the server (which is used to refer to the destination)

d. In this scenario, terminal 2 - using netcat, sends a file to another terminal 1, over the local network. The transfer is started by terminal 2, where netcat is run (“nc” command), to ultimately send the files data over to terminal 1, where it is received and written to a file. In this scenario, terminal 2 is the client, sending the data, and terminal 1 is the server (receiving the data). Note that the server (Terminal 1) is set up to listen

before the client (Terminal 2) makes an attempt to send the file. The server is waiting for the connection on port 9999 as specified on the command run in Terminal 1.

e. If the transfer was done using HTTP, the method used would be POST. In this method, the dada is sent from the client to the server for processing.

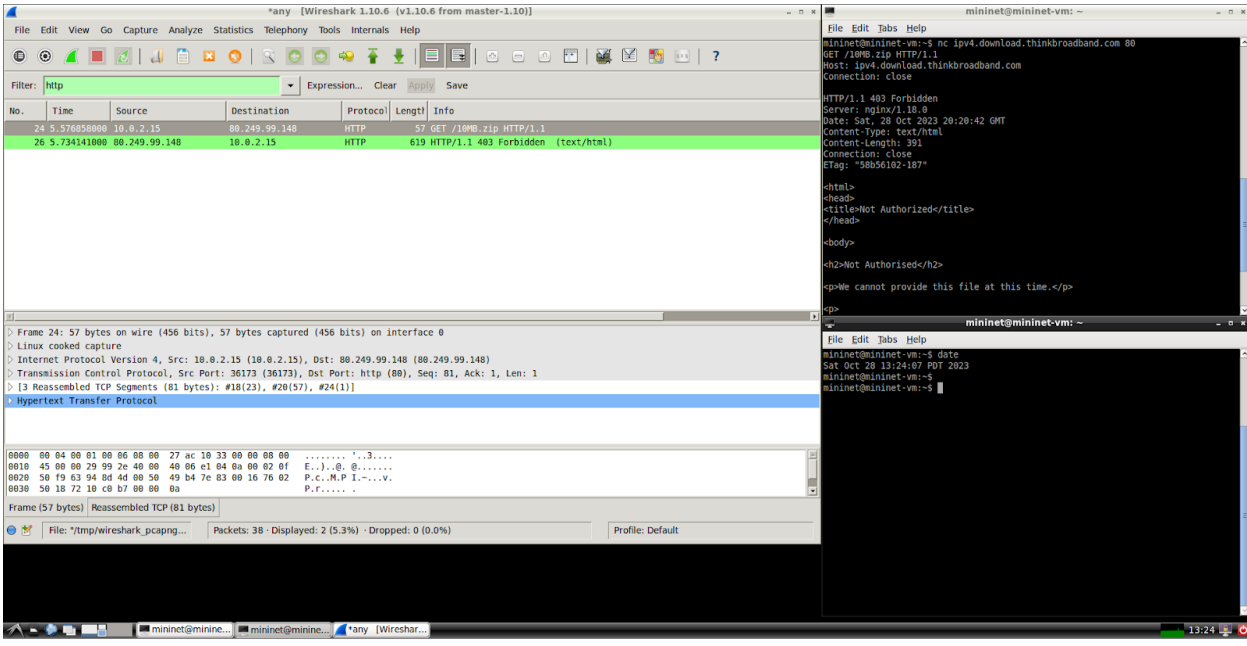
f. CIRCLE TCP IN THE SCREENSHOTS ABOVE. Netcat uses TCP for data transfer. This is because TCP is more reliable than UDP. Please refer to the circles in the screenshots about to verify.

g. When analyzing a packet that was transferred. I found that the source IP address was 127.0.0.1 and the destination IP address was also 127.0.0.1. This makes sense because netcat is transferring locally so it would make sense that the IP addresses are the same to make sure that the data stays within the local network.

14.

a. The first line I run in the terminal is nc ipv4.download.thinkbroadband.com 80. This line opens up stdin (standard input) for writing for the url and port it was run with. Note port 80 because that is a common HTTP port. After the standard input is open, I run the next 3 lines:
GET /10MB.zip HTTP/1.1
Host: www.ipv4.download.thinkbroadband.com
Connection: close
The purpose of the GET line is to detail what is being asked by the client (me). The next line details the webpage where to ask to find the link of the file to be downloaded. The final line specifies that once the request is done, the connection is to be closed.

b.



The file distribution time is quite short. This is due to the fact that despite working on the lab computers for this problem, I was experiencing a 403 Forbidden error when trying to download the file. I tried to work around this error but I couldn't figure it out.

File distribution time: 0.157283 seconds

c.