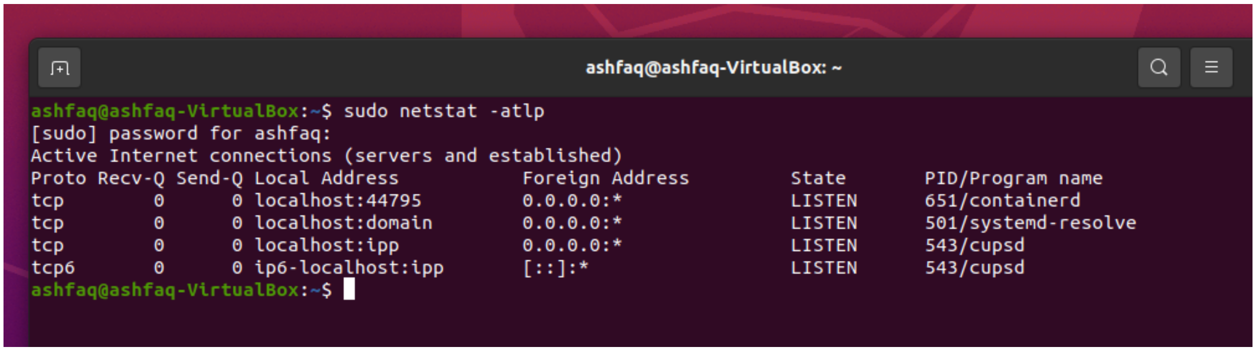
Add Headings (Format > Paragraph styles) and they will appear in your table of contents.

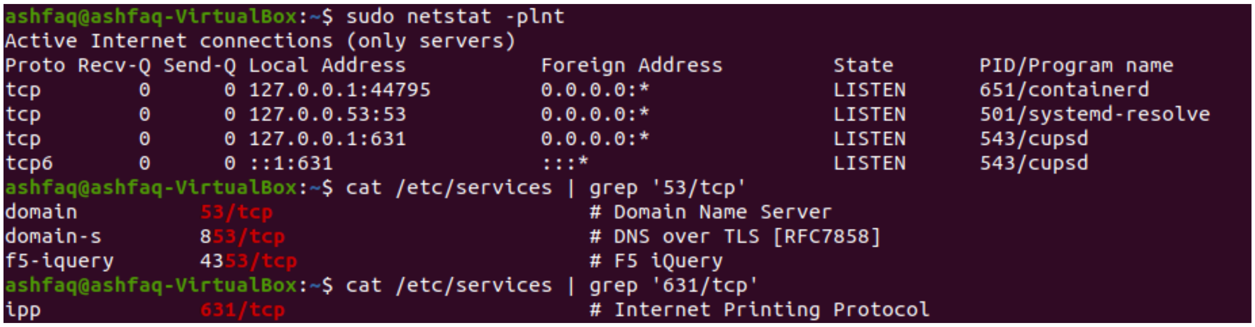
1. TCP #1 (netstat, lsof, nc)

- Run the command using sudo and take a screenshot of the output to include in your lab notebook.



```
ashfaq@ashfaq-VirtualBox:~$ sudo netstat -atlp
[sudo] password for ashfaq:
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 localhost:44795         0.0.0.0:*               LISTEN      651/containerd
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN      501/systemd-resolve
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN      543/cupsd
tcp6       0      0 ip6-localhost:ipp       [::]:*                  LISTEN      543/cupsd
ashfaq@ashfaq-VirtualBox:~$
```

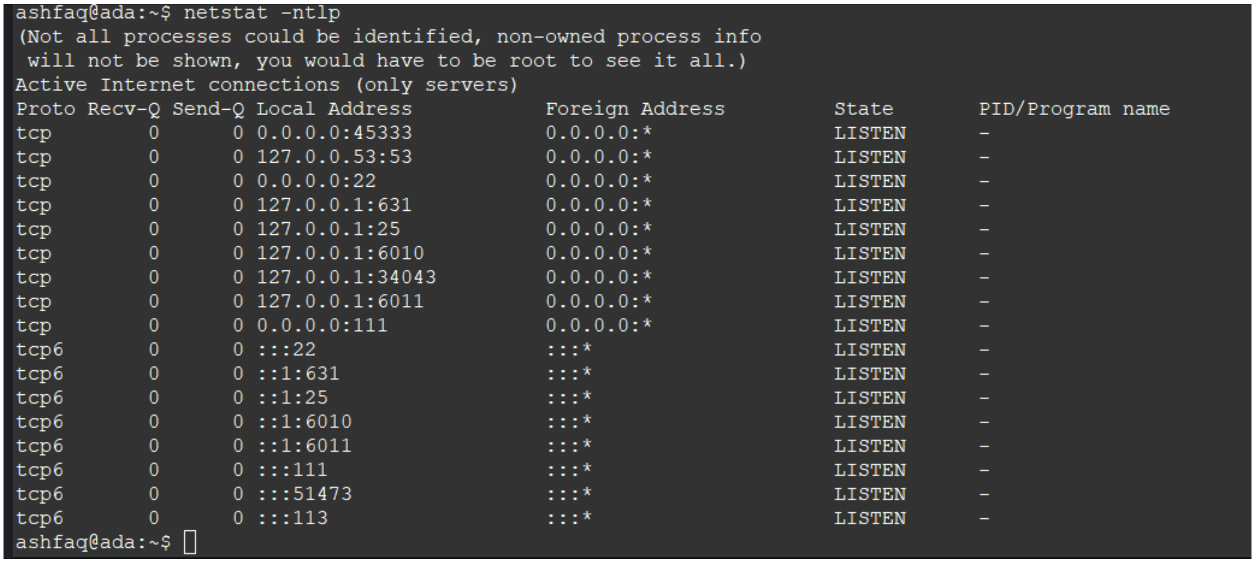
- For port numbers that are named, examine /etc/services and find the port number that corresponds to it. Include this mapping in your lab notebook.



```
ashfaq@ashfaq-VirtualBox:~$ sudo netstat -plnt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:44795         0.0.0.0:*               LISTEN      651/containerd
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      501/systemd-resolve
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      543/cupsd
tcp6       0      0 ::1:631                 :::*                    LISTEN      543/cupsd
ashfaq@ashfaq-VirtualBox:~$ cat /etc/services | grep '53/tcp'
domain          53/tcp                          # Domain Name Server
domain-s        853/tcp                         # DNS over TLS [RFC7858]
f5-iquery       4353/tcp                        # F5 iQuery
ashfaq@ashfaq-VirtualBox:~$ cat /etc/services | grep '631/tcp'
ipp             631/tcp                         # Internet Printing Protocol
```

44795 is an unassigned port.

- For ports that only have a number, what service might it be providing based on the name of the program that is being run?
- Run the netstat command again, but do not use sudo as this is a machine managed by CAT. Include a screenshot of the output.

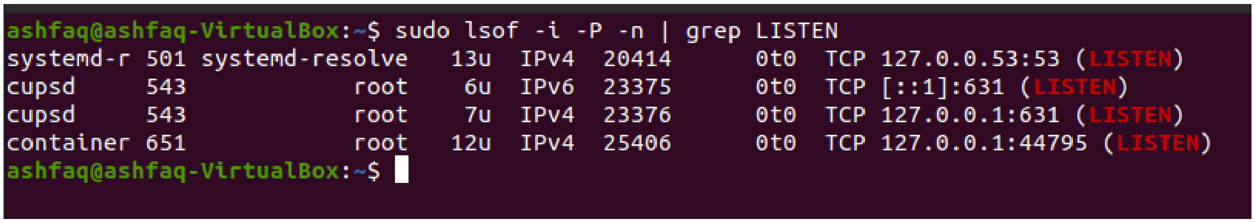


```
ashfaq@ada:~$ netstat -ntlp
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:45333           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6010          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:34043         0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6011          0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
tcp6       0      0 ::1:631                 :::*                    LISTEN      -
tcp6       0      0 ::1:25                  :::*                    LISTEN      -
tcp6       0      0 ::1:6010                :::*                    LISTEN      -
tcp6       0      0 ::1:6011                :::*                    LISTEN      -
tcp6       0      0 :::111                  :::*                    LISTEN      -
tcp6       0      0 :::51473                :::*                    LISTEN      -
tcp6       0      0 :::113                  :::*                    LISTEN      -
ashfaq@ada:~$
```

- What services does this machine provide for external access?

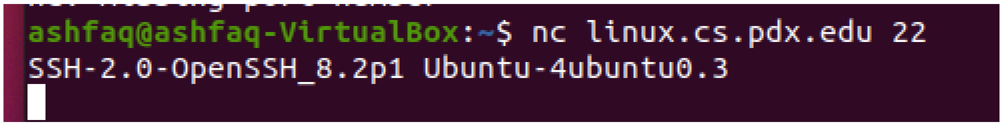
Port 22 is listening, so ssh.

- Use the -i and the -s flag of lsof to generate a listing that is equivalent to the one generated with netstat previously and include it in your lab notebook



```
ashfaq@ashfaq-VirtualBox:~$ sudo lsof -i -P -n | grep LISTEN
systemd-r 501 systemd-resolve   13u  IPv4  20414      0t0  TCP 127.0.0.53:53 (LISTEN)
cupsd     543            root    6u  IPv6  23375      0t0  TCP [::1]:631 (LISTEN)
cupsd     543            root    7u  IPv4  23376      0t0  TCP 127.0.0.1:631 (LISTEN)
container 651            root   12u  IPv4  25406      0t0  TCP 127.0.0.1:44795 (LISTEN)
ashfaq@ashfaq-VirtualBox:~$
```

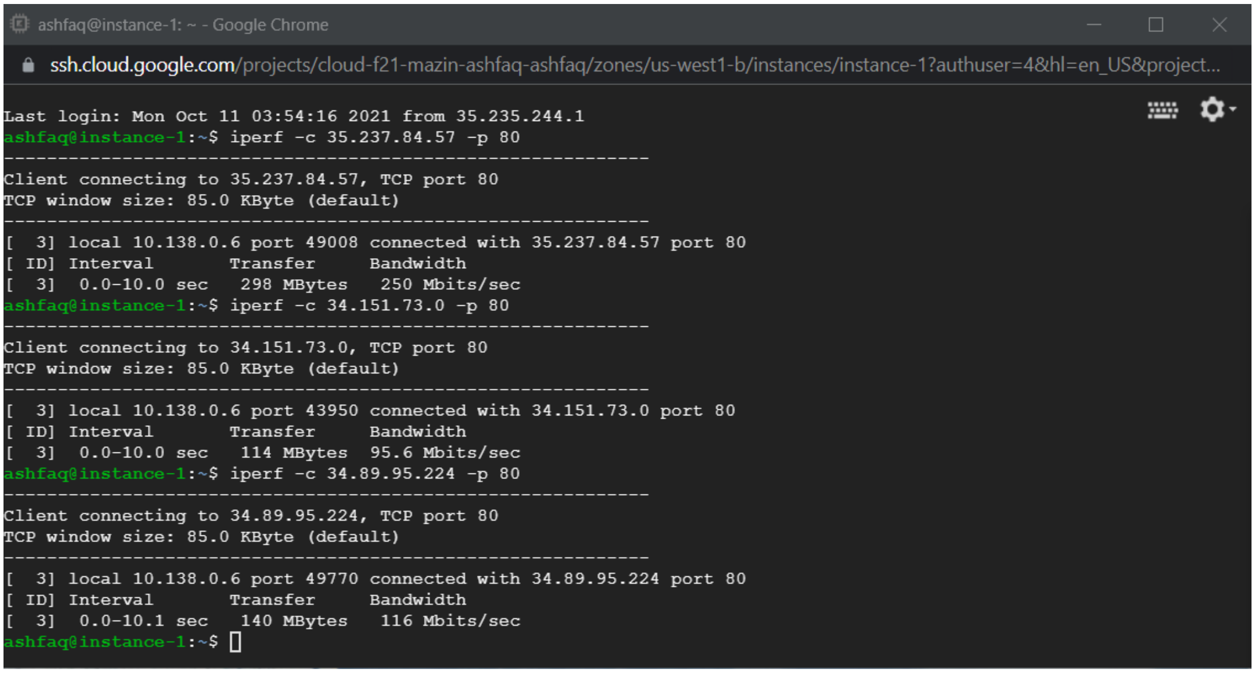
- Include for your lab notebook, the version of ssh that is being used. (Type Ctrl+c to exit)



```
ashfaq@ashfaq-VirtualBox:~$ nc linux.cs.pdx.edu 22
SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.3
```

1. Throughput tests

- Show a screenshot of the measured bandwidth available between your us-west1-b VM and each of the other Compute Engine VMs. Explain the relative differences (or lack thereof) in your results.



ashfaq@instance-1: ~ - Google Chrome
ssh.cloud.google.com/projects/cloud-f21-mazin-ashfaq-ashfaq/zones/us-west1-b/instances/instance-1?authuser=4&hl=en_US&project...

```
Last login: Mon Oct 11 03:54:16 2021 from 35.235.244.1
ashfaq@instance-1:~$ iperf -c 35.237.84.57 -p 80
------------------------------------------------------------
Client connecting to 35.237.84.57, TCP port 80
TCP window size: 85.0 KByte (default)
------------------------------------------------------------
[  3] local 10.138.0.6 port 49008 connected with 35.237.84.57 port 80
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec   298 MBytes   250 Mbits/sec
ashfaq@instance-1:~$ iperf -c 34.151.73.0 -p 80
------------------------------------------------------------
Client connecting to 34.151.73.0, TCP port 80
TCP window size: 85.0 KByte (default)
------------------------------------------------------------
[  3] local 10.138.0.6 port 43950 connected with 34.151.73.0 port 80
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec   114 MBytes  95.6 Mbits/sec
ashfaq@instance-1:~$ iperf -c 34.89.95.224 -p 80
------------------------------------------------------------
Client connecting to 34.89.95.224, TCP port 80
TCP window size: 85.0 KByte (default)
------------------------------------------------------------
[  3] local 10.138.0.6 port 49770 connected with 34.89.95.224 port 80
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.1 sec   140 MBytes   116 Mbits/sec
ashfaq@instance-1:~$ 
```

The further distances have lower bandwidth, meaning the amount of data transported is less.

1. Developer tools

- What is the URL being requested?

  http://google.com/

- What are the Host: (HTTP 1.1) or :authority: (HTTP 2.0) headers sent by the browser? What is the User-Agent: HTTP header that is sent?

  Host: google.com
  User-Agent: Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.71 Mobile Safari/537.36

- What is the HTTP status code in the response and what does it mean?

  Status Code: 301 Moved Permanently

  Used for permanent redirecting

- Look up the status code. Show the associated HTTP response header that is sent in conjunction with this status code for the request.

  The response is not shown because it was redirected.

- What is the URL being requested? Is it using HTTP or HTTPS?

  Request URL: http://www.google.com/

  Http

- What is the HTTP status code in the response and what does it mean? Is it different from the first status code? If so, what is the semantic difference?

  Status Code: 302 Found
  This means the redirection address was found

- Show the associated HTTP response header that is sent in conjunction with this status code for the request.

  Location: https://www.google.com/?gws_rd=ssl

- What is the URL being requested? Is it using HTTP or HTTPS?

  Request URL: https://www.google.com/?gws_rd=ssl

  https

- What is the HTTP status code in the response?

  The HTTP 200 OK success status response code indicates that **the request has succeeded**

- Look for an alt-svc: HTTP response header. Does the server believe the client can use HTTP3/QUIC?

  Yes

- Examine the HTTP response headers for cookies. Show the cookies that are set and which ones specify that no SameSite restrictions are in place. What does the setting indicate about the cookies that are set?

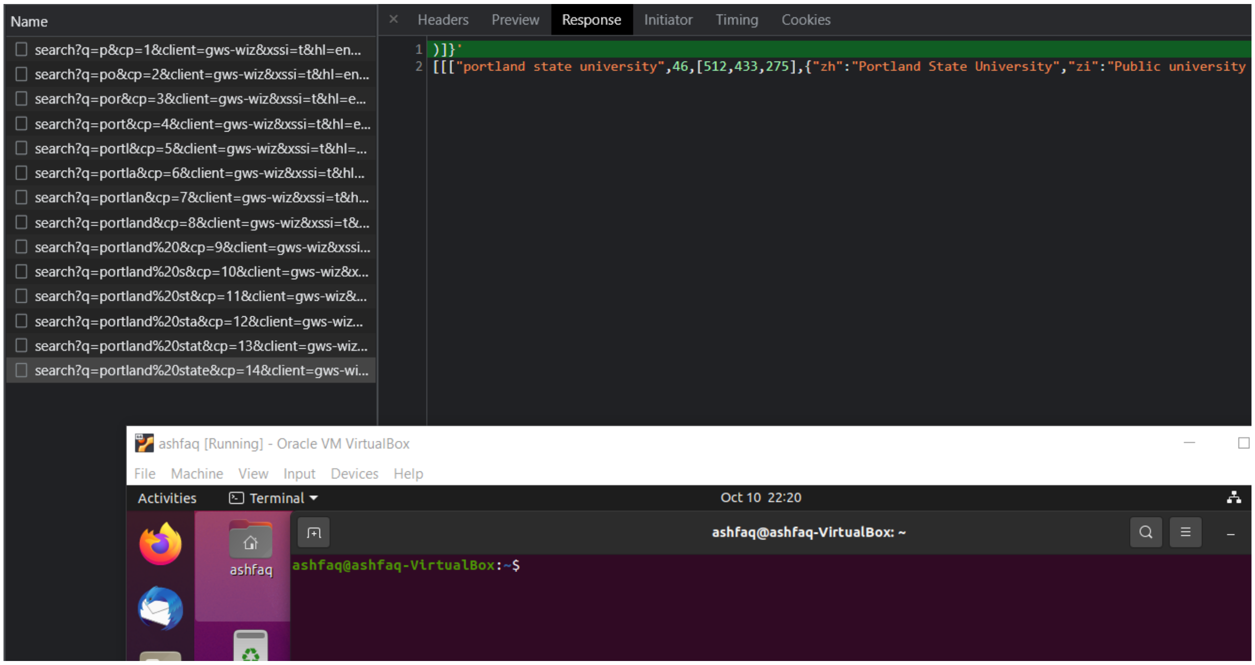
  set-cookie: 1P_JAR=2021-10-11-04; expires=Wed, 10-Nov-2021 04:08:41 GMT; path=/; domain=.google.com; Secure; SameSite=none

  set-cookie: NID=511=bJKkB8AlzVF9HRC3Tr_IE_Fy0-5Wfq2stBKrDUrUEFdaUpA9bp542UoDQAtV4HPnXHsUdjKm8CAETUhwqYcz2ejTcllkXqsA0I1BqoZ8gM8XBKZ6DANYYhjSY5ulWhQIO8-TF05-zlQjXe6iuc76jn-9Ea yWROkbD_rskOn_9so; expires=Tue, 12-Apr-2022 04:08:41 GMT; path=/; domain=.google.com; Secure; HttpOnly; SameSite=none

The cookies can only be accessed over HTTPS connections.

6. Asynchronous HTTP requests

- Show the requests and responses in the listing. Click on the last request sent, then click on the response to see that its payload has returned the data that is then rendered on the search page similar to what is shown below for "rabbid"



```
search?q=p&cp=1&client=gws-wiz&xssi=t&hl=en...
search?q=po&cp=2&client=gws-wiz&xssi=t&hl=en...
search?q=por&cp=3&client=gws-wiz&xssi=t&hl=e...
search?q=port&cp=4&client=gws-wiz&xssi=t&hl=e...
search?q=portl&cp=5&client=gws-wiz&xssi=t&hl=...
search?q=portla&cp=6&client=gws-wiz&xssi=t&hl...
search?q=portlan&cp=7&client=gws-wiz&xssi=t&h...
search?q=portland&cp=8&client=gws-wiz&xssi=t&...
search?q=portland%20&cp=9&client=gws-wiz&xssi...
search?q=portland%20s&cp=10&client=gws-wiz&x...
search?q=portland%20st&cp=11&client=gws-wiz&...
search?q=portland%20sta&cp=12&client=gws-wiz...
search?q=portland%20stat&cp=13&client=gws-wiz...
search?q=portland%20state&cp=14&client=gws-wi...
```

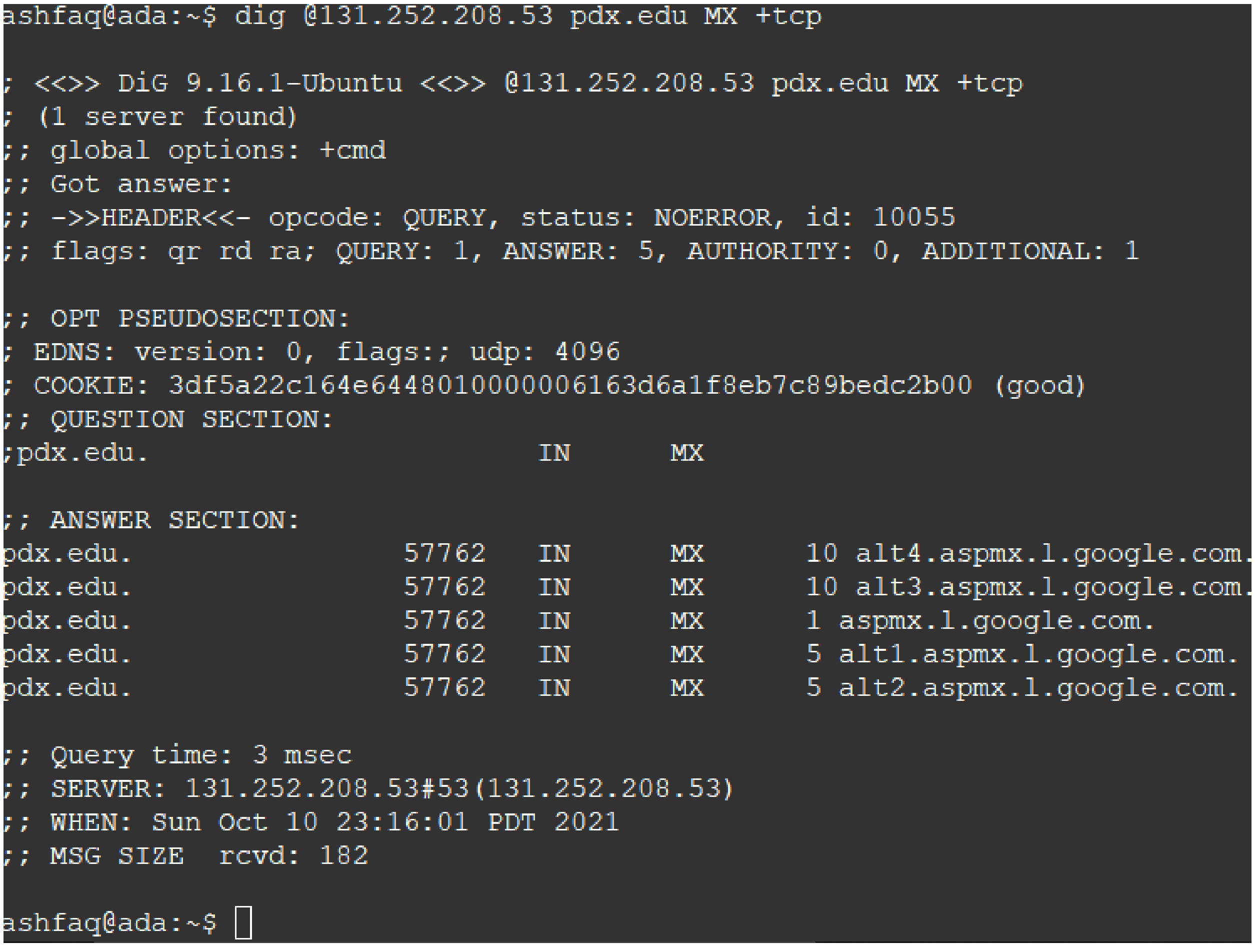
```
)]}'
[[["portland state university",46,[512,433,275],{"zh":"Portland State University","zi":"Public university
```

ashfaq@ashfaq-VirtualBox: ~

```
ashfaq@ashfaq-VirtualBox:~$
```

1. DNS #1 (dig)

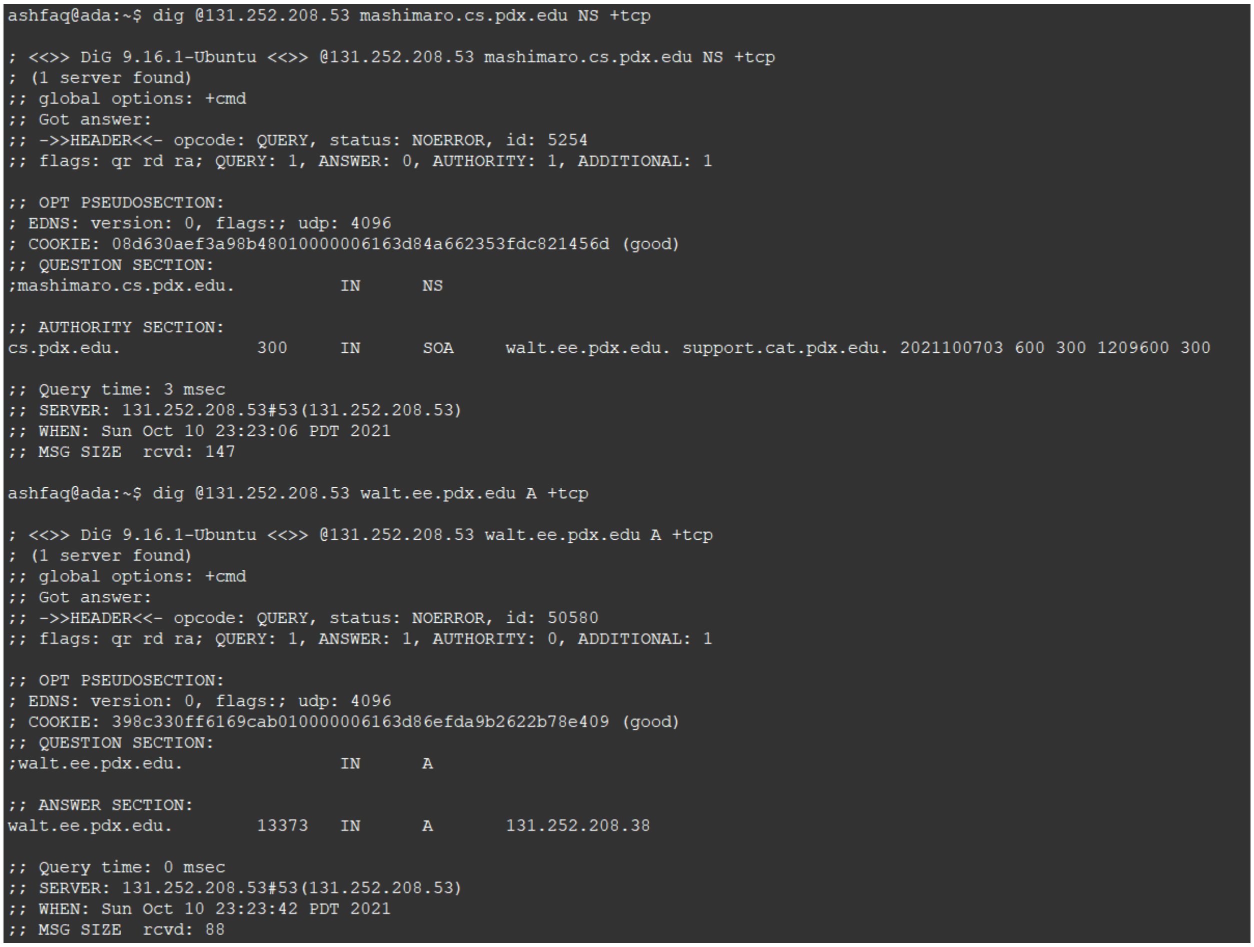
- Use dig to query the local DNS server for the A record of www.pdx.edu using TCP. Then, use dig to do the same for the MX record of pdx.edu. What do the ANSWER sections explain about where PSU's web/mail services are run from?



```
ashfaq@ada:~$ dig @131.252.208.53 pdx.edu MX +tcp

; <<>> DiG 9.16.1-Ubuntu <<>> @131.252.208.53 pdx.edu MX +tcp
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10055
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 3df5a22c164e6448010000006163d6a1f8eb7c89bedc2b00 (good)
;; QUESTION SECTION:
;pdx.edu.                       IN      MX

;; ANSWER SECTION:
pdx.edu.                57762   IN      MX      10 alt4.aspmx.l.google.com.
pdx.edu.                57762   IN      MX      10 alt3.aspmx.l.google.com.
pdx.edu.                57762   IN      MX      1 aspmx.l.google.com.
pdx.edu.                57762   IN      MX      5 alt1.aspmx.l.google.com.
pdx.edu.                57762   IN      MX      5 alt2.aspmx.l.google.com.

;; Query time: 3 msec
;; SERVER: 131.252.208.53#53(131.252.208.53)
;; WHEN: Sun Oct 10 23:16:01 PDT 2021
;; MSG SIZE  rcvd: 182

ashfaq@ada:~$ 
```

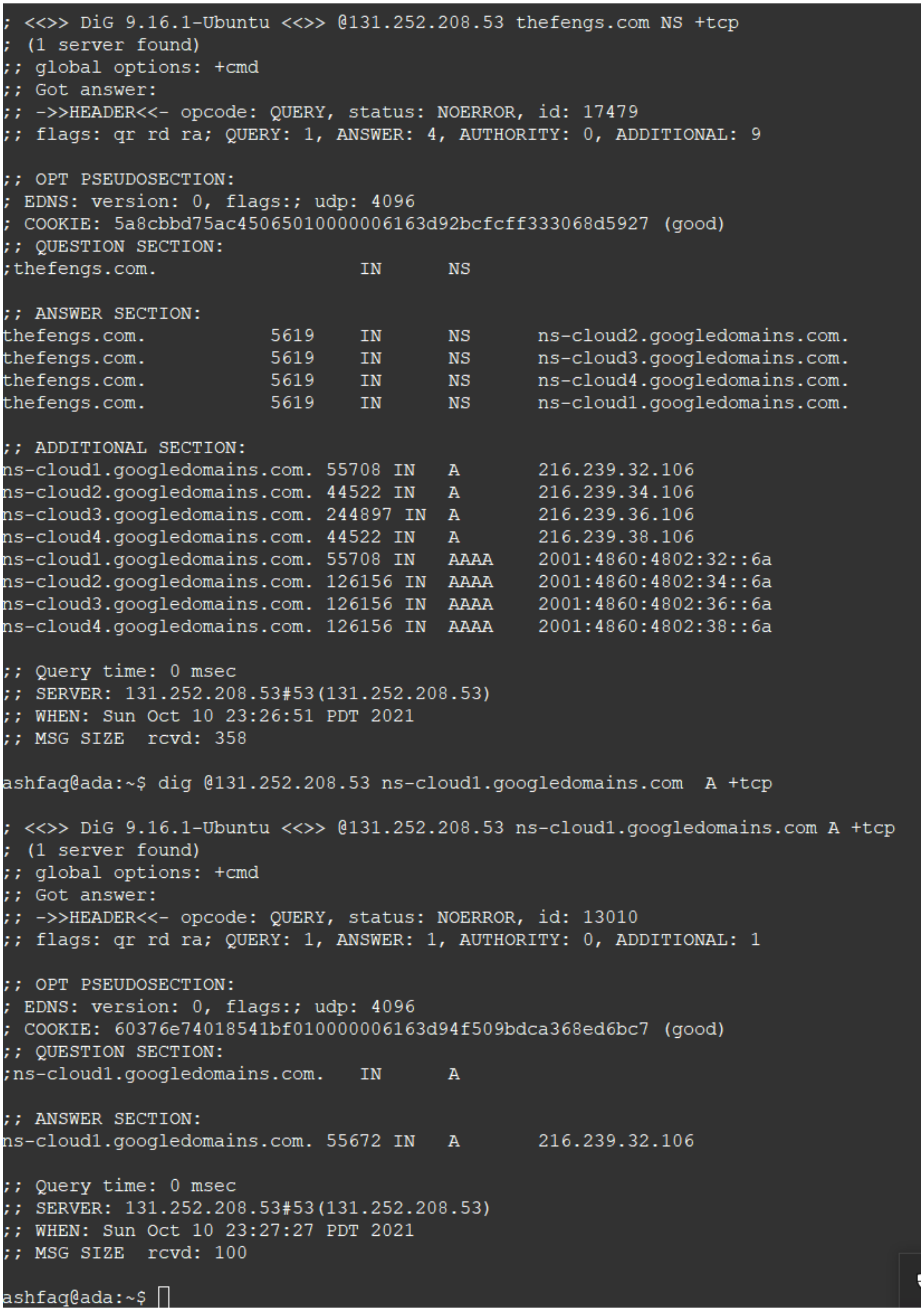
Running from Google

- Find the authoritative server (NS record type, AUTHORITY section response) for mashimaro.cs.pdx.edu and then query that server for the A record of mashimaro.cs.pdx.edu. Show both.



```
ashfaq@ada:~$ dig @131.252.208.53 mashimaro.cs.pdx.edu NS +tcp

; <<>> DiG 9.16.1-Ubuntu <<>> @131.252.208.53 mashimaro.cs.pdx.edu NS +tcp
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5254
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 08d630aef3a98b48010000006163d84a662353fdc821456d (good)
;; QUESTION SECTION:
;mashimaro.cs.pdx.edu.          IN      NS

;; AUTHORITY SECTION:
cs.pdx.edu.             300     IN      SOA     walt.ee.pdx.edu. support.cat.pdx.edu. 2021100703 600 300 1209600 300

;; Query time: 3 msec
;; SERVER: 131.252.208.53#53(131.252.208.53)
;; WHEN: Sun Oct 10 23:23:06 PDT 2021
;; MSG SIZE  rcvd: 147

ashfaq@ada:~$ dig @131.252.208.53 walt.ee.pdx.edu A +tcp

; <<>> DiG 9.16.1-Ubuntu <<>> @131.252.208.53 walt.ee.pdx.edu A +tcp
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 50580
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 398c330ff6169cab010000006163d86efda9b2622b78e409 (good)
;; QUESTION SECTION:
;walt.ee.pdx.edu.               IN      A

;; ANSWER SECTION:
walt.ee.pdx.edu.        13373   IN      A       131.252.208.38

;; Query time: 0 msec
;; SERVER: 131.252.208.53#53(131.252.208.53)
;; WHEN: Sun Oct 10 23:23:42 PDT 2021
;; MSG SIZE  rcvd: 88
```

- Find the authoritative server for thefengs.com and then query that server for the A record of thefengs.com

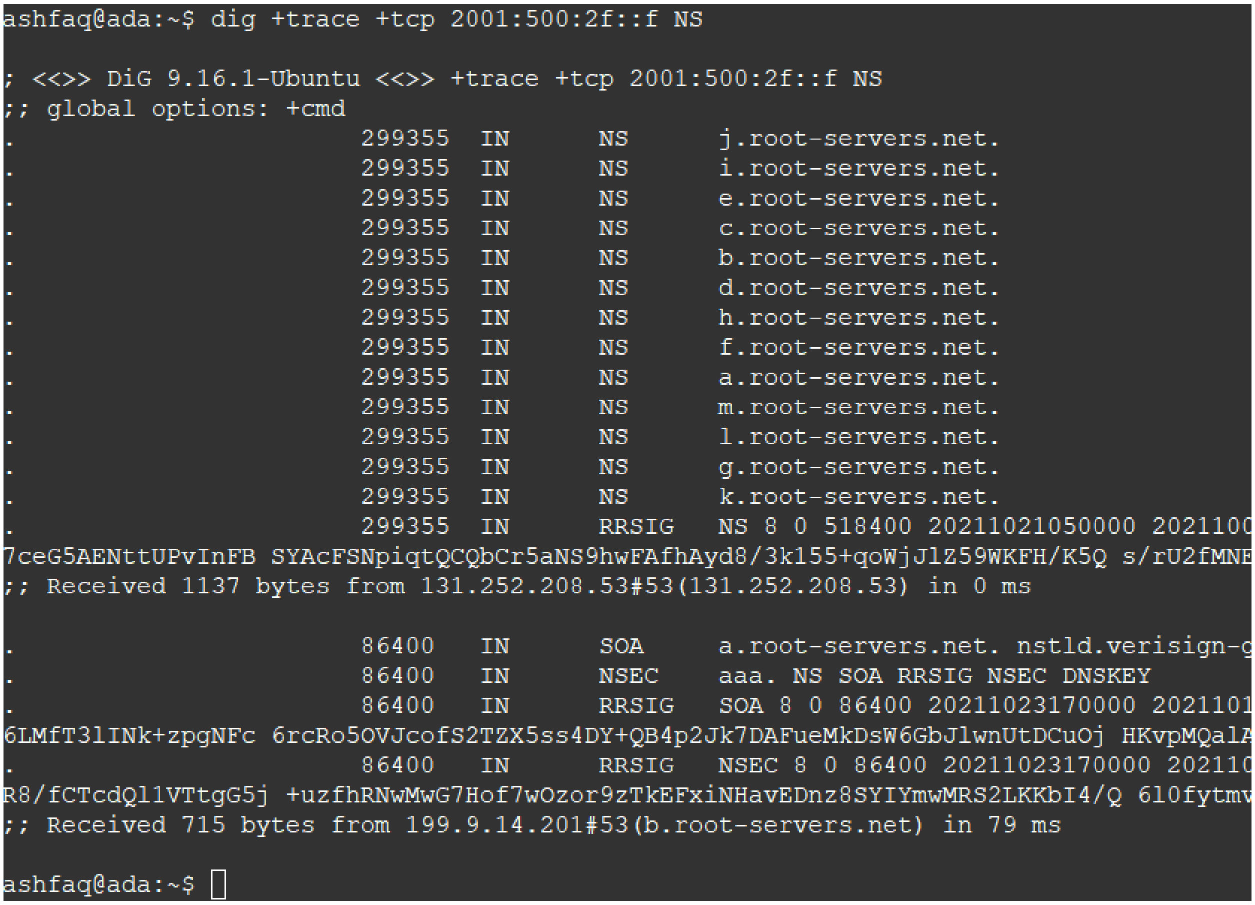


```
; <<>> DiG 9.16.1-Ubuntu <<>> @131.252.208.53 thefengs.com NS +tcp
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17479
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 5a8cbbd75ac4506501000000006163d92bcfcff333068d5927 (good)
;; QUESTION SECTION:
;thefengs.com.                  IN      NS

;; ANSWER SECTION:
thefengs.com.           5619    IN      NS      ns-cloud2.googledomains.com.
thefengs.com.           5619    IN      NS      ns-cloud3.googledomains.com.
thefengs.com.           5619    IN      NS      ns-cloud4.googledomains.com.
thefengs.com.           5619    IN      NS      ns-cloud1.googledomains.com.

;; ADDITIONAL SECTION:
ns-cloud1.googledomains.com. 55708 IN   A       216.239.32.106
ns-cloud2.googledomains.com. 44522 IN   A       216.239.34.106
ns-cloud3.googledomains.com. 244897 IN  A       216.239.36.106
ns-cloud4.googledomains.com. 44522 IN   A       216.239.38.106
ns-cloud1.googledomains.com. 55708 IN   AAAA    2001:4860:4802:32::6a
ns-cloud2.googledomains.com. 126156 IN  AAAA    2001:4860:4802:34::6a
ns-cloud3.googledomains.com. 126156 IN  AAAA    2001:4860:4802:36::6a
ns-cloud4.googledomains.com. 126156 IN  AAAA    2001:4860:4802:38::6a

;; Query time: 0 msec
;; SERVER: 131.252.208.53#53(131.252.208.53)
;; WHEN: Sun Oct 10 23:26:51 PDT 2021
;; MSG SIZE  rcvd: 358

ashfaq@ada:~$ dig @131.252.208.53 ns-cloud1.googledomains.com  A +tcp

; <<>> DiG 9.16.1-Ubuntu <<>> @131.252.208.53 ns-cloud1.googledomains.com A +tcp
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 13010
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 60376e74018541bf01000000006163d94f509bdca368ed6bc7 (good)
;; QUESTION SECTION:
;ns-cloud1.googledomains.com.   IN      A

;; ANSWER SECTION:
ns-cloud1.googledomains.com. 55672 IN   A       216.239.32.106

;; Query time: 0 msec
;; SERVER: 131.252.208.53#53(131.252.208.53)
;; WHEN: Sun Oct 10 23:27:27 PDT 2021
;; MSG SIZE  rcvd: 100

ashfaq@ada:~$ 
```

- When a web request hits port 80 of 131.252.220.66, how does the server know which site to serve from? (i.e. what protocol header)

  I believe that it goes to the authoritative server.

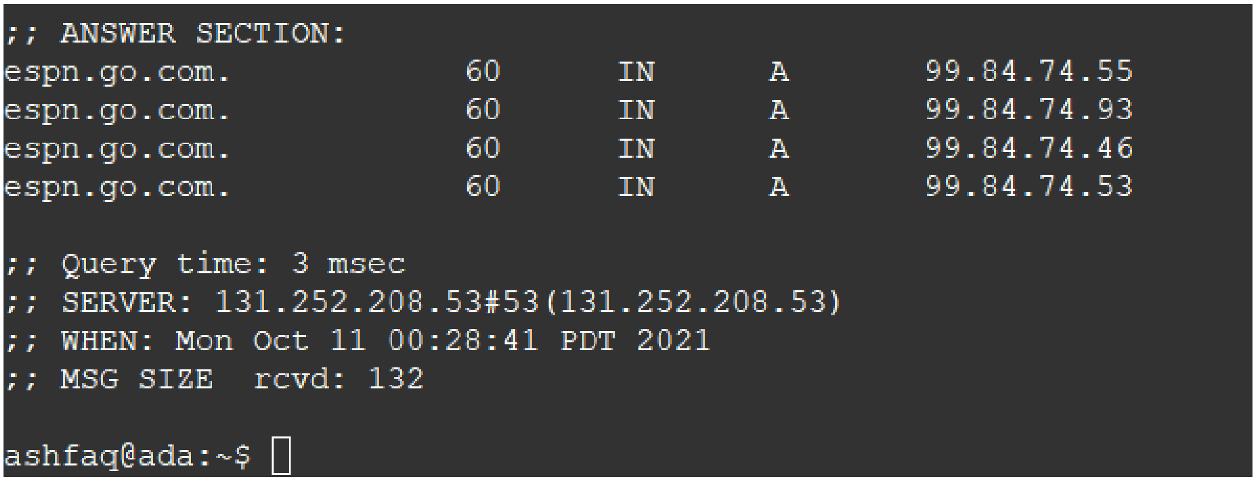
DNS iterative lookups



```
ashfaq@ada:~$ dig +trace +tcp 2001:500:2f::f NS

; <<>> DiG 9.16.1-Ubuntu <<>> +trace +tcp 2001:500:2f::f NS
;; global options: +cmd
.                       299355  IN      NS      j.root-servers.net.
.                       299355  IN      NS      i.root-servers.net.
.                       299355  IN      NS      e.root-servers.net.
.                       299355  IN      NS      c.root-servers.net.
.                       299355  IN      NS      b.root-servers.net.
.                       299355  IN      NS      d.root-servers.net.
.                       299355  IN      NS      h.root-servers.net.
.                       299355  IN      NS      f.root-servers.net.
.                       299355  IN      NS      a.root-servers.net.
.                       299355  IN      NS      m.root-servers.net.
.                       299355  IN      NS      l.root-servers.net.
.                       299355  IN      NS      g.root-servers.net.
.                       299355  IN      NS      k.root-servers.net.
.                       299355  IN      RRSIG   NS 8 0 518400 20211021050000 2021100
7ceG5AENttUPvInFB SYAcFSNpiqtQCQbCr5aNS9hwFAfhAyd8/3k155+qoWjJlZ59WKFH/K5Q s/rU2fMNE
;; Received 1137 bytes from 131.252.208.53#53(131.252.208.53) in 0 ms

.                       86400   IN      SOA     a.root-servers.net. nstld.verisign-g
.                       86400   IN      NSEC    aaa. NS SOA RRSIG NSEC DNSKEY
.                       86400   IN      RRSIG   SOA 8 0 86400 20211023170000 2021101
6LMfT3lINk+zpgNFc 6rcRo5OVJcofS2TZX5ss4DY+QB4p2Jk7DAFueMkDsW6GbJlwnUtDCuOj HKvpMQalA
.                       86400   IN      RRSIG   NSEC 8 0 86400 20211023170000 202110
R8/fCTcdQl1VTtgG5j +uzfhRNwMwG7Hof7wOzor9zTkEFxiNHavEDnz8SYIYmwMRS2LKKbI4/Q 6l0fytmv
;; Received 715 bytes from 199.9.14.201#53(b.root-servers.net) in 79 ms

ashfaq@ada:~$ 
```

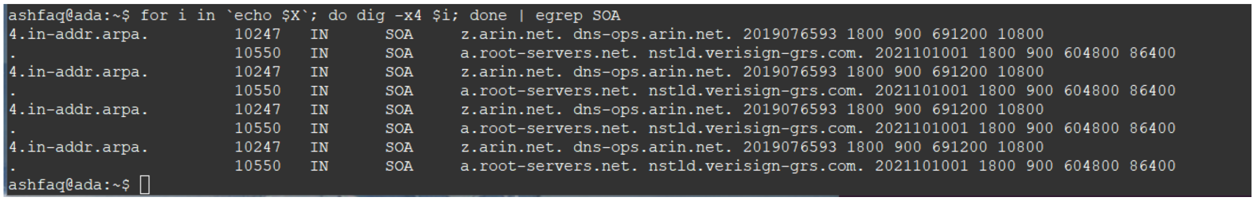
2. Reverse DNS lookups

- Use a single command line with commands dig, egrep, and awk, to list all IPv4 addresses that espn.go.com points to.



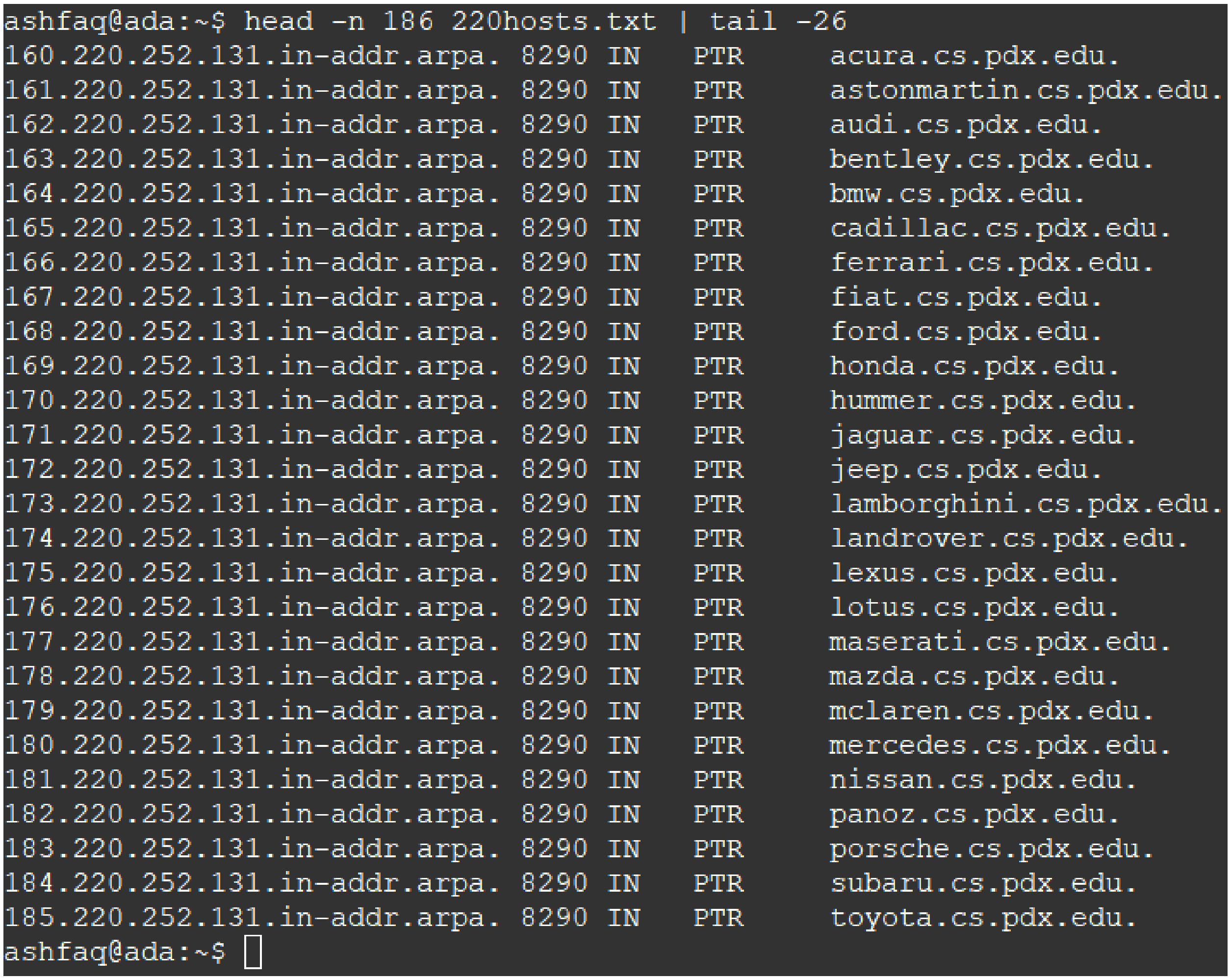
```
;; ANSWER SECTION:
espn.go.com.            60      IN      A       99.84.74.55
espn.go.com.            60      IN      A       99.84.74.93
espn.go.com.            60      IN      A       99.84.74.46
espn.go.com.            60      IN      A       99.84.74.53

;; Query time: 3 msec
;; SERVER: 131.252.208.53#53(131.252.208.53)
;; WHEN: Mon Oct 11 00:28:41 PDT 2021
;; MSG SIZE  rcvd: 132

ashfaq@ada:~$ 
```

- Take that list and create a single for loop in the shell that iterates over the list and performs a reverse lookup of each IP address to find each address's associated DNS name. As with the previous step, pipe the output of the for loop to egrep and awk so that the output consists only of the DNS names.



```
ashfaq@ada:~$ for i in `echo $X`; do dig -x4 $i; done | egrep SOA
4.in-addr.arpa.         10247   IN      SOA     z.arin.net. dns-ops.arin.net. 2019076593 1800 900 691200 10800
.                       10550   IN      SOA     a.root-servers.net. nstld.verisign-grs.com. 2021101001 1800 900 604800 86400
4.in-addr.arpa.         10247   IN      SOA     z.arin.net. dns-ops.arin.net. 2019076593 1800 900 691200 10800
.                       10550   IN      SOA     a.root-servers.net. nstld.verisign-grs.com. 2021101001 1800 900 604800 86400
4.in-addr.arpa.         10247   IN      SOA     z.arin.net. dns-ops.arin.net. 2019076593 1800 900 691200 10800
.                       10550   IN      SOA     a.root-servers.net. nstld.verisign-grs.com. 2021101001 1800 900 604800 86400
4.in-addr.arpa.         10247   IN      SOA     z.arin.net. dns-ops.arin.net. 2019076593 1800 900 691200 10800
.                       10550   IN      SOA     a.root-servers.net. nstld.verisign-grs.com. 2021101001 1800 900 604800 86400
ashfaq@ada:~$ 
```

3. Host enumeration



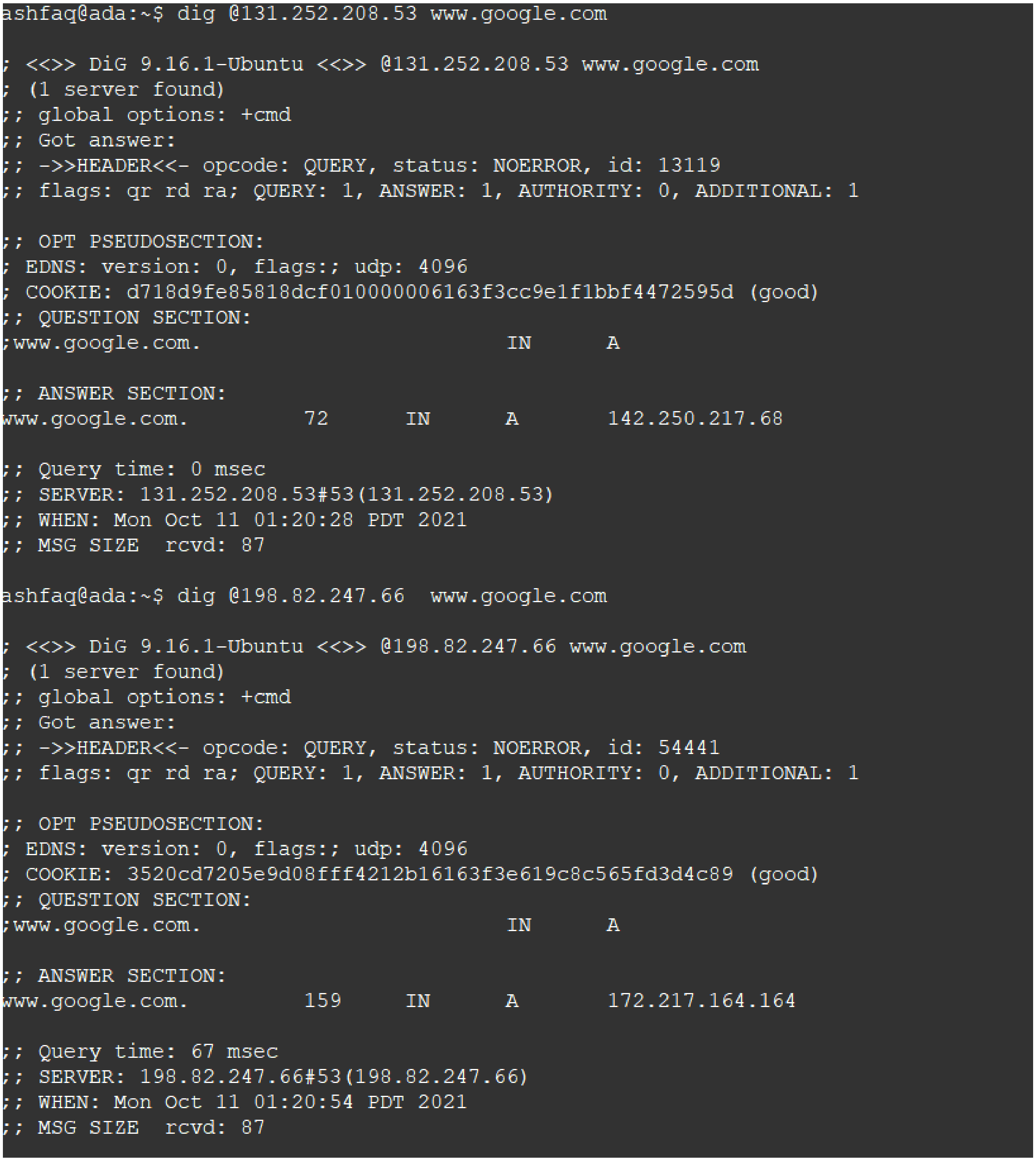
```
ashfaq@ada:~$ head -n 186 220hosts.txt | tail -26
160.220.252.131.in-addr.arpa. 8290 IN   PTR     acura.cs.pdx.edu.
161.220.252.131.in-addr.arpa. 8290 IN   PTR     astonmartin.cs.pdx.edu.
162.220.252.131.in-addr.arpa. 8290 IN   PTR     audi.cs.pdx.edu.
163.220.252.131.in-addr.arpa. 8290 IN   PTR     bentley.cs.pdx.edu.
164.220.252.131.in-addr.arpa. 8290 IN   PTR     bmw.cs.pdx.edu.
165.220.252.131.in-addr.arpa. 8290 IN   PTR     cadillac.cs.pdx.edu.
166.220.252.131.in-addr.arpa. 8290 IN   PTR     ferrari.cs.pdx.edu.
167.220.252.131.in-addr.arpa. 8290 IN   PTR     fiat.cs.pdx.edu.
168.220.252.131.in-addr.arpa. 8290 IN   PTR     ford.cs.pdx.edu.
169.220.252.131.in-addr.arpa. 8290 IN   PTR     honda.cs.pdx.edu.
170.220.252.131.in-addr.arpa. 8290 IN   PTR     hummer.cs.pdx.edu.
171.220.252.131.in-addr.arpa. 8290 IN   PTR     jaguar.cs.pdx.edu.
172.220.252.131.in-addr.arpa. 8290 IN   PTR     jeep.cs.pdx.edu.
173.220.252.131.in-addr.arpa. 8290 IN   PTR     lamborghini.cs.pdx.edu.
174.220.252.131.in-addr.arpa. 8290 IN   PTR     landrover.cs.pdx.edu.
175.220.252.131.in-addr.arpa. 8290 IN   PTR     lexus.cs.pdx.edu.
176.220.252.131.in-addr.arpa. 8290 IN   PTR     lotus.cs.pdx.edu.
177.220.252.131.in-addr.arpa. 8290 IN   PTR     maserati.cs.pdx.edu.
178.220.252.131.in-addr.arpa. 8290 IN   PTR     mazda.cs.pdx.edu.
179.220.252.131.in-addr.arpa. 8290 IN   PTR     mclaren.cs.pdx.edu.
180.220.252.131.in-addr.arpa. 8290 IN   PTR     mercedes.cs.pdx.edu.
181.220.252.131.in-addr.arpa. 8290 IN   PTR     nissan.cs.pdx.edu.
182.220.252.131.in-addr.arpa. 8290 IN   PTR     panoz.cs.pdx.edu.
183.220.252.131.in-addr.arpa. 8290 IN   PTR     porsche.cs.pdx.edu.
184.220.252.131.in-addr.arpa. 8290 IN   PTR     subaru.cs.pdx.edu.
185.220.252.131.in-addr.arpa. 8290 IN   PTR     toyota.cs.pdx.edu.
ashfaq@ada:~$
```

4. DNS #2 (Geographic DNS)

- What geographic locations do ipinfo.io and DB-IP return?

  Portland

- Record each result for your lab notebook.



```
ashfaq@ada:~$ dig @131.252.208.53 www.google.com

; <<>> DiG 9.16.1-Ubuntu <<>> @131.252.208.53 www.google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 13119
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: d718d9fe85818dcf010000006163f3cc9e1f1bbf4472595d (good)
;; QUESTION SECTION:
;www.google.com.                        IN      A

;; ANSWER SECTION:
www.google.com.         72      IN      A       142.250.217.68

;; Query time: 0 msec
;; SERVER: 131.252.208.53#53(131.252.208.53)
;; WHEN: Mon Oct 11 01:20:28 PDT 2021
;; MSG SIZE  rcvd: 87

ashfaq@ada:~$ dig @198.82.247.66  www.google.com

; <<>> DiG 9.16.1-Ubuntu <<>> @198.82.247.66 www.google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54441
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 3520cd7205e9d08fff4212b16163f3e619c8c565fd3d4c89 (good)
;; QUESTION SECTION:
;www.google.com.                        IN      A

;; ANSWER SECTION:
www.google.com.         159     IN      A       172.217.164.164

;; Query time: 67 msec
;; SERVER: 198.82.247.66#53(198.82.247.66)
;; WHEN: Mon Oct 11 01:20:54 PDT 2021
;; MSG SIZE  rcvd: 87
```

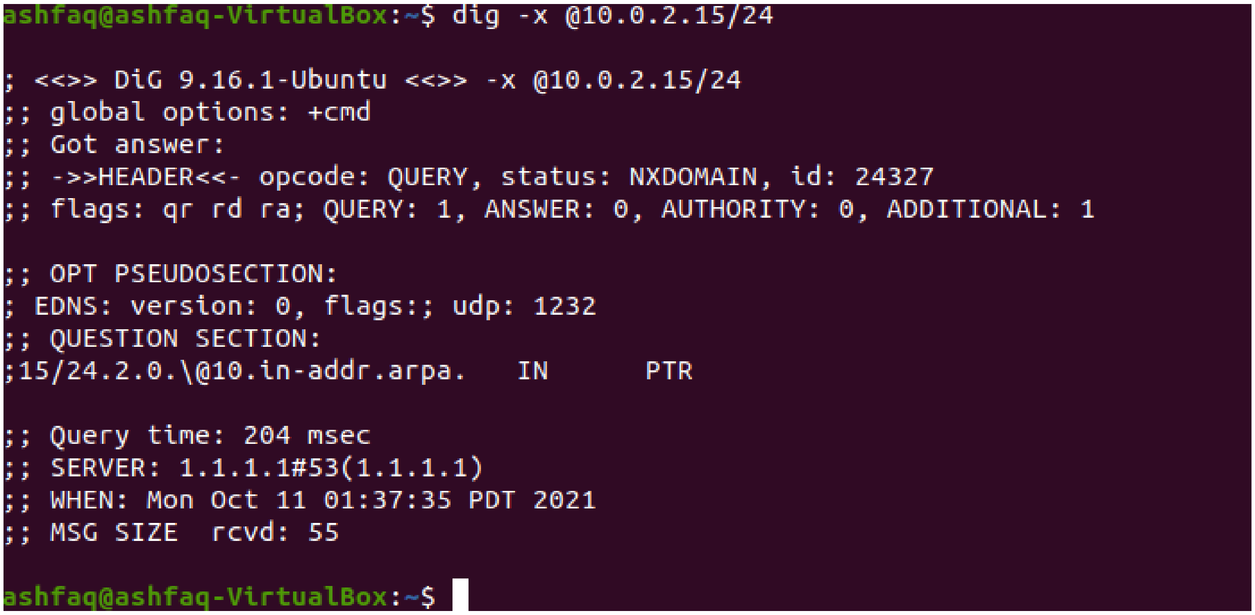
- What is the geographic distance between each pair of DNS servers and web servers?

  667 Miles for both IP addresses. California to Portland

- Do the routes reveal any information on the accuracy of the geographic locations given? (Answer might be no)

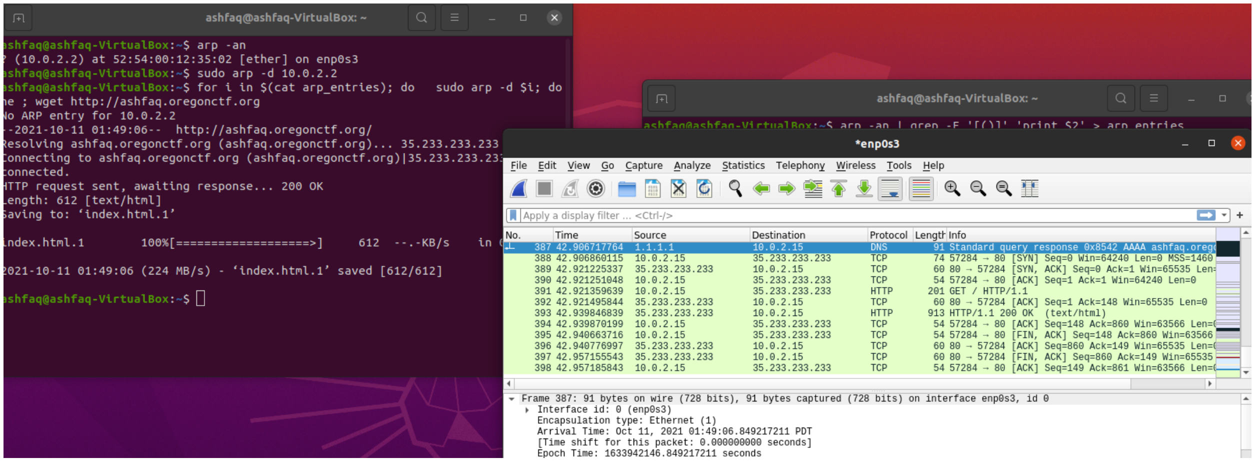
  Not really, one Ip too long to get to portland than it did to California. Without knowing where the locations were I would have assumed their locations differently.

5. Network Recap Lab #3



```
ashfaq@ashfaq-VirtualBox:~$ dig -x @10.0.2.15/24

; <<>> DiG 9.16.1-Ubuntu <<>> -x @10.0.2.15/24
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 24327
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;15/24.2.0.\@10.in-addr.arpa.    IN      PTR

;; Query time: 204 msec
;; SERVER: 1.1.1.1#53(1.1.1.1)
;; WHEN: Mon Oct 11 01:37:35 PDT 2021
;; MSG SIZE  rcvd: 55

ashfaq@ashfaq-VirtualBox:~$
```

6. Collect and analyze the network trace of a connection


ashfaq@ashfaq-VirtualBox: ~
ashfaq@ashfaq-VirtualBox: ~
*enp0s3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
ashfaq@ashfaq-VirtualBox:~$ arp -an
? (10.0.2.2) at 52:54:00:12:35:02 [ether] on enp0s3
ashfaq@ashfaq-VirtualBox:~$ sudo arp -d 10.0.2.2
ashfaq@ashfaq-VirtualBox:~$ for i in $(cat arp_entries); do   sudo arp -d $i; do
ne ; wget http://ashfaq.oregonctf.org
No ARP entry for 10.0.2.2
--2021-10-11 01:49:06--  http://ashfaq.oregonctf.org/
Resolving ashfaq.oregonctf.org (ashfaq.oregonctf.org)... 35.233.233.233
Connecting to ashfaq.oregonctf.org (ashfaq.oregonctf.org)|35.233.233.23
connected.
HTTP request sent, awaiting response... 200 OK
Length: 612 [text/html]
Saving to: ‘index.html.1’

index.html.1        100%[===================>]     612  --.-KB/s    in 

2021-10-11 01:49:06 (224 MB/s) - ‘index.html.1’ saved [612/612]

ashfaq@ashfaq-VirtualBox:~$ 
```

```
ashfaq@ashfaq-VirtualBox:~$ arp -an | grep -E '[()]' 'print $2' > arp_entries
```

No.	Time	Source	Destination	Protocol	Length	Info
387	42.906717764	1.1.1.1	10.0.2.15	DNS	91	Standard query response 0x8542 AAAA ashfaq.orego
388	42.906860115	10.0.2.15	35.233.233.233	TCP	74	57284 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
389	42.921225337	35.233.233.233	10.0.2.15	TCP	60	80 → 57284 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=
390	42.921251048	10.0.2.15	35.233.233.233	TCP	54	57284 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
391	42.921359639	10.0.2.15	35.233.233.233	HTTP	201	GET / HTTP/1.1
392	42.921495844	35.233.233.233	10.0.2.15	TCP	60	80 → 57284 [ACK] Seq=1 Ack=148 Win=65535 Len=0
393	42.939846839	35.233.233.233	10.0.2.15	HTTP	913	HTTP/1.1 200 OK (text/html)
394	42.939870199	10.0.2.15	35.233.233.233	TCP	54	57284 → 80 [ACK] Seq=148 Ack=860 Win=63566 Len=
395	42.940663716	10.0.2.15	35.233.233.233	TCP	54	57284 → 80 [FIN, ACK] Seq=148 Ack=860 Win=63566
396	42.940776997	35.233.233.233	10.0.2.15	TCP	60	80 → 57284 [ACK] Seq=860 Ack=149 Win=65535 Len=
397	42.957155543	35.233.233.233	10.0.2.15	TCP	60	80 → 57284 [FIN, ACK] Seq=860 Ack=149 Win=65535
398	42.957185843	10.0.2.15	35.233.233.233	TCP	54	57284 → 80 [ACK] Seq=149 Ack=861 Win=63566 Len=

```
▾ Frame 387: 91 bytes on wire (728 bits), 91 bytes captured (728 bits) on interface enp0s3, id 0
  ▸ Interface id: 0 (enp0s3)
    Encapsulation type: Ethernet (1)
    Arrival Time: Oct 11, 2021 01:49:06.849217211 PDT
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1633942146.849217211 seconds
```

- How many DNS requests are made?

  1 DNS request is made to the address to ask for the page.

- How many TCP connections does the browser initiate simultaneously to the site?

  3 TCP requests are made.

- How many HTTP GET requests are there for embedded objects?

  1 GET request is made and it receives an HTTP 1.1 200 OK