

# Lab 3 - DNS and TCP

Total Points: 100

**Platform:** This entire lab is to be run on the Mininet VM. The lab computers are available for all of these problems.

**Before getting started:** We highly recommend reading Section 2.4: Domain Name Service (DNS) in the textbook before beginning this assignment.

## Submission:

**Important Announcement about Gradescope submissions:** When submissions are made on Gradescope, the page/region for each problem MUST be marked. We will not grade submissions that are not marked. If you do not know how to do this, ask for help. If a submission does not have the region/page marked, you will be asked to resubmit. The recorded date for the assignment will be the resubmission date and late days will be counted.

One PDF file for this assignment is to be submitted directly on Gradescope. Naming convention of the file: [YourCruzID].pdf.

## Screenshots:

For all questions that require a screenshot, make sure that a **date timestamp** is visible next to your results. No credit will be given for screenshots without a timestamp. **Make sure to highlight as necessary in the screenshots to get full credit.**

**References:** Chapter 2, Computer Networking: A Top Down Approach

- Section 2.2: The Web and HTTP
- Section 2.4: Domain Name Service (DNS)
- Chapter 3: Transport Layer
- [RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1](#)
- <https://www.w3.org/Protocols/rfc2616/rfc2616-sec5.html>
- <https://www.ietf.org/rfc/rfc1035.txt>

## [50 pts] Domain Name Service

In this lab, we'll make extensive use of the command line tool **dig**. In its most basic operation, the `dig` tool allows the host running the tool to query any Nameserver for a DNS record. The queried DNS server can be a root DNS server, a top-level-domain DNS server, an authoritative DNS server, or an intermediate DNS server (see the textbook for definitions of these terms).

Example commands (Read man page for more information):

```
dig [domain name] [record type:optional]
```

```
dig www.google.com
```

```
dig www.google.com AAAA
```

## [35 pts] DNS Warmups - Resource Records:

1. Think about the role of the **Resource Records** (RR) in the **DNS** protocol and the role of a **web object** in the **HTTP** protocol.  
What **similarities** do they share in their respective protocols?  
Web object and RR both map hostnames however web objects have URLs and RR map to IP addresses, they both carry info about what their protocol is doing, and they both get cached for faster access
2. Run the command **dig www.santacruz.org**
  - a. What **type** of Resource Record(s) are returned and **what is their Function?**  
By default (according to the man page in the mininet vm, it will run an A Query. This will return the address record and domain name
  - b. What is the purpose of the **Time to Live (TTL)** field? To reflect the changes of the of the RR's since the last time you cached, and just caching data in general  
What is the **TTL** value of the returned Resource Record(s)? 18msec

Take a **timestamped screenshot** and highlight your answers for **2(a)** and **2(b)**

```
mininet@mininet-vm: ~  
File Edit Tabs Help  
Sun Feb  4 20:24:00 PST 2024  
mininet@mininet-vm:~$ dig www.santacruz.org  
  
; <<> DiG 9.9.5-3ubuntu0.19-Ubuntu <<> www.santacruz.org  
;; global options: +cmd  
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36712  
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1  
  
;; OPT PSEUDOSECTION:  
;; EDNS: version: 0, flags:; udp: 512  
;; QUESTION SECTION:  
;www.santacruz.org.          IN      A  
  
;; ANSWER SECTION:  
www.santacruz.org.          1443    IN      A      162.159.135.42  
  
;; Query time: 18 msec  
;; SERVER: 172.16.0.1#53(172.16.0.1)  
;; WHEN: Sun Feb 04 20:24:02 PST 2024  
;; MSG SIZE rcvd: 62  
  
mininet@mininet-vm:~$ dig www.santacruz.org | grep TTL  
mininet@mininet-vm:~$
```

3. **CNAME** records are used to map domains to aliases.

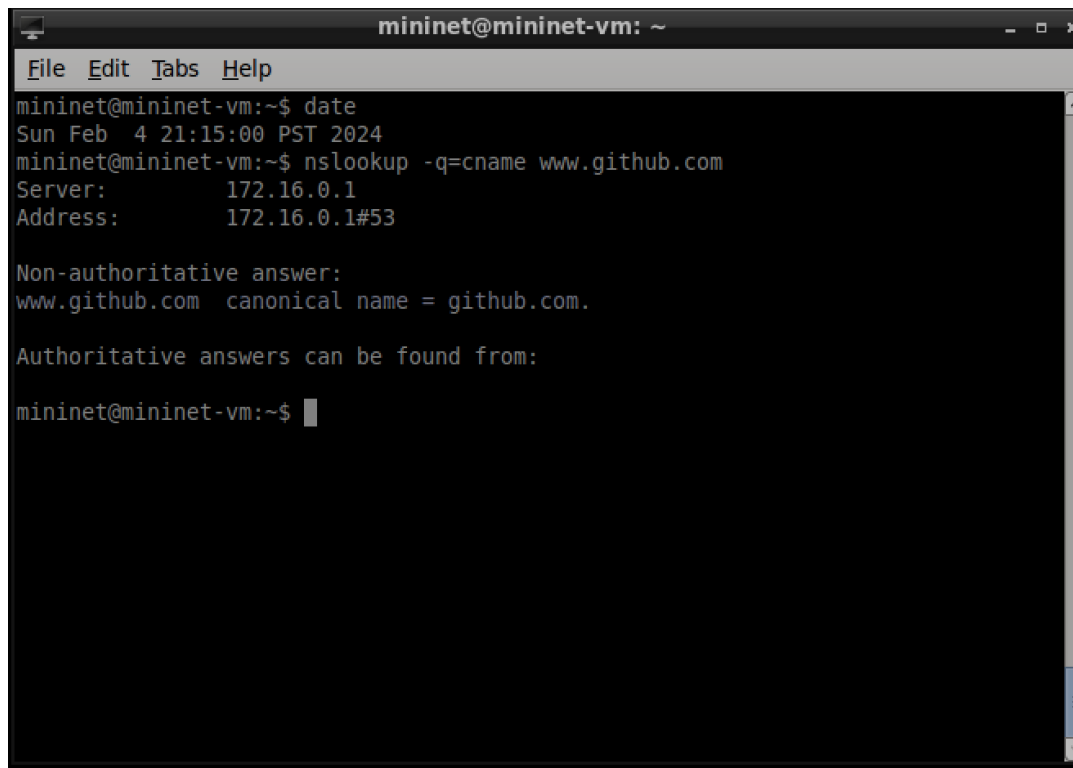
- a. Why are domain **aliases** used on the Internet?

Because it makes owning a domain much easier because no matter which TLD someone puts in you can reroute them to your website without reloading.

- b. Run a command to find the **CNAME** record for **www.github.com**. Which name is the **alias**? Which name is the **canonical** name?

[www.google.com](http://www.google.com) is the alias name and github.com is the canonical name

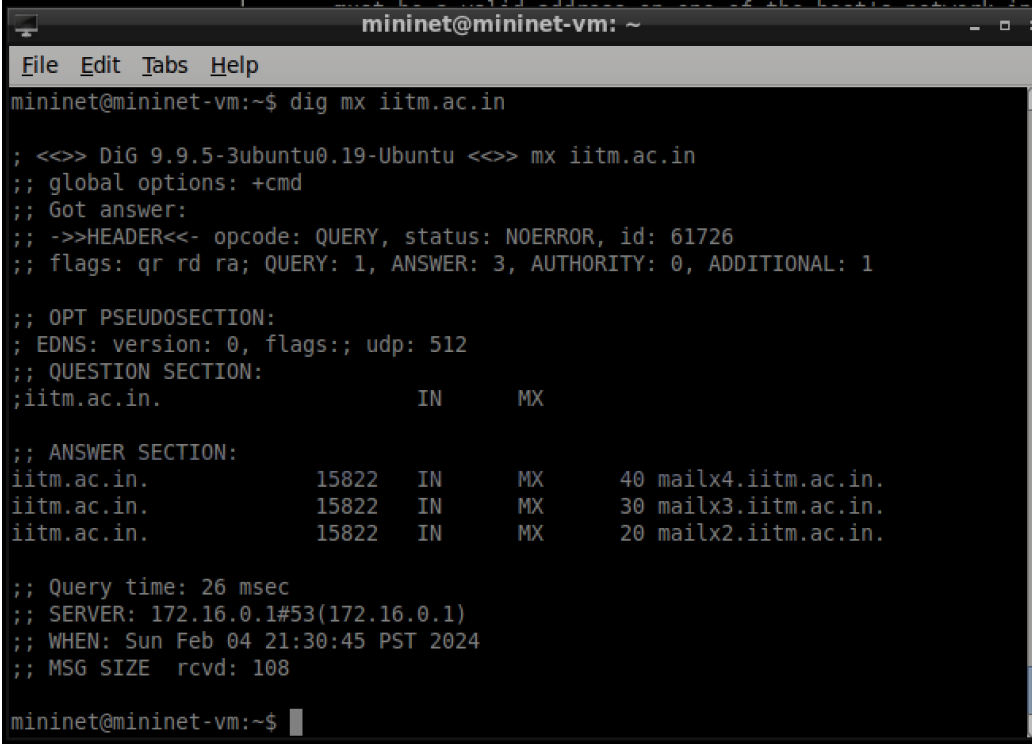
Take a **timestamped screenshot** of the command and output – highlight the **alias** and **canonical** name to support your answer.



```
mininet@mininet-vm: ~  
File Edit Tabs Help  
mininet@mininet-vm:~$ date  
Sun Feb  4 21:15:00 PST 2024  
mininet@mininet-vm:~$ nslookup -q=cname www.github.com  
Server:      172.16.0.1  
Address:     172.16.0.1#53  
  
Non-authoritative answer:  
www.github.com canonical name = github.com.  
  
Authoritative answers can be found from:  
  
mininet@mininet-vm:~$
```

4. Run **dig** to find the **MX** resource record of **iitm.ac.in**
  - a. What information does an **MX** resource record provide?  
An MX or Mail Exchange record lets you know what mail server is chosen for that domain
  - b. What **command** did you use to obtain the **MX** resource records for the given domain? Dig mx **iitm.ac.in**
  - c. Based on the output of the MX query, which mail server do you think your computer would contact when sending an email to [someone@iitm.ac.in](mailto:someone@iitm.ac.in)? Explain your answer. Mailx4.iitm.ac.in as that seems to be the first one responding that it is ready to receive.

Take a **timestamped screenshot** of the command and output – answers in (b) and (c) to support your answer.



```
mininet@mininet-vm: ~  
File Edit Tabs Help  
mininet@mininet-vm:~$ dig mx iitm.ac.in  
  
; <<>> DiG 9.9.5-3ubuntu0.19-Ubuntu <<>> mx iitm.ac.in  
;; global options: +cmd  
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61726  
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1  
  
;; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags:; udp: 512  
;; QUESTION SECTION:  
; iitm.ac.in.                IN      MX  
  
;; ANSWER SECTION:  
iitm.ac.in.      15822   IN      MX      40 mailx4.iitm.ac.in.  
iitm.ac.in.      15822   IN      MX      30 mailx3.iitm.ac.in.  
iitm.ac.in.      15822   IN      MX      20 mailx2.iitm.ac.in.  
  
;; Query time: 26 msec  
;; SERVER: 172.16.0.1#53(172.16.0.1)  
;; WHEN: Sun Feb 04 21:30:45 PST 2024  
;; MSG SIZE rcvd: 108  
  
mininet@mininet-vm:~$
```

- d. Think about accessing the domain in (a) in your browser compared to sending an email to a person with an email account in the domain (e.g., [www.ucsc.edu](mailto:sammyslug@ucsc.edu) in your browser versus sammyslug@ucsc.edu).

What corresponding **DNS** queries are made, how are they **different**?

When accessing the website you only use A or AAAA records but mail also uses an MX on top of that, also websites query static content meanwhile email uses dynamic messages which means that continuous queries happen while the email is being sent rather than just at the beginning like a website.

What **DNS** mechanisms/services are being used in this example?

Records, queries, caching,

## 5. Authoritative name servers

- a. What is an Authoritative name server and why is it **needed**?

It is a source of data that has IP addresses for domains. It is run by an official source that keeps people from adding fake IP addresses to domains in order to trick you. It is needed for a multitude of reasons, one being that remembering the IP address to everything is too much for most people, second IP addresses don't always stay the same, so this allows you to just put in the web address and it will point you to the correct IP. Lastly it makes it much harder to spoof domains.

- b. Run **dig** to determine the authoritative **DNS** servers for a university in

**South America**. What is the name of the university you chose?

Universidad de Buenos Aires

Take a **timestamped screenshot** highlighting your results (the Authoritative NS).

```
mininet@mininet-vm: ~  
File Edit Tabs Help  
Sun Feb  4 22:27:53 PST 2024  
mininet@mininet-vm:~$ dig https://www.uba.ar/  
  
; <<> DiG 9.9.5-3ubuntu0.19-Ubuntu <<> https://www.uba.ar/  
;; global options: +cmd  
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 45155  
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1  
  
;; OPT PSEUDOSECTION:  
;; EDNS: version: 0, flags:;, udp: 512  
;; QUESTION SECTION:  
;https://www.uba.ar/.          IN      A  
  
;; AUTHORITY SECTION:  
          86399  IN      SOA      a.root-servers.net. nstld.verisi  
gn-grs.com. 2024020500 1800 900 604800 86400  
  
;; Query time: 23 msec  
;; SERVER: 172.16.0.1#53(172.16.0.1)  
;; WHEN: Sun Feb 04 22:27:57 PST 2024  
;; MSG SIZE rcvd: 123  
  
mininet@mininet-vm:~$
```

- c. Suppose the university wanted to further partition the **DNS** namespace. For example, the **engineering** and **biology** departments want to be in charge of their own **namespace** (e.g., names of computers in their respective departments). How could **Authoritative nameservers** be used to accomplish this task? You could easily do this with subdomains, you would just tell the authoritative nameserver the new IP of the subdomain (being engineering.soe.uba.ar).
6. Run 'dig x.com ANY'  
Make sure you are doing this from the ucsc domain or 'ANY' may not work as expected.
  - a) What is the purpose of ANY?  
It returns any and all responses it gets when it sends out every check

b) Why are multiple A records returned? Explain and discuss the additional service DNS is performing by providing these multiple records and explain its importance. It seems that they have multiple Ips, this could be because they load balance or because they have multiple domain aliases, This would help with popular websites or if a website has multiple servers in different parts of the world and they want to send you to one thats closer to you.

7. Open Wireshark and listen on the 'any' interface. Open a terminal, and clear your dns cache using the command `systemd-resolve --flush-caches` (new VM) or `sudo /etc/init.d/dns-clean restart` (old VM).

Then, use `dig` to perform a DNS lookup to [kicker.de](http://kicker.de) **two consecutive times**.

- a) Does the DNS query take the same amount of time to run both times?  
No it doesn't take very long for the second one compared to the first which took significantly longer
- b) With respect to what you know about DNS, provide an explanation for the difference in lookup time - i.e. what is different between the two queries? Most likely the second one had a bunch of data cached already so it could easily find the info without asking the Authoritative name server. Which would make sense as you had us clear our cache earlier, the first needed to pull all new data and then cached it, the second however already had it all.

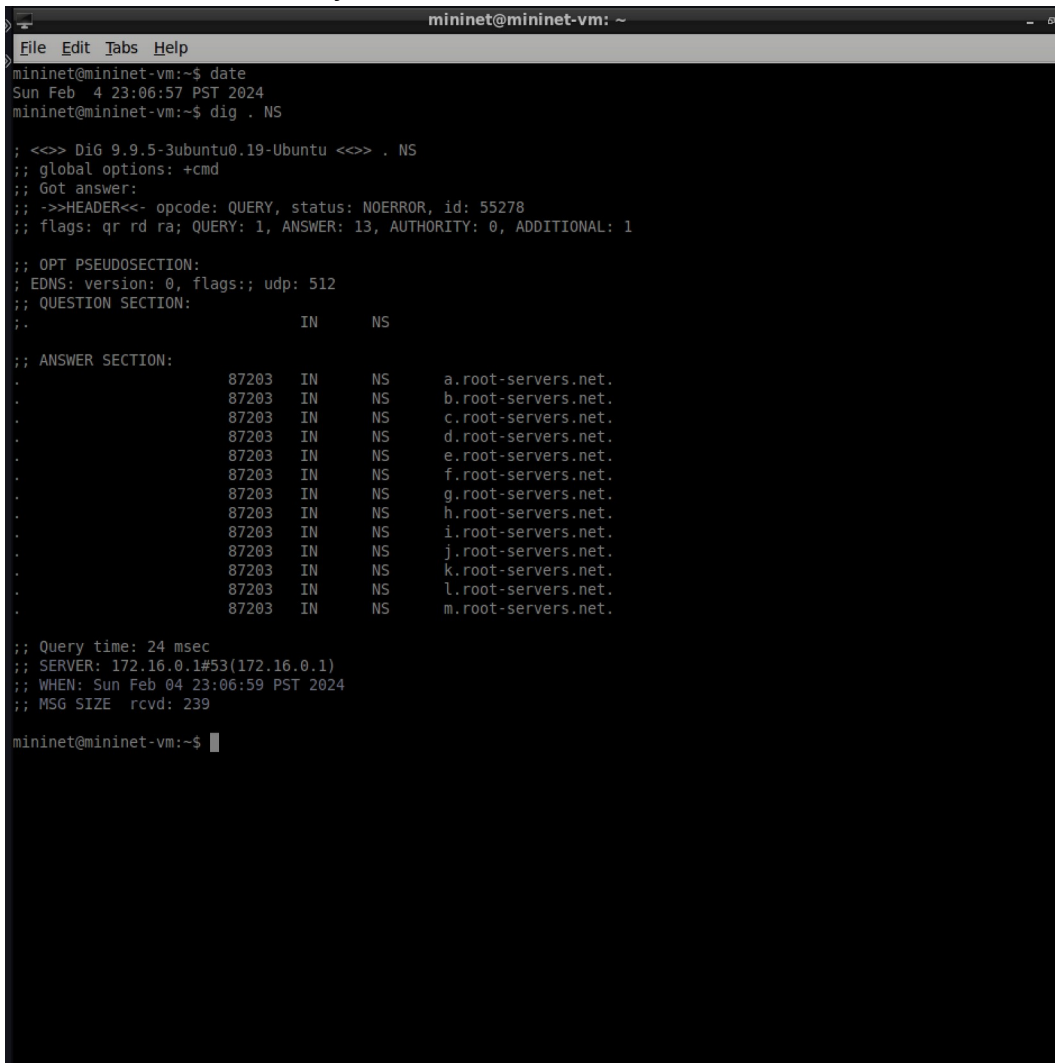
8. Root Name Servers

- a. What are root name servers?

They are the servers that hold all records for .com, .org, .net, etc

- b. What command is used to find the root name servers?  
`Dig . NS`

- c. Run the command and include a **timestamped screenshot of the results** and discuss what you see.



```
mininet@mininet-vm: ~  
File Edit Tabs Help  
mininet@mininet-vm:~$ date  
Sun Feb  4 23:06:57 PST 2024  
mininet@mininet-vm:~$ dig . NS  
  
; <<> DiG 9.9.5-3ubuntu0.19-Ubuntu <<> . NS  
;; global options: +cmd  
;; Got answer:  
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 55278  
;; flags: qr rd ra; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 1  
  
;; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags:;, udp: 512  
;; QUESTION SECTION:  
; .                IN      NS  
  
;; ANSWER SECTION:  
.                87203   IN      NS      a.root-servers.net.  
.                87203   IN      NS      b.root-servers.net.  
.                87203   IN      NS      c.root-servers.net.  
.                87203   IN      NS      d.root-servers.net.  
.                87203   IN      NS      e.root-servers.net.  
.                87203   IN      NS      f.root-servers.net.  
.                87203   IN      NS      g.root-servers.net.  
.                87203   IN      NS      h.root-servers.net.  
.                87203   IN      NS      i.root-servers.net.  
.                87203   IN      NS      j.root-servers.net.  
.                87203   IN      NS      k.root-servers.net.  
.                87203   IN      NS      l.root-servers.net.  
.                87203   IN      NS      m.root-servers.net.  
  
;; Query time: 24 msec  
;; SERVER: 172.16.0.1#53(172.16.0.1)  
;; WHEN: Sun Feb 04 23:06:59 PST 2024  
;; MSG SIZE rcvd: 239  
  
mininet@mininet-vm:~$
```

I see 13 different name servers, it also seems like it only needed to query once to get all the data on the name servers, and doing so took a very short amount of time.

- d. Is your access network physically close to a root name server? Explain how you have come to your conclusion. Yes, I tracerouted to one of them and because of both low ping time and few hops, I think I am

9. Treasure hunt! di:

What command would you use to run a reverse DNS lookup on IP Address 171.67.215.200? What domain is associated with the address 171.67.215.200?

Dig -x, web.stanford.edu

## [15 pts] Digging in with Wireshark - the DNS Messages

For the next two questions, we will observe how the DNS protocol operates at the packet level by using the Mininet VM and Wireshark.

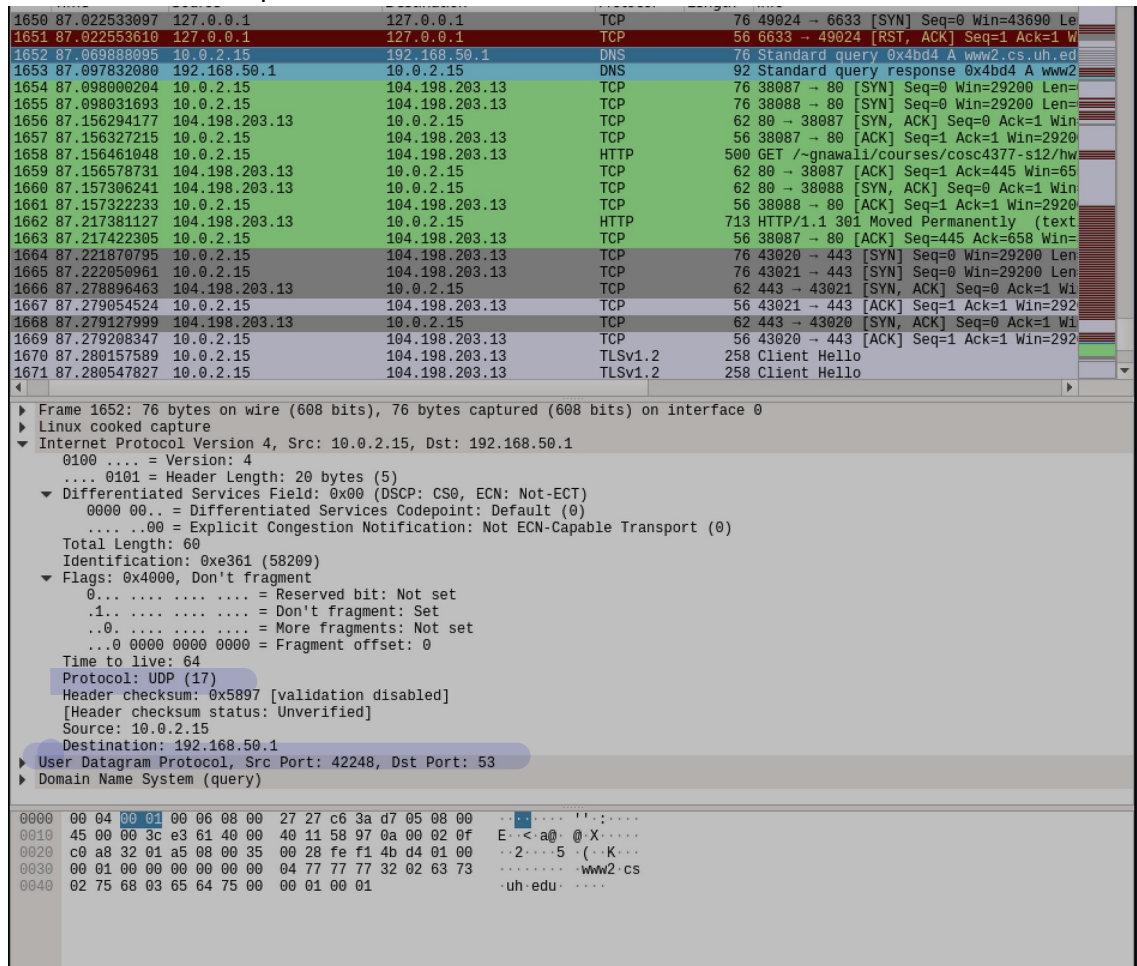
Begin by doing the following:

- Open Wireshark and listen on the 'any' interface without any filter
- Open Chromium, clear your web cache and navigate to



# 10. Wireshark capture of DNS Messages:

Show a **timestamped screenshot** in Wireshark and highlight the areas in your answers to the questions below:



- Find the DNS query and response messages. Are these messages sent with UDP or TCP? Circle the transport layer protocol in your screenshot.

UDP

- Discuss the transport layer protocol observed in (a) - why do you think the one you observed is used? Highlight in your screenshot. It most likely uses it because it doesn't need to keep a stable connection like tcp and because it can send multiple at a time its safer.

- What is the IP address of the local nameserver contacted by your client? Show in the screenshot.192.168.50.1

Look for the source and destination ports of the **DNS queries**:

		Well known port? yes/no
Source Port Number	42248	no
Destination Port Number	53	yes

- d. Is this destination port number what you expected? Why or why not? Highlight this port in your screenshot. Yes it is the DNS port
- e. If you were to close your browser and then access the site again, do you expect your source port number to stay the same? Explain your answer. Highlight the port number in your screenshot.  
Probably not because chrome is just finding a random port.

- f. Examine the DNS response message. How many “answers” are provided? What information is provided in each of these “answers”? Do you see a final answer or only referrals? What is the “final answer”? i.e. what is the IP address returned by DNS? 104.198.203.13

**11. Continuation of Q10 – Loading the webpage and Wireshark capture:**

- a. After DNS is finally resolved, we expect that a TCP connection should be opened before the HTTP request is sent. Look at a TCP packet and determine the source and destination addresses and ports. Fill in the table below.

IP Addresses:		
Source IP Address		10.0.2.15
Destination IP Address		104.198.203.13
Ports:		
		Well known port? yes/no
Source Port Number	38087	No
Destination Port Number	80	yes

- b. Web Server IP Address: Referring to the table above, Is the IP Address of the web server the same as what was returned in Q10(f)? Explain why or why not.  
Yes, because the web server must stay the same to download the file
- c. Why is the destination IP address for the HTTP request different from the destination IP address used by DNS queries? Because of DNS resolution
- d. What kind of file is downloaded from the web server? How do you know?  
A .zip file, because when I ls the file it says .zip

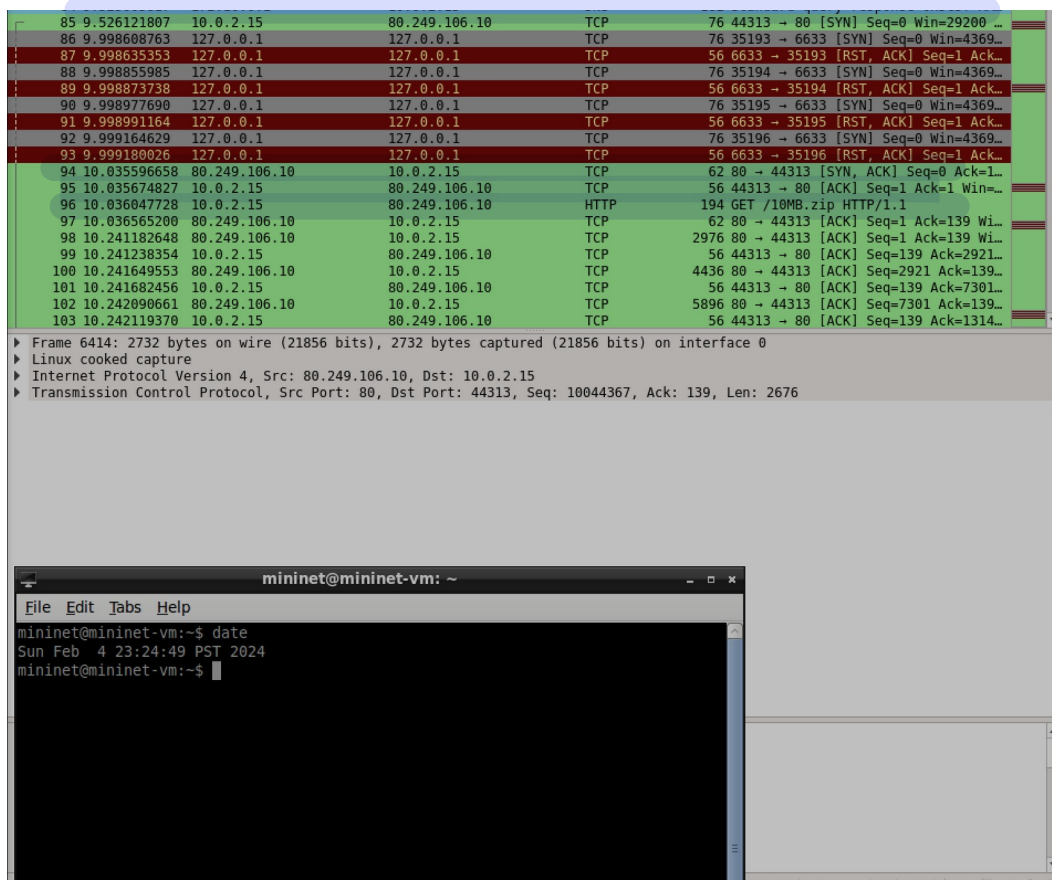
## [50 pts] Transport Control Protocol - TCP

In this section we look at a few different ways to transfer files and observe the transfers over Wireshark. We will give an express TCP overview in class, but we won't officially reach this topic until we study the Transport Layer in Chapter 3 (a good reference if you have questions).

### 12. File Transfer using **wget** from an origin server

Open Wireshark and listen on the 'any' interface. Then using **wget**, download <http://ipv4.download.thinkbroadband.com/10MB.zip> in a terminal window.

- Enumerate and describe all of the different protocols you see in your Wireshark capture from the time you enter the URL leading up to and including the file download (Ignore NTP packets).
  - DNS
  - Tcp
  - http
- Find the **TCP three-way handshake** for connection establishment. Take a **timestamped screenshot** of these TCP segments and highlight all of them in the screenshot.



- TCP Segments:** After the handshake is complete, find two TCP segments (packets) – one indicating the start of the file download and then the final packet in the transfer.
  - How many different TCP segments are transmitted to send the entire

file? How did you discover this in Wireshark? 241, in Wireshark's statistics you can access the conversations tools and see unique conversations

- What is the typical segment size? 14,656
- What value do you expect if you multiply the number segments transmitted by the typical number of bytes in a segment? Try it! Show your work and explain. Multiply the number of segments transmitted by the typical segment size to find the total amount of data transferred.  
 $241 * 14656 = 3532096$

d. **File Distribution Time according to Wireshark:** Using the two packets in (c), calculate the download time using the Time column in Wireshark.  
3.27

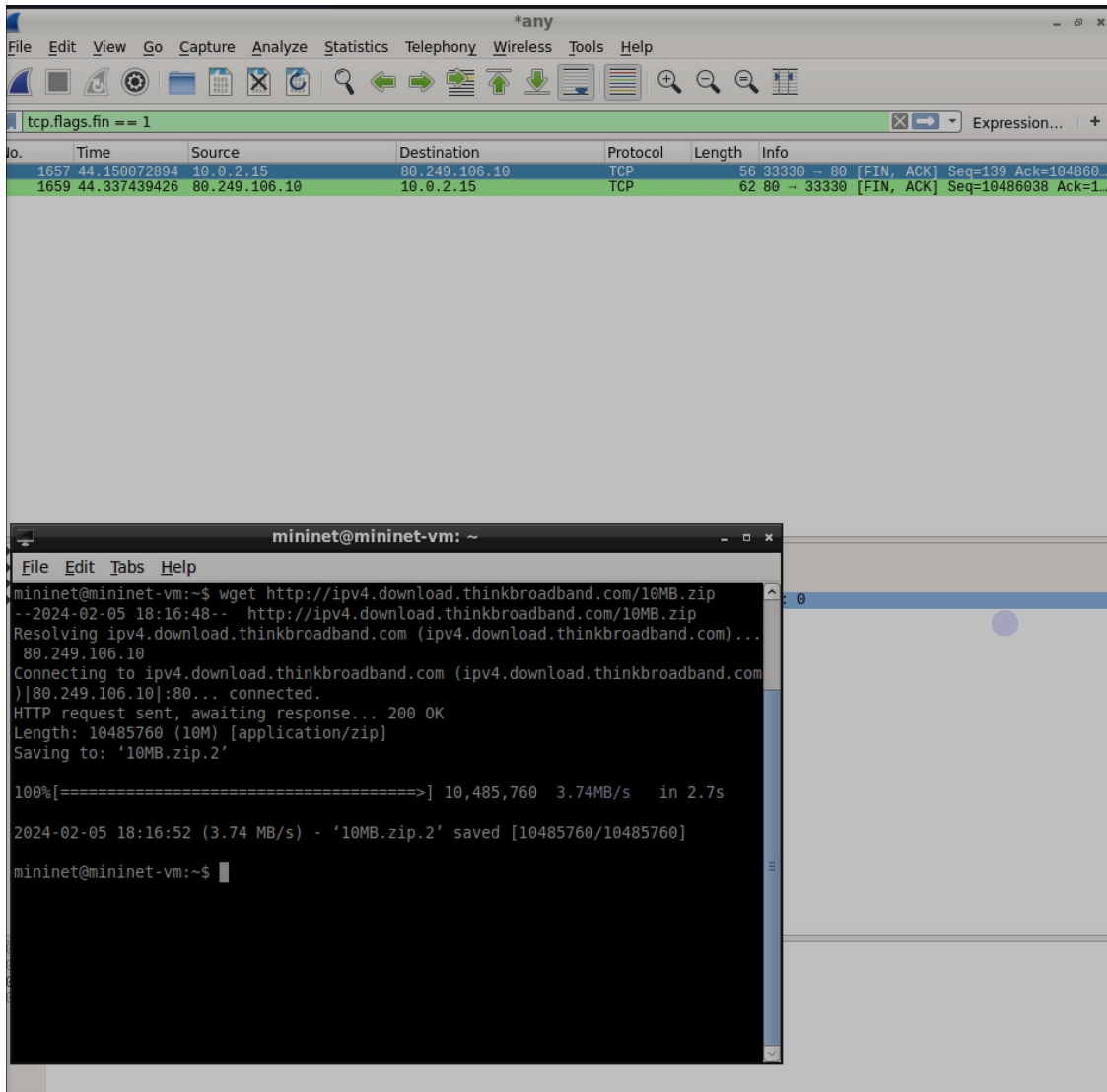
e. **Throughput according to Wireshark:** Using (d), calculate the average throughput seen by the Client and enter it in the table below. Show all of your work. Enter the throughput in the table below.  $14,656 / 3.27 = 4,482$

f. Enter the throughput as reported in the wget terminal in the table below. How does the value compare to the throughput calculation in (e)? Is it close? Discuss and account for any differences.

Application	Throughput	Notes (if any)
Wireshark capture of wget	4482	
Terminal output of wget	3740	

**Include two timestamped screenshots:**

- Wireshark output highlighting packets and values used to calculate (d) and (e)
- Terminal output of `wget` highlighting the reported download speed

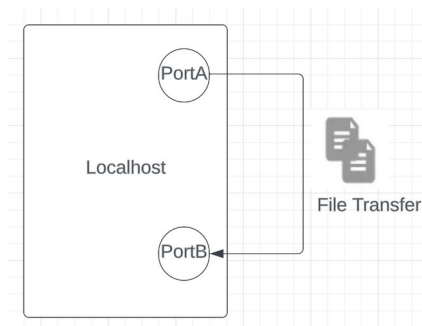


The image shows a Wireshark packet capture interface. The main pane displays a list of network packets. The selected packet (16116) is an HTTP 200 OK response from 80.249.106.10 to 10.0.2.15. The packet details pane shows the HTTP response structure, including the status line 'HTTP/1.1 200 OK (application/zip)'. The packet bytes pane shows the raw data in hexadecimal and ASCII. In the foreground, a terminal window titled 'mininet@mininet-vm: ~' displays the command 'date' and its output 'Mon Feb 5 18:42:22 PST 2024'.

No.	Time	Source	Destination	Protocol	Length	Info
1636	44.145969312	10.0.2.15	80.249.106.10	TCP	56	33330 → 80 [ACK] Seq=139 Ack=1030...
1637	44.146934967	80.249.106.10	10.0.2.15	TCP	26336	80 → 33330 [ACK] Seq=10306449 Ack...
1638	44.147153896	80.249.106.10	10.0.2.15	TCP	17576	80 → 33330 [ACK] Seq=10332729 Ack...
1639	44.147280613	80.249.106.10	10.0.2.15	TCP	112	80 → 33330 [PSH, ACK] Seq=1035024...
1640	44.147513840	10.0.2.15	80.249.106.10	TCP	56	33330 → 80 [ACK] Seq=139 Ack=1035...
1641	44.147678175	80.249.106.10	10.0.2.15	TCP	8816	80 → 33330 [ACK] Seq=10350305 Ack...
1642	44.147683933	10.0.2.15	80.249.106.10	TCP	56	33330 → 80 [ACK] Seq=139 Ack=1035...
1643	44.147732456	80.249.106.10	10.0.2.15	TCP	20496	80 → 33330 [ACK] Seq=10359065 Ack...
1644	44.147768018	80.249.106.10	10.0.2.15	TCP	14656	80 → 33330 [ACK] Seq=10379505 Ack...
1645	44.147810503	80.249.106.10	10.0.2.15	TCP	17576	80 → 33330 [ACK] Seq=10394105 Ack...
1646	44.147827216	10.0.2.15	80.249.106.10	TCP	56	33330 → 80 [ACK] Seq=139 Ack=1041...
1647	44.147890759	80.249.106.10	10.0.2.15	TCP	4436	80 → 33330 [PSH, ACK] Seq=1041162...
1648	44.147895560	10.0.2.15	80.249.106.10	TCP	56	33330 → 80 [ACK] Seq=139 Ack=1041...
1649	44.147974177	80.249.106.10	10.0.2.15	TCP	10276	80 → 33330 [ACK] Seq=10416005 Ack...
1650	44.148014121	80.249.106.10	10.0.2.15	TCP	16116	80 → 33330 [ACK] Seq=10426225 Ack...
1651	44.148043796	10.0.2.15	80.249.106.10	TCP	56	33330 → 80 [ACK] Seq=139 Ack=1042...
1652	44.148061371	80.249.106.10	10.0.2.15	TCP	19036	80 → 33330 [ACK] Seq=10442285 Ack...
1653	44.148095187	80.249.106.10	10.0.2.15	TCP	16116	80 → 33330 [PSH, ACK] Seq=1046126...
1654	44.148122631	10.0.2.15	80.249.106.10	TCP	56	33330 → 80 [ACK] Seq=139 Ack=1047...
1655	44.148254322	80.249.106.10	10.0.2.15	HTTP	8769	HTTP/1.1 200 OK (application/zip)
1656	44.148260214	10.0.2.15	80.249.106.10	TCP	56	33330 → 80 [ACK] Seq=139 Ack=1048...
1657	44.150072894	10.0.2.15	80.249.106.10	TCP	56	33330 → 80 [FIN, ACK] Seq=139 Ack...

### 13. File Transfer using netcat locally

In this problem we will use the `netcat` utility to perform a local file transfer of the file `10MB.zip`. One terminal is set up for listening and the other for sending as shown below:

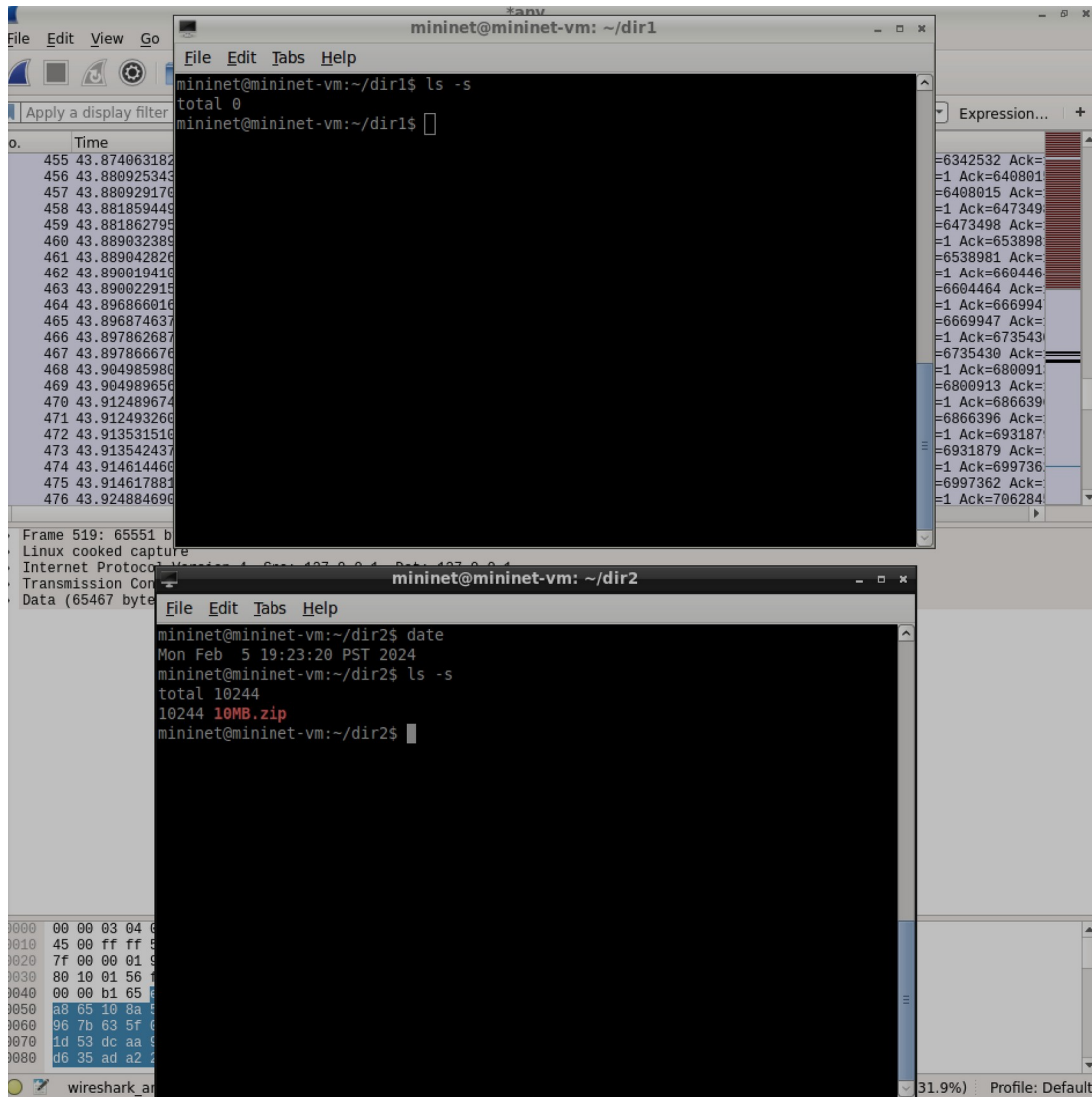


- a. Research the netcat tool and briefly describe in your own words what it does.

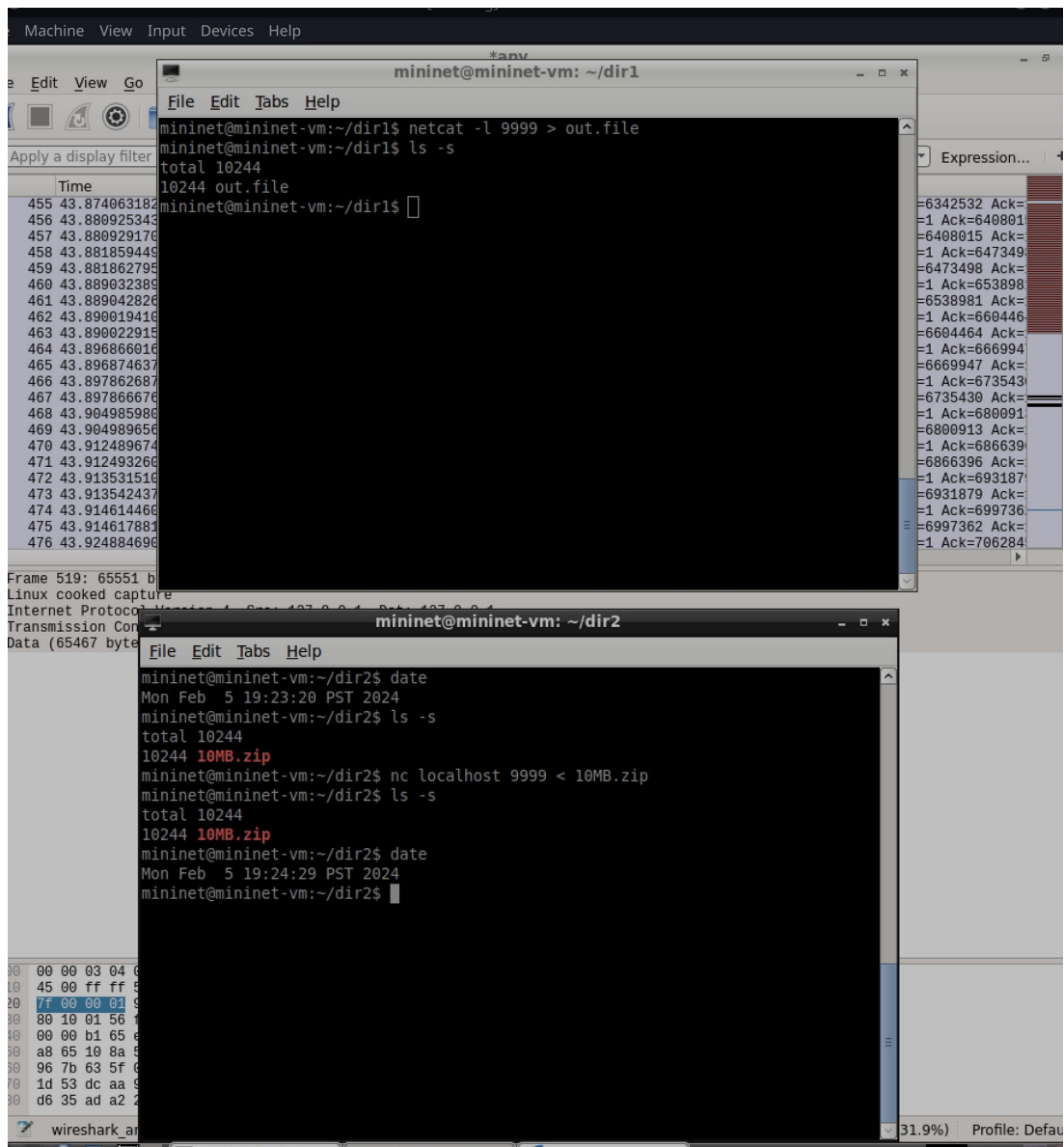
Follow the procedure outlined below to accomplish the transfer of the 10MB.zip file:

- Open Wireshark to capture the packet transfer
- Create two directories dir1 and dir2
- Open 2 terminals on your VM – one for each of the directories (let's refer to them as terminal 1 with dir1 and terminal 2 with dir2)
- Download <http://ipv4.download.thinkbroadband.com/10MB.zip> and save it in dir2.
- On terminal 1 inside dir1, run: `netcat -l 9999 > out.file`
- On terminal 2 inside dir2, run: `nc localhost 9999 < 10MB.zip`

- b. Take a **timestamped screenshot** showing the two terminal directory listings before and after the commands are run. Make sure the directory listing displays dates and file size.

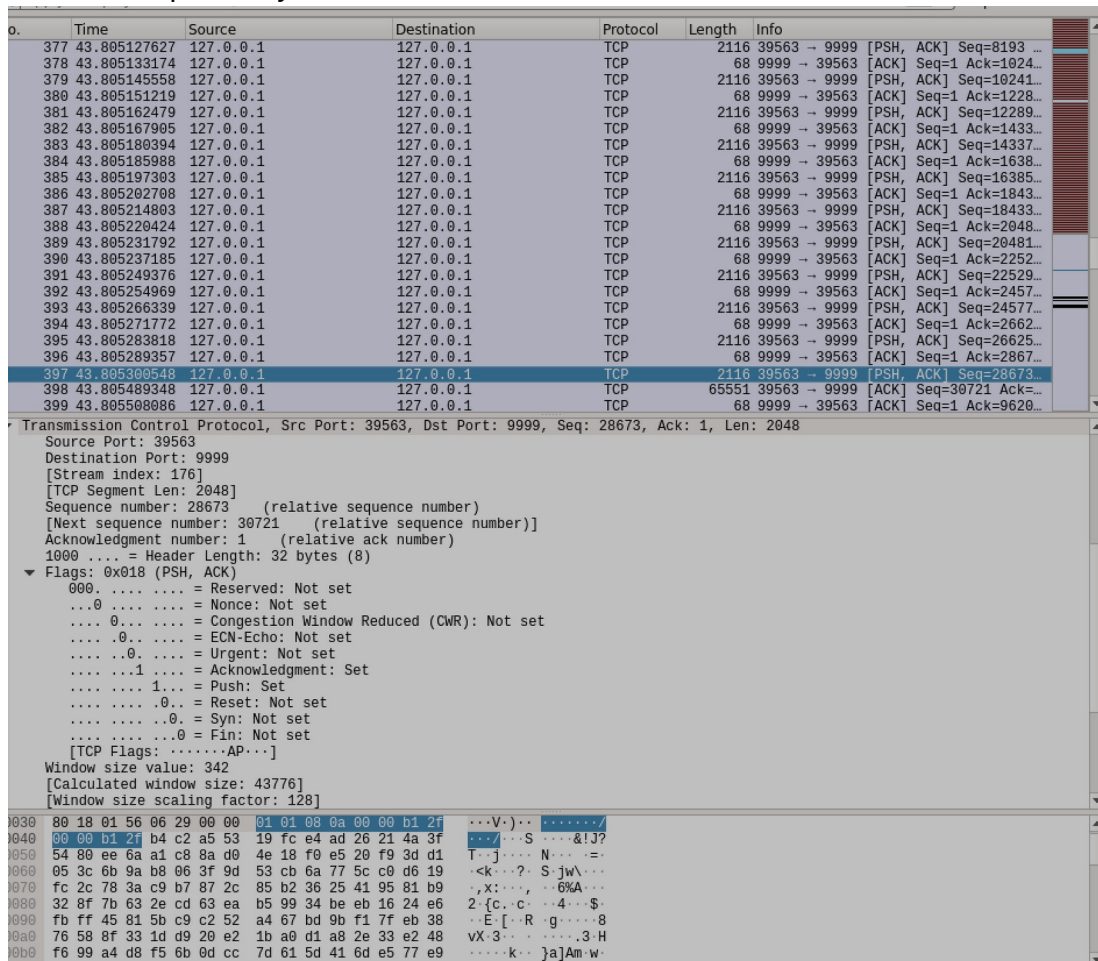






- c. These small questions make sure you understand the transfer:
  - Which terminal is sending the file? Terminal 2
  - Which one is receiving it? Identify the Client and the Server. Terminal 1
  - Which command initiates the data transfer? Nc localhost
  - For this transfer to work, which command must be run first? Why?
  - Netcat, so that it opens a listening port for the sender
  - What happens if terminal 2 is executed first?
  - It sends but nothing is recieved
- d. Explain this data transfer in your own words  
 the NC command opens a listening port and then the netcat connects to the open port 9999 and sends tcp pushes.
- e. If the above transfer was done in HTTP, what would be the corresponding HTTP method that accomplishes this transfer? It would use PUT commands to transfer files

- f. Use Wireshark to identify the transport layer protocol `netcat` used to transfer the file. Is this what you would expect? Why or why not? Verify in your Wireshark screenshot, markup and explanation. PUSH, I would expect this as its the simplest way to do it



- g. Choose one of the packets transferred and drill down to find the source and destination IP addresses - circle them in red in the screenshot in (f). Do the addresses make sense to you (you might want to research the loopback address)? Explain. Both are 127.0.0.1, yes because you are sending it to local host.

#### 14. [15 pts] Compare downloads using `wget` and `netcat` from the origin

server Do the following in your VM:

- Open Wireshark and listen on the 'any' interface.
  - Using `netcat`, download the same file used in question 12) (<http://ipv4.download.thinkbroadband.com/10MB.zip>)
- a. What is the complete command to download the file using `netcat` from the origin server? Explain your command - i.e. what each part does. (Hint: Construct a HTTP 1.1 request to download the file – Host header is needed). `echo -e "GET /10MB.zip HTTP/1.1\r\nHost:`

```
ipv4.download.thinkbroadband.com\r\nConnection: close\r\n\r\n" | nc  
ipv4.download.thinkbroadband.com 80 > downloaded_file  
you first make an echo of an http header then use netcat to pull from origin.
```

- b. Is netcat a useful tool? How is it different from using wget?

Yes, it allows you to send files from one device to another using tcp, however it is not very secure, a system like scp would be better. For downloading files from a server it is quite inefficient since you need to make an http header. Wget just does everything automatically

- c. Would you expect the file distribution time and throughput using netcat to be similar to using wget or different? Explain your answer. It should be the same as it uses the same tools to download however it takes longer to write the http header.

### Extra Credit (5 points)

#### 15. Networking meets art – Let's draw what's going on!

Netcat transfer: Draw a simple Client-Server architecture overview of the file transfer in Question 13. Remember that the client and server were set up inside your own VM, so start your drawing with a box representing the VM and label it "Virtual Machine". Show the following elements and the interaction of components involved in the transfer in the drawing:

- The hardware and software components (clients, servers, VM, etc).
- All IP addresses and ports. (refer to Wireshark capture if needed)
- Arrows representing the data flow.

Web Transfer: Suppose now the client requests a web page from an web external server with IP address 220.15.34.30.

- Add this server to your drawing, including the additional intermediate elements (such as Internet connection). Indicate in your drawing the communication

between the client and server. Label the IP address and the port number that the server would be listening on.

You can update your drawing above to show the new communication (don't need a second drawing).

Note: refer to your class notes for sample drawings containing a simple network, with clients and server. Additionally, **hand made drawings are not acceptable for Extra Credit- please take this opportunity to learn how to use drawing software.** You can ask your TA for suggestions!

Timestamped Screenshot Check-off: (just for you to keep track of your progress)

Problem #	Screenshot	Problem #	Screenshot
Q2		Q8	
Q3		Q10	
Q4		Q12 b,f	
Q5		Q13 b, f	