

## 2.1 TCP, HTTP

### 2.1.1 TCP #1: Sockets

ss, netstat

### 2.1.2 lsof

### 2.1.3 TCP #2: Throughput

### 2.1.4 iperf

### 2.1.5 HTTP #3: Requests

### 2.1.6 Asynchronous HTTP requests

## 2.2 DNS, Recap

### 2.2.1 DNS reconnaissance #1 (dig)

pdx.edu

mashimaro.cs.pdx.edu

### 2.2.2 Iterative DNS lookups

### 2.2.3 Reverse DNS lookups

Aliases and reverse lookups

### 2.2.4 Host enumeration

### 2.2.5 Geographic DNS #2

### 2.2.6 Wireshark Lab #3

### 2.2.7 -

### 2.2.8 Network Recap Lab #4

### 2.2.9 Collect trace

### 2.2.10 Analyze trace

ARP

DNS

TCP

HTTP

### 2.2.11 Clean up



## 2.1 TCP, HTTP

### 2.1.1 TCP #1: Sockets

ss, netstat

- Take a screenshot of the prompt and the command that ChatGPT generates

find a single command and its command-line flags that, when executed, lists all TCP sockets in a LISTEN state on an IPv4 address, showing the program that is using it.

The command to list all TCP sockets in a LISTEN state on an IPv4 address, along with the program that is using each socket, is:

```
bash
```

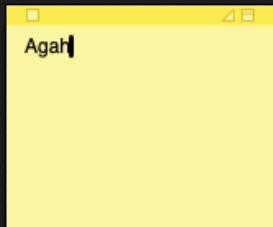
```
sudo netstat -tlnp --inet
```

Copy code

**Explanation of the flags:**

- `-t` : Shows only TCP sockets.
- `-l` : Lists only sockets that are in the LISTEN state.
- `-n` : Displays addresses and port numbers in numeric form.
- `-p` : Shows the PID and name of the program using the socket.
- `--inet` : Filters for IPv4 addresses only.

This command requires `sudo` to display information about the programs associated with each socket.



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

```
ahmadagah@course-vm: ~  
ahmadagah@course-vm:~$ sudo netstat -tlnp --inet  
Active Internet connections (only servers)  
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name  
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      390/systemd-resolve  
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      839/sshd: /usr/sbin  
tcp        0      0 127.0.0.1:42983          0.0.0.0:*               LISTEN      471/containerd  
ahmadagah@course-vm:~$
```

- List a service that can be contacted from any interface on the machine. List a service that can only be contacted by local processes.

The sshd service (Secure Shell Daemon) is listening on `0.0.0.0:22`. The IP `0.0.0.0` indicates that it is bound to all available network interfaces, meaning it can be contacted from any network interface on the machine, both local and external.

The systemd-resolve service is listening on `127.0.0.53:53`. The IP `127.0.0.53` is within the `127.0.0.0/8` loopback range, which means it is only accessible from local processes on the same machine and cannot be accessed externally.

Similarly, the containerd service is listening on `127.0.0.1:42983`, another loopback address, which also restricts it to local processes only.

- Run the command again, but do not use `sudo` as this is a machine managed by CAT. Include a screenshot of the output.

```
agah@ada:~$ netstat -tlnp --inet
(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      0 127.0.0.54:53          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:35055        0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:34463        0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6112         0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6104         0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6105         0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6107         0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6109         0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6100         0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6101         0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6102         0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6103         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      -
agah@ada:~$
```

- List the services that this machine provides for external access

none of the listed services are available for external access, all services shown are bound to **127.0.0.1** (the loopback address), which means they are only accessible from the local machine.

## 2.1.2 lsof

- Take a screenshot of the prompt and the command that ChatGPT generates

find a single lsof command and its command-line flags that, when executed, lists all TCP sockets in a LISTEN state on an IPv4 address, showing the program that is using it, using lsof and grep to list only ipv4 including the header row to show column names , something like sudo lsof. .... | grep -E 'COMMAND | IPv4'

To list all TCP sockets in a LISTEN state on an IPv4 address with column headers included, you can use the following `lsof` command combined with `grep` :

```
bash
```

```
Copy code
```

```
sudo lsof -iTCP -sTCP:LISTEN -Pn | grep -E 'COMMAND | IPv4'
```


### Explanation:

- `sudo` : Runs the command with superuser privileges to access all processes.
- `lsof` : Lists open files, including network connections.
- `-iTCP` : Filters for TCP sockets.
- `-sTCP:LISTEN` : Shows only TCP sockets that are in the LISTEN state.
- `-P` : Displays port numbers without resolving them to service names.
- `-n` : Disables DNS resolution for IP addresses, showing raw IP addresses and making the command faster.
- `| grep -E 'COMMAND | IPv4'` : Uses `grep` to filter the output, including only lines containing `COMMAND` (the header row) or `IPv4` .
  - `COMMAND` is part of the header line, which means the header will be included in the output.
  - `IPv4` will filter for only IPv4 entries.

### What This Command Does

- The command first lists all listening TCP sockets with `lsof` and includes port numbers and IP addresses in numeric format.
- The `grep` command filters the output to display only lines containing the headers or IPv4 entries, resulting in a clean output with ↓ jmn names and only IPv4 listening sockets.

Agah

 Get citation

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

```

ahmadagah@course-vm: ~
linux.cs.pdx.edu (ssh)  %1  ahmadagah@course-vm: ~ (ssh)
ahmadagah@course-vm:~$ sudo lsof -iTCP -sTCP:LISTEN -Pn | grep -E 'COMMAND|IPv4'
COMMAND  PID      USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
systemd-r 390    systemd-resolve 14u  IPv4  4188    0t0  TCP 127.0.0.53:53 (LISTEN)
container 471      root    10u  IPv4  5756    0t0  TCP 127.0.0.1:42983 (LISTEN)
sshd      839      root     3u   IPv4  5659    0t0  TCP *:22 (LISTEN)
ahmadagah@course-vm:~$

```

notebook.

## 2.1.3 TCP #2: Throughput

### 2.1.4 iperf

- 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.

```

agah@west1-b:~$ sudo iperf -c 10.152.0.2 -p 80
-----
Client connecting to 10.152.0.2, TCP port 80
TCP window size: 16.0 KByte (default)
-----
[ 1] local 10.138.0.7 port 52162 connected with 10.152.0.2 port 80 (icwnd/mss/irrt=13/1408/139085)
[ ID] Interval      Transfer    Bandwidth
[ 1] 0.0000-10.2317 sec  183 MBytes  150 Mbits/sec
agah@west1-b:~$ sudo iperf -c 10.142.0.3 -p 80
-----
Client connecting to 10.142.0.3, TCP port 80
TCP window size: 16.0 KByte (default)
-----
[ 1] local 10.138.0.7 port 38528 connected with 10.142.0.3 port 80 (icwnd/mss/irrt=13/1408/66240)
[ ID] Interval      Transfer    Bandwidth
[ 1] 0.0000-10.1041 sec   412 MBytes  342 Mbits/sec
agah@west1-b:~$ sudo iperf -c 10.132.0.2 -p 80
-----
Client connecting to 10.132.0.2, TCP port 80
TCP window size: 16.0 KByte (default)
-----
[ 1] local 10.138.0.7 port 59910 connected with 10.132.0.2 port 80 (icwnd/mss/irrt=13/1408/135867)
[ ID] Interval      Transfer    Bandwidth
[ 1] 0.0000-10.2021 sec   195 MBytes  160 Mbits/sec
agah@west1-b:~$

```

VM at **10.142.0.3** has the highest bandwidth, possibly indicating that it is either on the same network segment or has fewer competing network resources compared to the others, VMs at **10.152.0.2** and **10.132.0.2** have lower bandwidths, potentially due to being located farther away, experiencing more network congestion, or having less optimal network paths.

## 2.1.5 HTTP #3: Requests

- Take a screenshot of the initial requests for your lab notebook.

The screenshot shows the Chrome DevTools Network tab with the 'All' filter selected. The network log displays a series of requests for google.com. The table below summarizes the visible requests.

Name	Status	Type	Initiator	Size	Time
google.com.	301	document / Redir...	Other	0 B	Pending
google.com.	301	document / Redir...	google.com /	154 B	Pending
www.google.com	200	document	google.com /	58.4 kB	110 ms
google.com	200	document / Redir...	google.com /	425 B	Pending
4UaGrENHsxJIGDuGo1OILL3Owp5eKQtG.woff2	200	font	(index):0	14.7 kB	50 ms
4UabrENHsxJIGDuGo1OILLU94YtzCwZsPF4o.woff2	200	font	(index):0	15.2 kB	40 ms
googlelogo_color_160x56dp.png	200	png	(index):63	7.4 kB	19 ms
searchbox_sprites317_hr.webp	200	webp	(index):63	1.8 kB	566 ms
rs=AA2YrTuWzR02q-lADuEe1tZdVix-MATZjg	200	script	(index):108	80.0 kB	61 ms
rs=AA2YrTutjgcOSyX6KH6kYJp5HabmF8SGA	200	stylesheet	(index):108	2.8 kB	41 ms
m=bct,cdos,hsm,jsa,qim,rr4R9e,d,csi,cEt90b,SNUn3,qddgKe,...	200	stylesheet	(index):116	427 B	9 ms
gen_204?s=webhp&t=aft&atyp=csi&ei=yuUKZ93cKtWCm9cP...	204	ping	(index):12	219 B	52 ms
m=bct,cdos,hsm,jsa,qim,rr4R9e,d,csi,cEt90b,SNUn3,qddgKe,...	200	script	(index):120	336 kB	47 ms
nav_logo325_hr.webp	200	webp	(index):116	5.2 kB	46 ms
cb=gapi.loaded_0	200	script	rs=AA2YrTuWzR02q-lADuEe1tZdV	43.2 kB	46 ms
gen_204?atyp=csi&ei=yuUKZ93cKtWCm9cPi930gQY&s=web...	204	ping	m=bct,cdos,hsm,jsa,qim,rr4R9e,d	209 B	18 ms
search?q&cp=0&client=mobile-gws-wiz-hp&xssi=t&gs_p...0&...	200	xhr	m=bct,cdos,hsm,jsa,qim,rr4R9e,d	1.8 kB	97 ms
m=sb_wiz,aa,abd,syv3,syv2,sygb,syv1,syup,syyl,syxn...sy9,s...	200	script	(index):120	176 kB	30 ms
rs=ACT90oF9JYICDED6BY7VQ765jJEDUNVCCw	200	fetch	m=bct,cdos,hsm,jsa,qim,rr4R9e,d	1.5 kB	28 ms
client_204?atyp=i&biw=1676&bih=126&dpr=2&ei=yuUKZ93c...	204	text/html	(index):3	223 B	27 ms
24px.svg	200	svg+xml	(index):123	973 B	51 ms
m=WINQGD,kQvlef,syyj,lIlQlf,syyv,syyp,nabPbb?xjs=s4	200	script	m=bct,cdos,hsm,jsa,qim,rr4R9e,d	3.9 kB	10 ms
gen_204?atyp=csi&ei=yuUKZ93cKtWCm9cPi930gQY&s=pro...	204	ping	m=bct,cdos,hsm,jsa,qim,rr4R9e,d	210 B	18 ms
search?q&pg&cp=0&client=mweb-insp&xssi=t&dpr=2	200	xhr	m=bct,cdos,hsm,jsa,qim,rr4R9e,d	1.6 kB	73 ms
hpba?vet=10ahUKEwjD6eDG4ImJAxVvweYEHYsuPWAQj-0KC...	200	xhr	m=sb_wiz,aa,abd,syv3,syv2,sygb,	258 B	43 ms
gen_204?atyp=i&ei=yuUKZ93cKtWCm9cPi930gQY&vet=10ah...	204	ping	m=bct,cdos,hsm,jsa,qim,rr4R9e,d	210 B	20 ms
m=syg9,aLUfP?xjs=s4	200	script	m=bct,cdos,hsm,jsa,qim,rr4R9e,d	688 B	8 ms

45 requests | 764 kB transferred | 2.2 MB resources | Finish: 4.31 s | DOMContentLoaded: 151 ms | Load: 694 ms

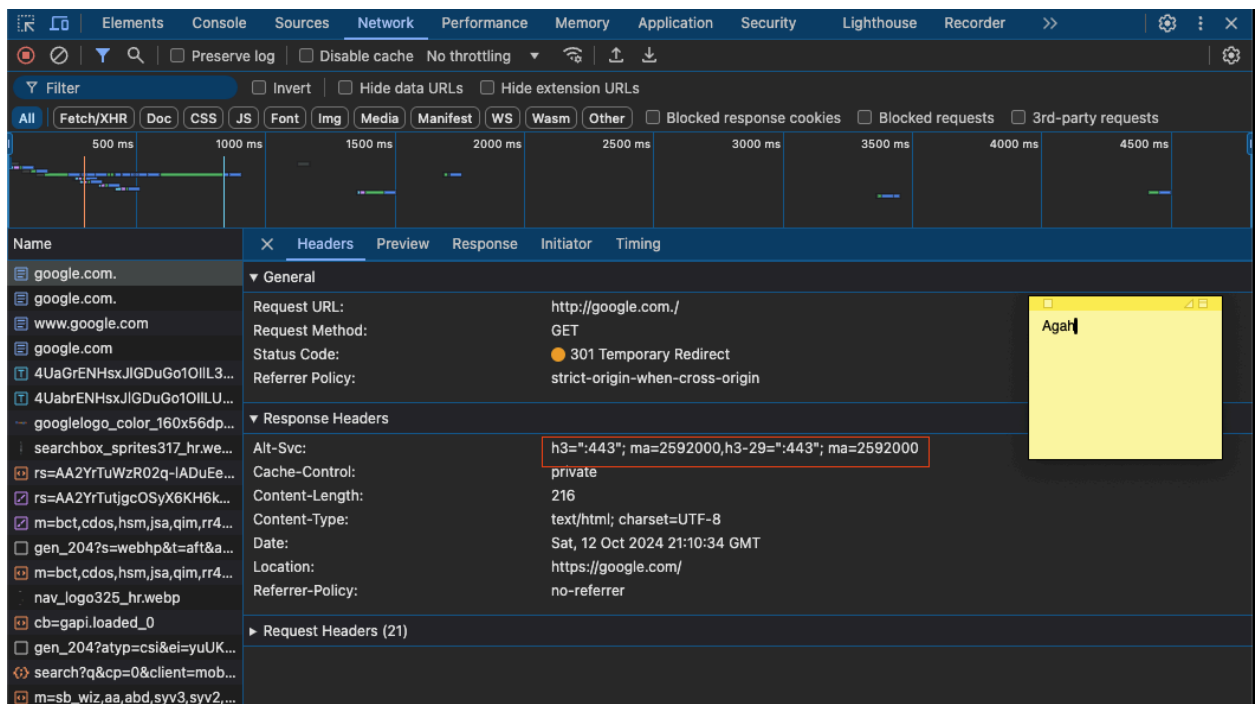
- What is the URL being requested?

`http://google.com/`

- Explain the HTTP status code that is returned and what the code indicates

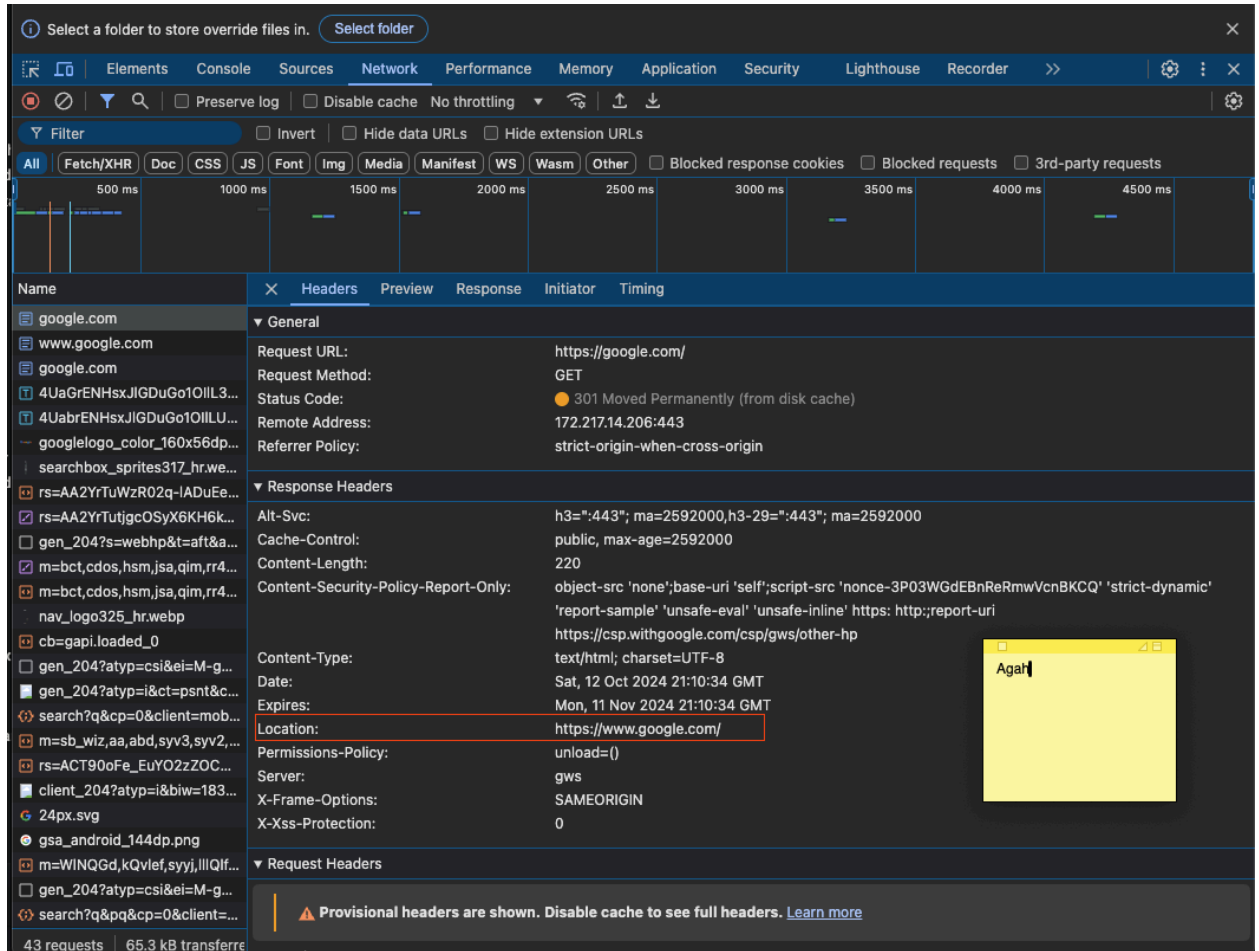
301 Moved Permanently: This status code indicates that the requested resource has been permanently moved to a different URL. For example, <http://google.com> redirects to <https://www.google.com>. The 301 status code instructs the browser to make future requests to the new URL.

- Take a screenshot indicating the version of the HTTP protocol that is used for each request. (Hint: look at the response status line and `alt-svc` HTTP response headers indicating HTTP/2 or HTTP/3).



- Show the URLs the browser is redirected to via this header.





- Take a screenshot of when cookies are set via **Set-Cookie:**

The screenshot shows the Chrome DevTools Network tab with a list of requests to google.com. The 'Headers' tab is selected for the first request, displaying various headers including Accept-Ch, Alt-Svc, Cache-Control, Content-Encoding, Content-Length, Content-Security-Policy-Report-Only, Content-Type, Date, Expires, P3p, Permissions-Policy, Server, Set-Cookie, and another Set-Cookie. A yellow sticky note with the text 'Agar' is placed over the Set-Cookie header.

Name	Headers	Preview	Response	Initiator	Timing	Cookies
google.com	Accept-Ch:		Sec-CH-UA-Platform-Version			
www.google.com	Accept-Ch:		Sec-CH-UA-Full-Version			
google.com	Accept-Ch:		Sec-CH-UA-Arch			
googlelogo_color_160x56dp.png	Accept-Ch:		Sec-CH-UA-Model			
m=bct,cdos,hsm,jsa,qim,rr4R9e,d,csi,cEt90b,SNUn3,...	Accept-Ch:		Sec-CH-UA-Bitness			
4UaGrENHsxJIGDuGo1OILL3Owp5eKQtG.woff2	Accept-Ch:		Sec-CH-UA-Full-Version-List			
4UabrENHsxJIGDuGo1OILLU94YtzCwZsPF4o.woff2	Accept-Ch:		Sec-CH-UA-WoW64			
searchbox_sprites317_hr.webp	Alt-Svc:		h3=":443"; ma=2592000,h3-29=":443"; ma=2592000			
rs=AA2YrTuWzR02q-lADuEe1tZdVix-MATZjg	Cache-Control:		private			
rs=AA2YrTutjgcOSyX6KH6kYXJp5HabmF8SGA	Content-Encoding:		br			
gen_204?s=webhp&t=af&tatyp=csi&ei=3-oKZ6WxGf...	Content-Length:		58635			
nav_logo325_hr.webp	Content-Security-Policy-Report-Only:		object-src 'none';base-uri 'self';script-src 'nonce-spyBov8NVPNQb5DSCGwzMQ' 'strict-dynamic' 'report-sample' 'unsafe-eval' 'unsafe-inline' https://http://report-uri https://csp.withgoogle.com/csp/gws/other-hp			
m=bct,cdos,hsm,jsa,qim,rr4R9e,d,csi,cEt90b,SNUn3,...	Content-Type:		text/html; charset=UTF-8			
cb=gapi.loaded_0	Date:		Sat, 12 Oct 2024 21:32:15 GMT			
gen_204?atyp=csi&ei=3-oKZ6WxGf3E0PEPq7KwUA&...	Expires:		Sat, 12 Oct 2024 21:32:15 GMT			
search?q&cp=0&client=mobile-gws-wiz-hp&xssi=t&g...	P3p:		CP="This is not a P3P policy! See g.co/p3phelp for more info."			
m=sb_wiz,aa,abd,syv3,syv2,sygb,syv1,syup,syyl,syxn...	Permissions-Policy:		unload=()			
rs=ACT90oG7jjGnlhzKoJNF3BZPr_Wk2SjpbQ	Server:		gws			
client_204?atyp=i&biw=1830&bih=126&dpr=2&ei=3-...	Set-Cookie:		AEC=AVYB7cr12aPI1w6AOuDC12O27ubJbAXSZQDpNFv_-wcA6KSpxMGM-jiukg; expires=Thu, 10-Apr-2025 21:32:15 GMT; path=/; domain=.google.com; Secure; HttpOnly; SameSite=lax			
24px.svg	Set-Cookie:		NID=518=euNxuQXv0Gsgw7g4AHefWYcJO3Umk5XX0-NedYv01bCqpbhtlSiccsij01GihR2GQUu0Q_XG3xytbzsXXREV1YBuVN5lopOJO TayxCW1luazVwnyyXNf5UbD1RYxKmsl9bSj1L1Yr6NRfpM0YHNDmvx9THQs XsqadVpN0xpdP2fd7Ok7ZfWlVhouluTzdKJgb2; expires=Sun, 13-Apr-2025 21:32:15 GMT; path=/; domain=.google.com; Secure; HttpOnly; SameSite=none			
m=WINQGD,kQvlef,syyl,IlIqIf,syvv,syyp,nabPbb?xjs=s4						
gen_204?atyp=csi&ei=3-oKZ6WxGf3E0PEPq7KwUA&...						
search?q&pq&cp=0&client=mweb-insp&xssi=t&dpr=2						
hpba?vet=10ahUKEwjlnP6y5YmJAXv9IjQIHsZDAoQj-...						
gen_204?atyp=i&ei=3-oKZ6WxGf3E0PEPq7KwUA&ve...						
m=syg9,aLUFp?xjs=s4						
m=IOOVd,sy8p,P6sQOc?xjs=s4						

44 requests | 765 kB transferred | 2.2 MB resources

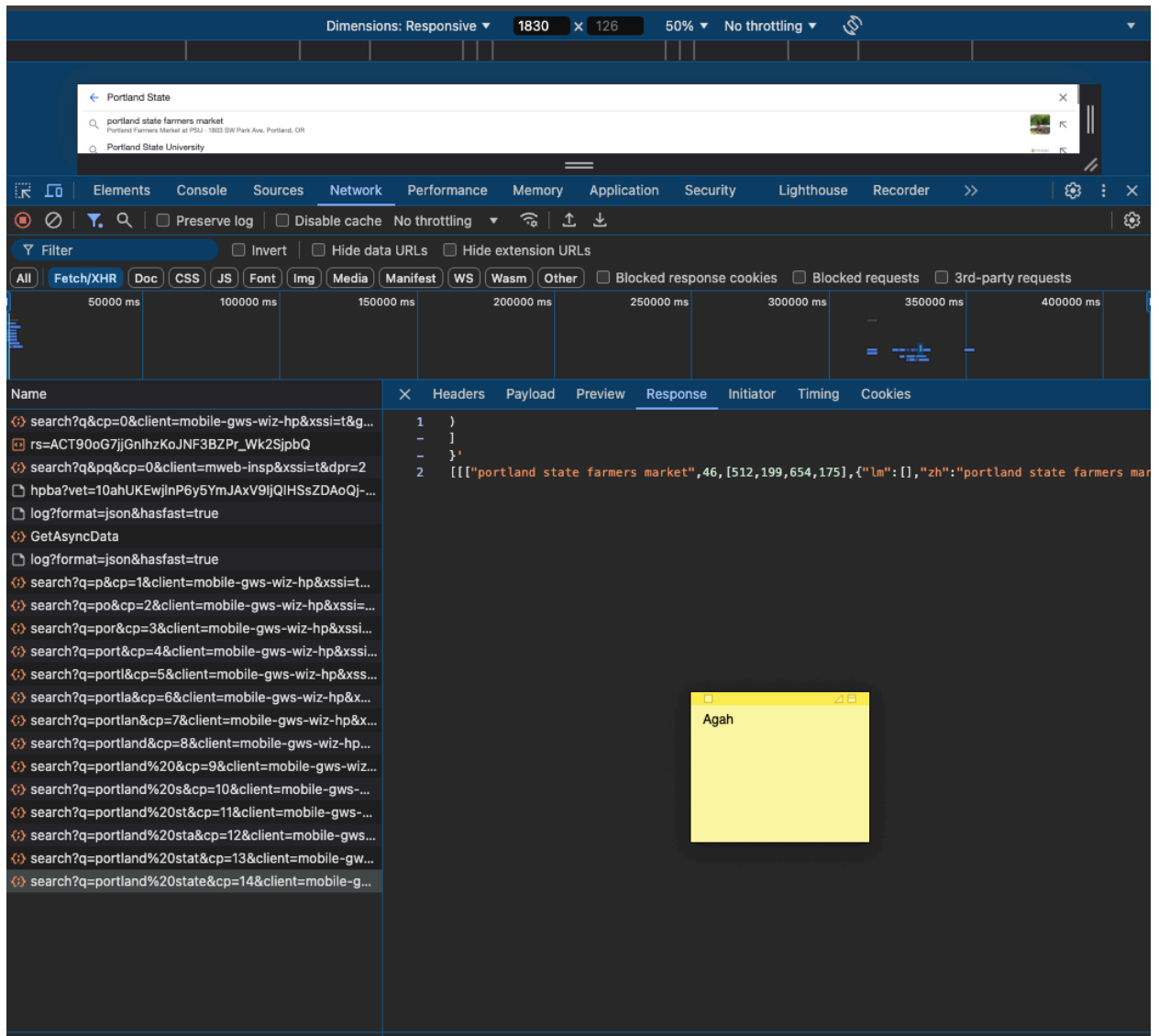
- Take a screenshot of when cookies are attached via **Cookie:**

The screenshot shows the Chrome DevTools Network tab. The top panel displays a timeline of network requests. The bottom panel shows the details of the selected request, which is a GET request to `google.com`. The details panel is divided into sections for Response Headers, Request Headers, and Cookies. A yellow sticky note with the word "Agah" is placed over the top right of the details panel.

Name	Headers	Preview	Response	Initiator	Timing	Cookies
google.com	▶ Response Headers (14)					
www.google.com	▼ Request Headers					
google.com	:authority: www.google.com :method: GET :path: /images/branding/googlelogo/2x/googlelogo_color_160x56dp.png :scheme: https Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*;*/q=0.8 Accept-Encoding: gzip, deflate, br, zstd Accept-Language: en-US,en;q=0.9 Cookie: AEC=AVYB7cr12aPl1w6AOuDc12O27ubJbAXSZQDpNFv_-wcA6KSpxMGM-jiukg; NID=518=euNxfuQXv0Gsgw7g4AHefWYcJO3Umk5XX0-NedYv01bCqpbhtlSiccsij01GihR2GQUq0Q_XG3xytbzsXXREV1YBuVN5lopOjOTayx CW1luazVwnyyXNf5UbD1RYxKMSl9bSj1L1YJr6NRfpM0YHNDmvx9THQsXsqadVp N0xpdP2fd7Ok7ZfwlVhouluTzdKJgb2 Priority: i Referer: https://www.google.com/ Sec-Ch-Prefers-Color-Scheme: dark Sec-Ch-Ua: "Google Chrome",v="129", "Not=A?Brand";v="8", "Chromium";v="129" Sec-Ch-Ua-Arch: "" Sec-Ch-Ua-Bitness: "64" Sec-Ch-Ua-Form-Factors: Sec-Ch-Ua-Full-Version: "129.0.6668.100" Sec-Ch-Ua-Full-Version-List: "Google Chrome";v="129.0.6668.100", "Not=A?Brand";v="8.0.0.0", "Chromium";v="129.0.6668.100" Sec-Ch-Ua-Mobile: ?1 Sec-Ch-Ua-Model: "Nexus 5" Sec-Ch-Ua-Platform: "Android" Sec-Ch-Ua-Platform-Version: "6.0" Sec-Ch-Ua-Wow64: ?0					

## 2.1.6 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**"



## 2.2 DNS, Recap



### 2.2.1 DNS reconnaissance #1 (dig)

pdx.edu

- Take a screenshot of the prompt and the dig command produced.

Agah

produce a dig command that queries PSU's local DNS server at 131.252.208.53 for the A record of www.pdx.edu using TCP.



To query PSU's local DNS server at 131.252.208.53 for the A record of www.pdx.edu using TCP with the dig command, you can use the following syntax:

bash

Copy code

```
dig +tcp @131.252.208.53 www.pdx.edu A
```

### Explanation of the Command:

- `+tcp` : Forces the query to use TCP instead of the default UDP.
- `@131.252.208.53` : Specifies the DNS server to query, in this case, PSU's local DNS server.
- `www.pdx.edu` : The domain name you want to look up.
- `A` : Specifies that you want the A record, which provides the IPv4 address for the domain.

This command will query the DNS server at 131.252.208.53 for the IPv4 address of www.pdx.edu using TCP.

- Take a screenshot of the records returned for your lab notebook.

```
agah@ada:~$ dig +tcp @131.252.208.53 www.pdx.edu A

; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> +tcp @131.252.208.53 www.pdx.edu A
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60178
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 3e2c656aefd475ae01000000670aedef48837f58a894e4c06 (good)
;; QUESTION SECTION:
;www.pdx.edu.                IN      A

;; ANSWER SECTION:
www.pdx.edu.                 60      IN      A       108.138.94.85
www.pdx.edu.                 60      IN      A       108.138.94.58
www.pdx.edu.                 60      IN      A       108.138.94.13
www.pdx.edu.                 60      IN      A       108.138.94.27

;; Query time: 78 msec
;; SERVER: 131.252.208.53#53(131.252.208.53) (TCP)
;; WHEN: Sat Oct 12 14:45:24 PDT 2024
;; MSG SIZE rcvd: 132
```

```

agah@ada:~$ dig @131.252.208.53 pdx.edu MX

; <<> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<> @131.252.208.53 pdx.edu MX
; (1 server found)
;; global options: +cmd
;; Got answer:
;; -->HEADER<-- opcode: QUERY, status: NOERROR, id: 27560
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 140d389359642ca701000000670af4c5fc79bd85048e212a (good)
;; QUESTION SECTION:
;pdx.edu.                IN      MX

;; ANSWER SECTION:
pdx.edu.                2735    IN      MX      5 alt1.aspmx.l.google.com.
pdx.edu.                2735    IN      MX      10 alt4.aspmx.l.google.com.
pdx.edu.                2735    IN      MX      1 aspmx.l.google.com.
pdx.edu.                2735    IN      MX      5 alt2.aspmx.l.google.com.
pdx.edu.                2735    IN      MX      10 alt3.aspmx.l.google.com.

;; Query time: 2 msec
;; SERVER: 131.252.208.53#53(131.252.208.53) (UDP)
;; WHEN: Sat Oct 12 15:14:29 PDT 2024
;; MSG SIZE rcvd: 182

```

- What cloud provider hosts the web site for [www.pdx.edu](http://www.pdx.edu)?

Amazon Web Services (AWS) hosts the website for [www.pdx.edu](http://www.pdx.edu) through its CloudFront service.

```

agah@ada:~$ whois 108.138.94.85

#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/resources/registry/whois/tou/
#
# If you see inaccuracies in the results, please report at
# https://www.arin.net/resources/registry/whois/inaccuracy\_reporting/
#
# Copyright 1997–2024, American Registry for Internet Numbers, Ltd.
#

# start

NetRange:      108.128.0.0 - 108.139.255.255
CIDR:          108.136.0.0/14, 108.128.0.0/13
NetName:       AMAZO-4
NetHandle:     NET-108-128-0-0-1
Parent:        NET108 (NET-108-0-0-0-0)
NetType:       Direct Allocation
OriginAS:
Organization:  Amazon.com, Inc. (AMAZO-4)
RegDate:       2018-09-18
Updated:       2018-09-18
Ref:           https://rdap.arin.net/registry/ip/108.128.0.0

OrgName:       Amazon.com, Inc.
OrgId:         AMAZO-4
Address:       Amazon Web Services, Inc.
Address:       P.O. Box 81226
City:          Seattle
StateProv:     WA
PostalCode:    98108-1226
Country:       US
RegDate:       2005-09-29
Updated:       2022-09-30
Comment:       For details of this service please see
Comment:       http://ec2.amazonaws.com
Ref:           https://rdap.arin.net/registry/entity/AMAZO-4

```

- What cloud provider handles mail for `pdx.edu`?

The MX record query for `pdx.edu` returned mail servers such as `aspmx.l.google.com`, `alt1.aspmx.l.google.com`, etc.

These servers are part of Google Workspace (formerly G Suite), which indicates that **Google handles email for pdx.edu.**

mashimaro.cs.pdx.edu

- Take a screenshot of the results for both records for your lab notebook.

```
agah@ada:~$ dig cs.pdx.edu NS

; <<> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<> cs.pdx.edu NS
;; global options: +cmd
;; Got answer:
;; -->HEADER<-- opcode: QUERY, status: NOERROR, id: 13134
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 7

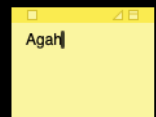
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;cs.pdx.edu.                IN      NS

;; ANSWER SECTION:
cs.pdx.edu.                14400   IN      NS      adns3.cat.pdx.edu.
cs.pdx.edu.                14400   IN      NS      adns2.cat.pdx.edu.
cs.pdx.edu.                14400   IN      NS      adns1.cat.pdx.edu.

;; ADDITIONAL SECTION:
adns1.cat.pdx.edu.        5006    IN      A        131.252.208.38
adns2.cat.pdx.edu.        5006    IN      A        35.83.149.241
adns3.cat.pdx.edu.        5006    IN      A        52.11.85.139
adns1.cat.pdx.edu.        5006    IN      AAAA     2610:10:20:208::38
adns2.cat.pdx.edu.        5006    IN      AAAA     2600:1f13:d7a:ef00:e934:6a5c:f7c4:3e64
adns3.cat.pdx.edu.        5006    IN      AAAA     2600:1f13:d7a:ef00:d17c:c07a:9bb8:348e

;; Query time: 5 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Sat Oct 12 15:28:20 PDT 2024
;; MSG SIZE rcvd: 235

agah@ada:~$
```





```
agah@ada:~$ dig @131.252.208.38 mashimaro.cs.pdx.edu A

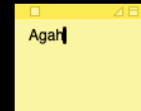
; <<> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<> @131.252.208.38 mashimaro.cs.pdx.edu A
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29442
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 06b4153f7c282a4c0100000670b0652f02b1e993b1e1a7f (good)
;; QUESTION SECTION:
;mashimaro.cs.pdx.edu.          IN      A

;; ANSWER SECTION:
mashimaro.cs.pdx.edu.  14400   IN      A      131.252.220.66

;; Query time: 0 msec
;; SERVER: 131.252.208.38#53(131.252.208.38) (UDP)
;; WHEN: Sat Oct 12 16:29:22 PDT 2024
;; MSG SIZE rcvd: 93

agah@ada:~$
```



## 2.2.2 Iterative DNS lookups

- List all of the iterative dig commands performed for the lookup
  1. Dig
  2. dig @192.5.5.241 cloud.google.com NS +norecurse +tcp
  3. dig @192.43.172.30 google.com NS +norecurse +tcp
  4. dig @216.239.32.10 cloud.google.com NS +norecurse +tcp
  5. dig @216.239.32.10 console.cloud.google.com A +norecurse +tcp
- Take a screenshot of the results of the final query for your lab notebook.

```
agah@ada:~$ dig @216.239.32.10 console.cloud.google.com A +norecurse +tcp

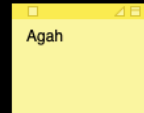
; <<> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<> @216.239.32.10 console.cloud.google.com A +norecurse +tcp
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49726
;; flags: qr aa; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags;; udp: 512
;; QUESTION SECTION:
;console.cloud.google.com.      IN      A

;; ANSWER SECTION:
console.cloud.google.com. 300     IN      CNAME   www3.l.google.com.
www3.l.google.com.       300     IN      A       142.250.69.206

;; Query time: 29 msec
;; SERVER: 216.239.32.10#53(216.239.32.10) (TCP)
;; WHEN: Sat Oct 12 16:52:24 PDT 2024
;; MSG SIZE rcvd: 90

agah@ada:~$
```



## 2.2.3 Reverse DNS lookups

Aliases and reverse lookups

- Take a screenshot of the prompt and the command produced

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

You can use the following single command to find all IPv4 addresses that `espn.go.com` points to by combining `dig`, `egrep`, and `awk`:

bash

Copy code

```
dig +short espn.go.com | egrep '^([0-9]+\.[0-9]+\.[0-9]+\.[0-9]+)$' | awk '{print $1}'
```

## Command Breakdown

1. `dig +short espn.go.com`:

- `dig`: A command-line tool for querying DNS servers.
- `+short`: Provides a simplified output, listing only the IP addresses without extra information.
- `espn.go.com`: The domain you're querying for A records (IPv4 addresses).

2. `| egrep '^([0-9]+\.[0-9]+\.[0-9]+\.[0-9]+)$'`:

- `egrep`: Filters the output based on a regular expression.
- `'^([0-9]+\.[0-9]+\.[0-9]+\.[0-9]+)$'`: A regular expression that matches only lines with IPv4 addresses (in the format `X.X.X.X`, where X is a number).
- This filters out any other records (such as IPv6 addresses) that may appear.

3. `| awk '{print $1}'`:

- `awk '{print $1}'`: Extracts the first field from each line (the IP address) and prints it.
- Since only the IP addresses are passed through from the previous step, this simply outputs the list of IPv4 addresses.

- Take a screenshot of its results for your lab notebook

```
agah@ada:~$ dig +short espn.go.com | egrep '^([0-9]+\.[0-9]+\.[0-9]+\.[0-9]+)$' | awk '{print $1}'
18.161.6.94
18.161.6.89
18.161.6.38
18.161.6.80
agah@ada:~$
```

- Take a screenshot of the command and its results for your lab notebook

```

agah@ada:~$ for ip in $(dig +short espn.go.com | egrep '^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+$' | awk '{print $1}'); do
    dig +short -x "$ip"
done | egrep '^[a-zA-Z0-9.-]+$' | awk '{print $1}'
server-18-161-6-80.hio52.r.cloudfront.net.
server-18-161-6-89.hio52.r.cloudfront.net.
server-18-161-6-94.hio52.r.cloudfront.net.
server-18-161-6-38.hio52.r.cloudfront.net.
agah@ada:~$ █

```

## 2.2.4 Host enumeration

- Take a screenshot of the results in your lab notebook

```

agah@ada:~$
agah@ada:~$ for ip in 131.252.220.{0..255}; do    dig +short -x "$ip"; done | e
grep '^[a-zA-Z0-9.-]+$' | awk '{print $1}' > 220hosts.txt
agah@ada:~$ grep -i -E "ford|toyota|honda|bmw|aidu" 220hosts.txt
bmw.cs.pdx.edu.
ford.cs.pdx.edu.
honda.cs.pdx.edu.
toyota.cs.pdx.edu.
agah@ada:~$ █

```

## 2.2.5 Geographic DNS #2

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

## Geolocation data from

ipinfo.io

Product: API, real-time




**IP ADDRESS:** 131.252.208.53



**ISP:** Not available



**COUNTRY:** United States 



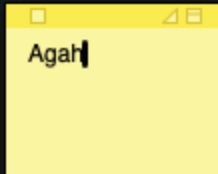
**ORGANIZATION:** AS6366  
Portland State  
University



**REGION:** Oregon



**CITY:** Portland



**LATITUDE:** 45.5234



**LONGITUDE:** -122.6762

Incorrect location? [Contact ipinfo.io](#)



[view map](#)

## Geolocation data from

DB-IP

Product: API, real-time




**IP ADDRESS:** 131.252.208.53



**ISP:** Portland State University



**COUNTRY:** United States 



**ORGANIZATION:** Portland State  
University



**REGION:** Oregon



**CITY:** Portland (North Portland)



**LATITUDE:** 45.584




**LONGITUDE:** -122.728



Incorrect location? [Contact DB-IP](#)





[view map](#)

Geolocation data from


 IP ADDRESS: 198.82.247.66


 COUNTRY: United States 


 REGION: Virginia


 CITY: Blacksburg

ipinfo.io

 ISP: Not available


 ORGANIZATION: AS1312 Virginia Polytechnic Institute and State Univ.

 LATITUDE: 37.2296


 LONGITUDE: -80.4139



Product: API, real-time


Incorrect location? Contact [ipinfo.io](https://ipinfo.io)


 [view map](#)

Geolocation data from


 IP ADDRESS: 198.82.247.66


 COUNTRY: United States 


 REGION: Virginia


 CITY: Blacksburg (Farmview - Ramble)

DB-IP

 ISP: Virginia Polytechnic Institute and State Univ.

 ORGANIZATION: Virginia Polytechnic Institute and State Univ.

 LATITUDE: 37.2037

 LONGITUDE: -80.4143

Product: API, real-time

- 
- Record one address for [www.google.com](https://www.google.com) from each result for your lab notebook.

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

; <<> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<> @131.252.208.53 www.google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45427
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: fec17b15411aa00d01000000670b6911b6e269247eb41ab2 (good)
;; QUESTION SECTION:
;www.google.com.                IN      A

;; ANSWER SECTION:
www.google.com.                219     IN      A      142.250.217.100

;; Query time: 1 msec
;; SERVER: 131.252.208.53#53(131.252.208.53) (UDP)
;; WHEN: Sat Oct 12 23:30:41 PDT 2024
;; MSG SIZE rcvd: 87
```

```

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

; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> @198.82.247.66 www.google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 50099
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 94c245ffffaf9fa69793e08e0670b69793751ee299b4bb83e (good)
;; QUESTION SECTION:
;www.google.com.                IN      A

;; ANSWER SECTION:
www.google.com.      24      IN      A      142.251.167.104
www.google.com.      24      IN      A      142.251.167.106
www.google.com.      24      IN      A      142.251.167.103
www.google.com.      24      IN      A      142.251.167.147
www.google.com.      24      IN      A      142.251.167.105
www.google.com.      24      IN      A      142.251.167.99

;; Query time: 70 msec
;; SERVER: 198.82.247.66#53(198.82.247.66) (UDP)
;; WHEN: Sat Oct 12 23:32:25 PDT 2024
;; MSG SIZE rcvd: 167

```

- What are the geographic coordinates of each DNS server and the IP address it resolves for www.google.com?

IP addresses are included in the screenshot below.










- Take a screenshot of the results for your lab notebook.




Geolocation data from

IP2Location

Product: DB6, 2024-10-1

	IP ADDRESS: 142.250.217.100		ISP: Google LLC
	COUNTRY: United States 		ORGANIZATION: Not available
	REGION: Washington		LATITUDE: 47.6043
	CITY: Seattle		LONGITUDE: -122.3298










Incorrect location? Contact IP2Location

 view map

Geolocation data from

ipinfo.io

Product: API, real-time

	IP ADDRESS: 142.250.217.100		ISP: Not available
	COUNTRY: United States 		ORGANIZATION: AS15169 Google LLC
	REGION: Washington		LATITUDE: 47.6062
	CITY: Seattle		LONGITUDE: -122.3321

**Geolocation data from**

**ipinfo.io**

**Product: API, real-time**

**IP ADDRESS:** 142.251.167.104

**ISP:** Not available

**COUNTRY:** United States

**ORGANIZATION:** AS15169  
Google LLC

**REGION:** Virginia

**LATITUDE:** 38.9687

**CITY:** Reston

**LONGITUDE:** -77.3411

Incorrect location? Contact ipinfo.io

view map

**Geolocation data from**

**DB-IP**

**Product: API, real-time**

**IP ADDRESS:** 142.251.167.104

**ISP:** Google LLC

**COUNTRY:** United States

**ORGANIZATION:** Google LLC

**REGION:** California

**LATITUDE:** 37.4225

**CITY:** Mountain View

**LONGITUDE:** -122.085

Incorrect location? Contact DB-IP

view map

## 2.2.6 Wireshark Lab #3

```

ahmadagah@course-vm:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 42:01:0a:8a:00:02 brd ff:ff:ff:ff:ff:ff
    inet 10.138.0.2/32 metric 100 scope global dynamic ens4
        valid_lft 71051sec preferred_lft 71051sec
    inet6 fe80::4001:aff:fe8a:2/64 scope link
        valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:b0:f6:25:6f brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
  
```

```

ahmadagah@course-vm:~$ ip route show
default via 10.138.0.1 dev ens4 proto dhcp src 10.138.0.2 metric 100
10.138.0.1 dev ens4 proto dhcp scope link src 10.138.0.2 metric 100
169.254.169.254 via 10.138.0.1 dev ens4 proto dhcp src 10.138.0.2 metric 100
172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1 linkdown
ahmadagah@course-vm:~$

```

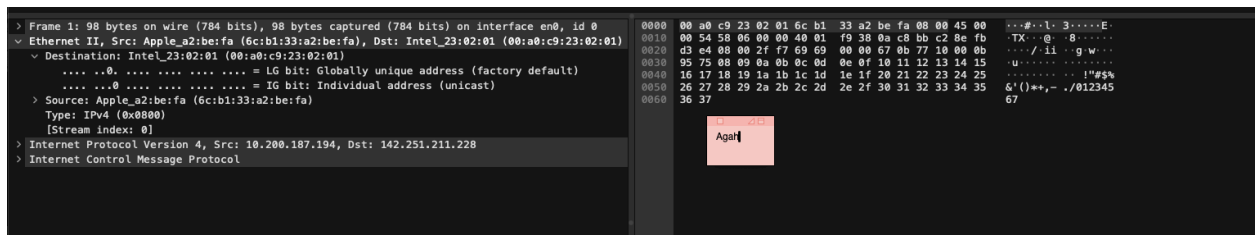
The IP address of the VM ----- src 10.138.0.2

The name of the local virtual ethernet interface ----- dev ens

The IP address of the default router ----- default via 10.138.0.1

## 2.2.7 -

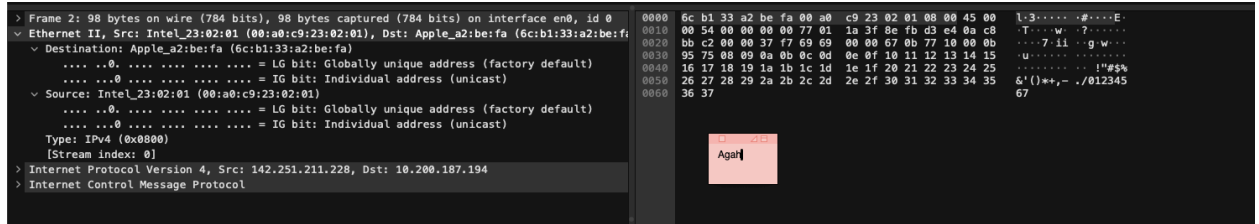
- Take a screenshot of the bytes in the packet dump window as shown below



- Does the destination MAC address correspond to an interface on the VM, an interface on the default router or an interface on Google's web site?

the destination MAC address (00:a0:c9:23:02:01) represents the MAC of my local router interface, which the VM is using to reach Google's servers on the internet.

- Does the destination MAC address correspond to an interface on the VM, an interface on the default router or an interface on Google's web site?



The destination MAC address (6c:b1:33:a2:be:fa) in this second packet corresponds to the VM, showing that this packet is directed back to your VM after being processed by the router and Google's servers.

## 2.2.8 Network Recap Lab #4

## 2.2.9 Collect trace

## 2.2.10 Analyze trace

- Take a screenshot of the all of the packets returned within Wireshark that includes their packet numbers

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.200.187.194	142.251.33.78	TCP	54	53488 → 443 [ACK] Seq=1 Ack=1 Win=2048 Len=0
2	0.000312	10.200.187.194	35.186.224.24	TCP	54	62994 → 443 [ACK] Seq=1 Ack=1 Win=2048 Len=0
3	0.009596	142.251.33.78	10.200.187.194	TCP	66	[TCP ACKed unseen segment] 443 → 53488 [ACK] Seq=1 Ack=2
4	0.014955	35.186.224.24	10.200.187.194	TCP	66	[TCP ACKed unseen segment] 443 → 62994 [ACK] Seq=1 Ack=2
5	0.688854	10.200.187.194	17.248.232.64	TLSv1...	90	Application Data
6	0.689870	10.200.187.194	17.248.232.64	TCP	66	57721 → 443 [FIN, ACK] Seq=25 Ack=1 Win=2048 Len=0 TSval=
7	0.703263	17.248.232.64	10.200.187.194	TCP	66	443 → 57721 [ACK] Seq=1 Ack=25 Win=126 Len=0 TSval=207123
8	0.703266	17.248.232.64	10.200.187.194	TCP	66	443 → 57721 [FIN, ACK] Seq=1 Ack=25 Win=126 Len=0 TSval=2
9	0.703601	17.248.232.64	10.200.187.194	TCP	66	443 → 57721 [ACK] Seq=2 Ack=26 Win=126 Len=0 TSval=207123
10	0.703768	10.200.187.194	17.248.232.64	TCP	66	[TCP Retransmission] 57721 → 443 [FIN, ACK] Seq=25 Ack=2
11	0.824189	10.200.187.194	44.229.222.36	TLSv1...	120	Application Data
12	0.833517	44.229.222.36	10.200.187.194	TLSv1...	122	Application Data
13	0.833713	10.200.187.194	44.229.222.36	TCP	66	52661 → 443 [ACK] Seq=55 Ack=57 Win=2047 Len=0 TSval=6800
14	2.277466	10.200.187.194	170.114.4.221	TLSv1...	96	Application Data
15	2.393538	170.114.4.221	10.200.187.194	TCP	66	443 → 52588 [ACK] Seq=1 Ack=31 Win=57 Len=0 TSval=2571542
16	2.828249	10.200.187.194	35.186.224.46	TLSv1...	98	Application Data
17	2.828466	10.200.187.194	35.186.224.46	TLSv1...	98	Application Data
18	2.828549	10.200.187.194	35.186.224.46	TLSv1...	98	Application Data
19	2.828614	10.200.187.194	35.186.224.46	TLSv1...	98	Application Data
20	2.828687	10.200.187.194	35.186.224.46	TLSv1...	98	Application Data
21	2.828745	10.200.187.194	35.186.224.46	TLSv1...	98	Application Data
22	2.828803	10.200.187.194	35.186.224.46	TLSv1...	98	Application Data
23	2.828858	10.200.187.194	35.186.224.46	TLSv1...	98	Application Data
24	2.828944	10.200.187.194	35.186.224.46	TLSv1...	98	Application Data
25	2.829005	10.200.187.194	35.186.224.46	TLSv1...	98	Application Data
26	2.829098	10.200.187.194	35.186.224.46	TLSv1...	98	Application Data
27	2.829167	10.200.187.194	35.186.224.46	TLSv1...	98	Application Data
28	2.829225	10.200.187.194	35.186.224.46	TLSv1...	98	Application Data
29	2.829277	10.200.187.194	35.186.224.46	TLSv1...	98	Application Data
30	2.829329	10.200.187.194	35.186.224.46	TLSv1...	98	Application Data
31	2.829391	10.200.187.194	35.186.224.46	TLSv1...	98	Application Data
32	2.829462	10.200.187.194	35.186.224.46	TLSv1...	98	Application Data
33	2.836479	35.186.224.46	10.200.187.194	TCP	66	443 → 57443 [ACK] Seq=1 Ack=353 Win=837 Len=0 TSval=89082
34	2.836854	10.200.187.194	35.83.181.24	TLSv1...	122	Application Data
35	2.837654	35.186.224.46	10.200.187.194	TCP	66	443 → 57443 [ACK] Seq=1 Ack=481 Win=837 Len=0 TSval=89082

Frame 351: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on 0000 6c b1 33 a2 be fa 00 a0 c9 23 02 01 08 06 00 01 1.3.....#

network\_trace.pcap Packets: 484 Profile: Default

## ARP

- What packet numbers in the trace are the result of the VM attempting to get the hardware address of the default router?

network\_trace.pcap

arp

No.	Time	Source	Destination	Protocol	Length	Info
350	18.281478	Apple_a2:be:fa	Broadcast	ARP	42	Who has 10.200.176.1? Tell 10.200.187.194
351	18.290292	Intel_23:02:01	Apple_a2:be:fa	ARP	56	10.200.176.1 is at 00:a0:c9:23:02:01

Frame 351: 56 bytes on wire (448 bits), 56 bytes captured (448 bits)  
 Ethernet II, Src: Intel\_23:02:01 (00:a0:c9:23:02:01), Dst: Apple\_a2:b  
 Address Resolution Protocol (reply)

```

0000  6c b1 33 a2 be fa 00 a0 c9 23 02 01 08 06 00 01  l.3....#..
0010  08 00 06 04 00 02 00 a0 c9 23 02 01 0a c8 b0 01  .....#..
0020  6c b1 33 a2 be fa 0a c8 bb c2 00 00 00 00 00 00  l.3....#..
0030  00 00 00 00 00 00 00 00
  
```

network\_trace.pcap

arp

No.	Time	Source	Destination	Protocol	Length	Info
350	18.281478	Apple_a2:be:fa	Broadcast	ARP	42	Who has 10.200.176.1? Tell 10.200.187.194
351	18.290292	Intel_23:02:01	Apple_a2:be:fa	ARP	56	10.200.176.1 is at 00:a0:c9:23:02:01

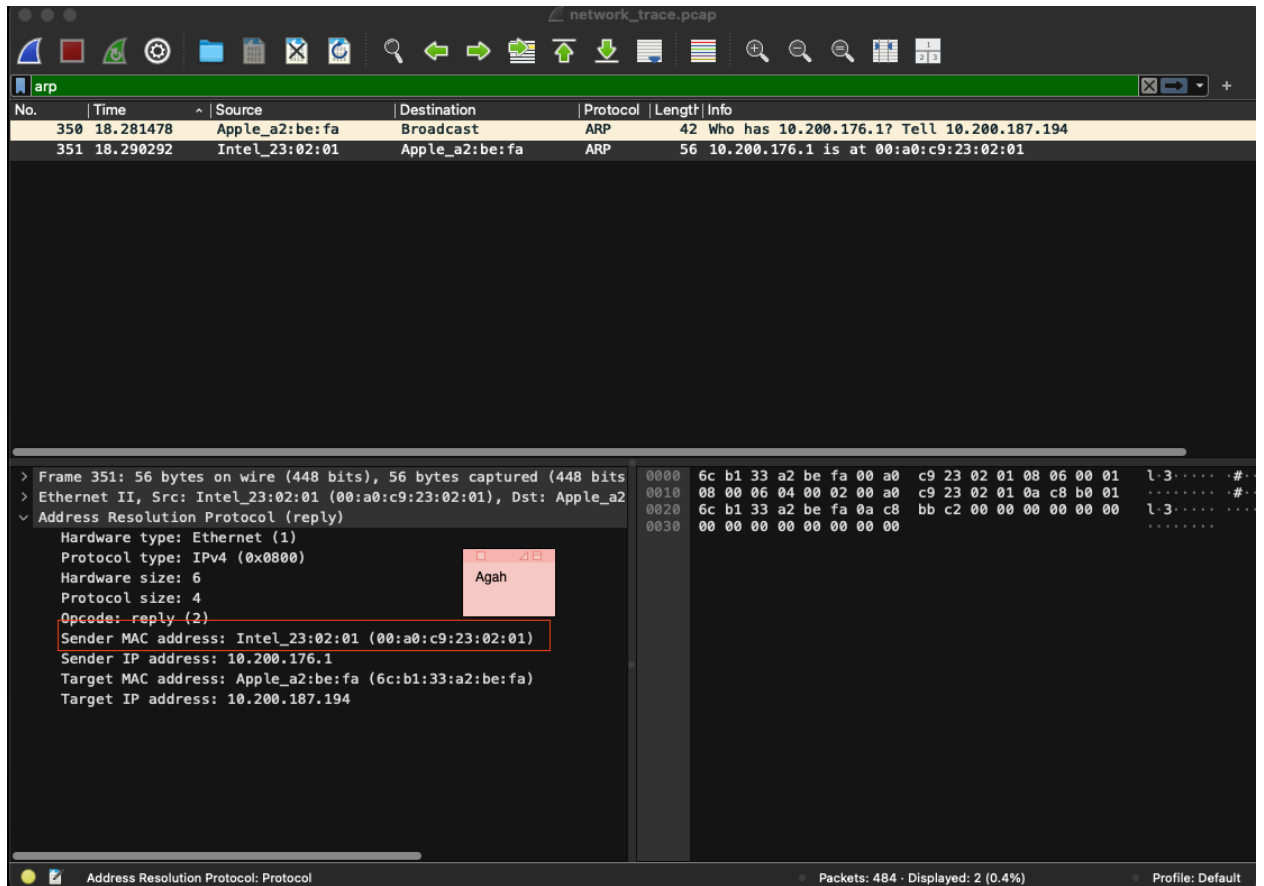
Frame 350: 42 bytes on wire (336 bits), 42 bytes captured (336 bits)  
 Ethernet II, Src: Apple\_a2:be:fa (6c:b1:33:a2:be:fa), Dst: Broadcast  
 Address Resolution Protocol (request)

```

0000  ff ff ff ff ff 6c b1 33 a2 be fa 08 06 00 01  .....l.3..
0010  08 00 06 04 00 01 6c b1 33 a2 be fa 0a c8 bb c2  .....l.3..
0020  00 00 00 00 00 00 0a c8 b0 01
  
```

Packet 350 and 351

- What is this hardware address?



network\_trace.pcap

arp

No.	Time	Source	Destination	Protocol	Length	Info
350	18.281478	Apple_a2:be:fa	Broadcast	ARP	42	Who has 10.200.176.1? Tell 10.200.187.194
351	18.290292	Intel_23:02:01	Apple_a2:be:fa	ARP	56	10.200.176.1 is at 00:a0:c9:23:02:01

> Frame 351: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface 0

> Ethernet II, Src: Intel\_23:02:01 (00:a0:c9:23:02:01), Dst: Apple\_a2:be:fa (6c:b1:33:a2:be:fa)

Address Resolution Protocol (reply)

Hardware type: Ethernet (1)

Protocol type: IPv4 (0x0800)

Hardware size: 6

Protocol size: 4

Opcode: reply (2)

Sender MAC address: Intel\_23:02:01 (00:a0:c9:23:02:01)

Sender IP address: 10.200.176.1

Target MAC address: Apple\_a2:be:fa (6c:b1:33:a2:be:fa)

Target IP address: 10.200.187.194

0000 6c b1 33 a2 be fa 00 a0 c9 23 02 01 08 06 00 01 1:3.....#..

0010 08 00 06 04 00 02 00 a0 c9 23 02 01 0a c8 b0 01 .....#..

0020 6c b1 33 a2 be fa 0a c8 bb c2 00 00 00 00 00 00 1:3.....#..

0030 00 00 00 00 00 00 00 00

Address Resolution Protocol: Protocol

Packets: 484 · Displayed: 2 (0.4%)

Profile: Default

DNS

- What packet numbers in the trace correspond to the DNS request for the web site?

No.	Time	Source	Destination	Protocol	Length	Info
66	5.947098	10.200.187.194	131.252.110.100	DNS	76	Standard query 0xbdf8 A in.grammarly.com
67	5.947178	10.200.187.194	131.252.110.100	DNS	76	Standard query 0xa80f HTTPS in.grammarly.com
68	5.950111	131.252.110.100	10.200.187.194	DNS	158	Standard query response 0xa80f HTTPS in.grammarly.com SO
69	5.950422	131.252.110.100	10.200.187.194	DNS	308	Standard query response 0xbdf8 A in.grammarly.com A 50.1
75	6.052181	10.200.187.194	131.252.110.100	DNS	78	Standard query 0x139e A gnar.grammarly.com
76	6.052600	10.200.187.194	131.252.110.100	DNS	78	Standard query 0xe266 HTTPS gnar.grammarly.com
77	6.055065	131.252.110.100	10.200.187.194	DNS	312	Standard query response 0x139e A gnar.grammarly.com A 18
78	6.056603	131.252.110.100	10.200.187.194	DNS	160	Standard query response 0xe266 HTTPS gnar.grammarly.com
190	14.837802	10.200.187.194	131.252.110.100	DNS	86	Standard query 0x3e41 A waa-pa.clients6.google.com
191	14.837842	10.200.187.194	131.252.110.100	DNS	86	Standard query 0x56b0 HTTPS waa-pa.clients6.google.com
196	14.841181	131.252.110.100	10.200.187.194	DNS	102	Standard query response 0x3e41 A waa-pa.clients6.google.com
197	14.841571	131.252.110.100	10.200.187.194	DNS	136	Standard query response 0x56b0 HTTPS waa-pa.clients6.google.com
397	25.324230	10.200.187.194	131.252.110.100	DNS	92	Standard query 0xd101 A extension.femetrics.grammarly.io
398	25.324357	10.200.187.194	131.252.110.100	DNS	92	Standard query 0xf83c HTTPS extension.femetrics.grammarly.io
399	25.329481	131.252.110.100	10.200.187.194	DNS	179	Standard query response 0xf83c HTTPS extension.femetrics.grammarly.io
400	25.329835	131.252.110.100	10.200.187.194	DNS	360	Standard query response 0xd101 A extension.femetrics.grammarly.io
429	25.594717	10.200.187.194	131.252.110.100	DNS	80	Standard query 0xd116 A beacons.dcp.qvt2.com

- What is the IP address of the local DNS server being queried?

No.	Time	Source	Destination	Protocol	Length	Info
66	5.947098	10.200.187.194	131.252.110.100	DNS	76	Standard query 0xbdf8 A in.grammarly.com
67	5.947178	10.200.187.194	131.252.110.100	DNS	76	Standard query 0xa80f HTTPS in.grammarly.com
68	5.950111	131.252.110.100	10.200.187.194	DNS	158	Standard query response 0xa80f HTTPS in.grammarly.com SO
69	5.950422	131.252.110.100	10.200.187.194	DNS	308	Standard query response 0xbdf8 A in.grammarly.com A 50.1
75	6.052181	10.200.187.194	131.252.110.100	DNS	78	Standard query 0x139e A gnar.grammarly.com
76	6.052600	10.200.187.194	131.252.110.100	DNS	78	Standard query 0xe266 HTTPS gnar.grammarly.com
77	6.055065	131.252.110.100	10.200.187.194	DNS	312	Standard query response 0x139e A gnar.grammarly.com A 18
78	6.056603	131.252.110.100	10.200.187.194	DNS	160	Standard query response 0xe266 HTTPS gnar.grammarly.com
190	14.837802	10.200.187.194	131.252.110.100	DNS	86	Standard query 0x3e41 A waa-pa.clients6.google.com
191	14.837842	10.200.187.194	131.252.110.100	DNS	86	Standard query 0x56b0 HTTPS waa-pa.clients6.google.com
196	14.841181	131.252.110.100	10.200.187.194	DNS	102	Standard query response 0x3e41 A waa-pa.clients6.google.com
197	14.841571	131.252.110.100	10.200.187.194	DNS	136	Standard query response 0x56b0 HTTPS waa-pa.clients6.google.com
397	25.324230	10.200.187.194	131.252.110.100	DNS	92	Standard query 0xd101 A extension.femetrics.grammarly.io
398	25.324357	10.200.187.194	131.252.110.100	DNS	92	Standard query 0xf83c HTTPS extension.femetrics.grammarly.io
399	25.329481	131.252.110.100	10.200.187.194	DNS	179	Standard query response 0xf83c HTTPS extension.femetrics.grammarly.io
400	25.329835	131.252.110.100	10.200.187.194	DNS	360	Standard query response 0xd101 A extension.femetrics.grammarly.io
429	25.594717	10.200.187.194	131.252.110.100	DNS	80	Standard query 0xd116 A beacons.dcp.qvt2.com

> Frame 66: 76 bytes on wire (608 bits), 76 bytes captured (608 bits)	0000	00 a0 c9 23 02 01 6c b1 33 a2 be fa 08 00 45 00	...
> Ethernet II, Src: Apple_a2:be:fa (6c:b1:33:a2:be:fa), Dst: Intel_23	0010	00 3e cc 45 00 00 40 11 f5 7e 0a c8 bb c2 83 fc	...
> Internet Protocol Version 4, Src: 10.200.187.194, Dst: 131.252.110.100	0020	6e 64 4d ee 00 35 00 2a d6 f6 bd f8 01 00 00 01	...
0100 .... = Version: 4	0030	00 00 00 00 00 00 02 69 6e 09 67 72 61 6d 6d 61	...
.... 0101 = Header Length: 20 bytes (5)	0040	72 6c 79 03 63 6f 6d 00 00 01 00 01	...
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)			
Total Length: 62			
Identification: 0xcc45 (52293)			
> 000. .... = Flags: 0x0			
...0 0000 0000 0000 = Fragment Offset: 0			
Time to Live: 64			
Protocol: UDP (17)			
Header Checksum: 0xf57e [validation disabled]			
[Header checksum status: Unverified]			
Source Address: 10.200.187.194			
Destination Address: 131.252.110.100			
[Stream index: 11]			
> User Datagram Protocol, Src Port: 19950, Dst Port: 53			
> Domain Name System (query)			



## TCP

- What packet numbers in the trace correspond to the initial TCP handshake for the web request?

No.	Time	Source	Destination	Protocol	Length	Info
70	5.950823	10.200.187.194	50.17.194.57	TCP	78	54806 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=11151
78	6.057291	10.200.187.194	18.214.27.234	TCP	78	54807 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=27181
181	14.833400	10.200.187.194	51.222.39.185	TCP	78	54844 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=22877
198	14.841733	10.200.187.194	142.251.211.234	TCP	78	54845 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=39377
237	14.882364	10.200.187.194	142.251.211.234	TCP	78	54846 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=50554
365	21.644924	10.200.187.194	35.233.233.233	TCP	78	54875 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=290308
401	25.330308	10.200.187.194	3.221.5.109	TCP	78	54892 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=75225

- How long does it take to perform the initial TCP handshake?

No.	Time	Source	Destination	Protocol	Length	Info
62	5.834932	35.83.181.24	10.200.187.194	TLSv1...	122	Application Data
63	5.835095	10.200.187.194	35.83.181.24	TCP	66	52736 → 443 [ACK] Seq=111 Ack=57 Win=2047 Len=0 TSval=221
64	5.930342	170.114.52.3	10.200.187.194	TCP	66	443 → 54225 [ACK] Seq=1 Ack=1 Win=9 Len=0 TSval=295391906
65	5.930575	10.200.187.194	170.114.52.3	TCP	66	[TCP ACKed unseen segment] 54225 → 443 [ACK] Seq=1 Ack=2
66	5.947098	10.200.187.194	131.252.110.100	DNS	76	Standard query 0xbdf8 A in.grammarly.com
67	5.947178	10.200.187.194	131.252.110.100	DNS	76	Standard query 0xa80f HTTPS in.grammarly.com
68	5.950111	131.252.110.100	10.200.187.194	DNS	158	Standard query response 0xa80f HTTPS in.grammarly.com 504
69	5.950422	131.252.110.100	10.200.187.194	DNS	308	Standard query response 0xbdf8 A in.grammarly.com A 50.17
70	5.950823	10.200.187.194	50.17.194.57	TCP	78	54806 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TS
71	6.019784	50.17.194.57	10.200.187.194	TCP	74	443 → 54806 [SYN, ACK] Seq=0 Ack=1 Win=26847 Len=0 MSS=1
72	6.020073	10.200.187.194	50.17.194.57	TCP	66	54806 → 443 [ACK] Seq=1 Ack=1 Win=131200 Len=0 TSval=111
73	6.021261	10.200.187.194	50.17.194.57	TCP	1304	54806 → 443 [ACK] Seq=1 Ack=1 Win=131200 Len=1238 TSval=1
74	6.021297	10.200.187.194	50.17.194.57	TLSv1...	723	Client Hello (SNI=in.grammarly.com)
75	6.052181	10.200.187.194	131.252.110.100	DNS	78	Standard query 0x139e A gnar.grammarly.com
76	6.052600	10.200.187.194	131.252.110.100	DNS	78	Standard query 0xe266 HTTPS gnar.grammarly.com
77	6.055065	131.252.110.100	10.200.187.194	DNS	312	Standard query response 0x139e A gnar.grammarly.com A 18
78	6.056603	131.252.110.100	10.200.187.194	DNS	160	Standard query response 0xe266 HTTPS gnar.grammarly.com 5
79	6.057291	10.200.187.194	18.214.27.234	TCP	78	54807 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TS
80	6.090338	50.17.194.57	10.200.187.194	TCP	66	443 → 54806 [ACK] Seq=1 Ack=1896 Win=44800 Len=0 TSval=24
81	6.091762	50.17.194.57	10.200.187.194	TLSv1...	1304	Server Hello
82	6.091764	50.17.194.57	10.200.187.194	TCP	1304	443 → 54806 [ACK] Seq=1239 Ack=1896 Win=44800 Len=1238 TS

Frame 70: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface 0 (en0)

Section number: 1

Interface id: 0 (en0)

Packet flags: 0x00000002

Encapsulation type: Ethernet (1)

Arrival Time: Oct 13, 2024 01:05:50.881477000 PDT

UTC Arrival Time: Oct 13, 2024 08:05:50.881477000 UTC

Epoch Arrival Time: 172806750.881477000

[Time shift for this packet: 0.000000000 seconds]

[Time delta from previous captured frame: 0.000401000 seconds]

[Time delta from previous displayed frame: 0.000401000 seconds]

[Time since reference or first frame: 5.950823000 seconds]

Frame Number: 70

Frame Length: 78 bytes (624 bits)

Encapsulation type (frame.encap\_type)

Packets: 484

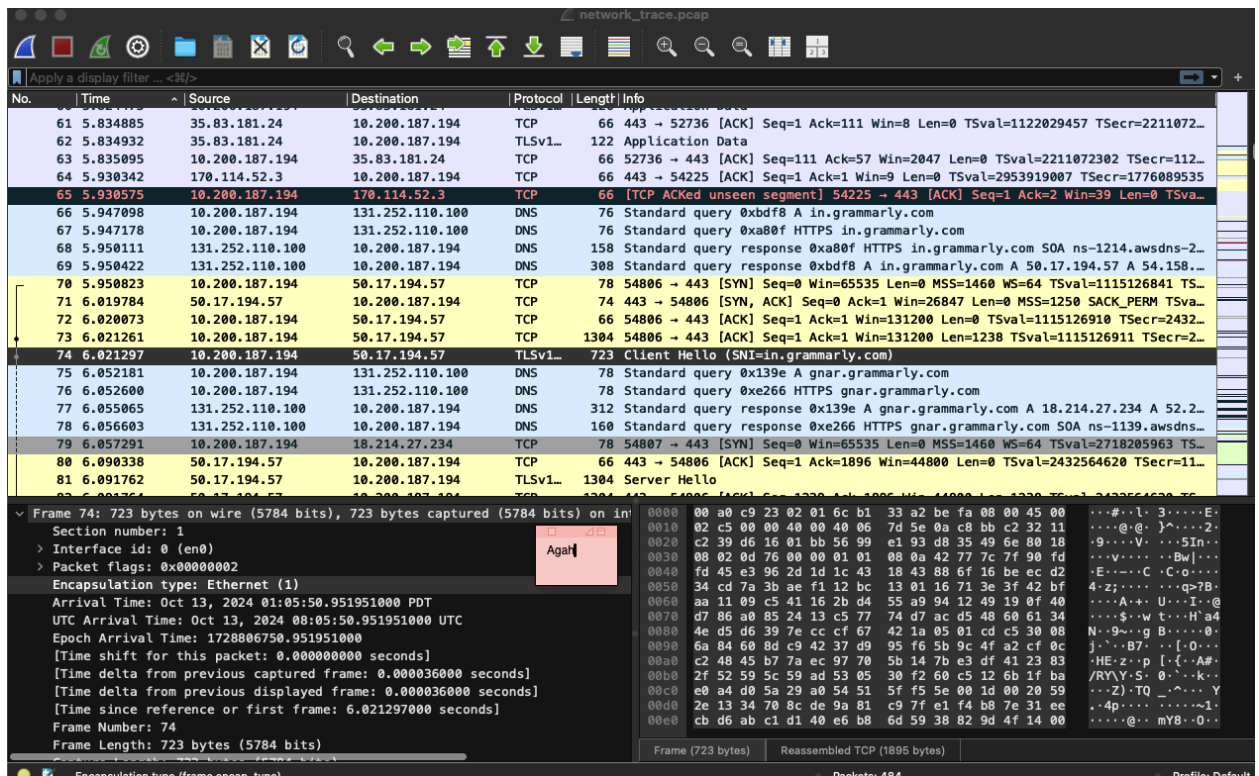
Profile: Default

Subtract the timestamp of the SYN packet from the timestamp of the ACK packet:  $6.020073 - 5.950823 = 0.06925$  seconds  
 $6.020073 - 5.950823 = 0.06925$

## HTTP

- What packet numbers in the trace correspond to the actual HTTP request and response?

## Request



The image shows a Wireshark network trace titled "network\_trace.pcap". The packet list pane displays the following packets:

No.	Time	Source	Destination	Protocol	Length	Info
61	5.834885	35.83.181.24	10.200.187.194	TCP	66	443 → 52736 [ACK] Seq=1 Ack=111 Win=8 Len=0 TSval=1122029457 TSecr=22111072...
62	5.834932	35.83.181.24	10.200.187.194	TLSv1...	122	Application Data
63	5.835095	10.200.187.194	35.83.181.24	TCP	66	52736 → 443 [ACK] Seq=111 Ack=57 Win=2047 Len=0 TSval=2211072302 TSecr=112...
64	5.930342	170.114.52.3	10.200.187.194	TCP	66	443 → 54225 [ACK] Seq=1 Ack=1 Win=9 Len=0 TSval=2953919007 TSecr=1776089535
65	5.930575	10.200.187.194	170.114.52.3	TCP	66	[TCP ACKed unseen segment] 54225 → 443 [ACK] Seq=1 Ack=2 Win=39 Len=0 TSva...
66	5.947098	10.200.187.194	131.252.110.100	DNS	76	Standard query 0xbdf8 A in.grammarly.com
67	5.947178	10.200.187.194	131.252.110.100	DNS	76	Standard query 0xa80f HTTPS in.grammarly.com
68	5.950111	131.252.110.100	10.200.187.194	DNS	158	Standard query response 0xa80f HTTPS in.grammarly.com SOA ns-1214.awsdns-2...
69	5.950422	131.252.110.100	10.200.187.194	DNS	308	Standard query response 0xbdf8 A in.grammarly.com A 50.17.194.57 A 54.158...
70	5.950823	10.200.187.194	50.17.194.57	TCP	78	54806 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=1115126841 TS...
71	6.019784	50.17.194.57	10.200.187.194	TCP	74	443 → 54806 [SYN, ACK] Seq=0 Ack=1 Win=26847 Len=0 MSS=1250 SACK_PERM TSva...
72	6.020073	10.200.187.194	50.17.194.57	TCP	66	54806 → 443 [ACK] Seq=1 Ack=1 Win=131200 Len=0 TSval=1115126910 TSecr=2432...
73	6.021261	10.200.187.194	50.17.194.57	TCP	1304	54806 → 443 [ACK] Seq=1 Ack=1 Win=131200 Len=1238 TSval=1115126911 TSecr=2...
74	6.021297	10.200.187.194	50.17.194.57	TLSv1...	723	Client Hello (SNI=in.grammarly.com)
75	6.052181	10.200.187.194	131.252.110.100	DNS	78	Standard query 0x139e A gnar.grammarly.com
76	6.052600	10.200.187.194	131.252.110.100	DNS	78	Standard query 0xe266 HTTPS gnar.grammarly.com
77	6.055065	131.252.110.100	10.200.187.194	DNS	312	Standard query response 0x139e A gnar.grammarly.com A 18.214.27.234 A 52.2...
78	6.056603	131.252.110.100	10.200.187.194	DNS	160	Standard query response 0xe266 HTTPS gnar.grammarly.com SOA ns-1139.awsdns...
79	6.057291	10.200.187.194	18.214.27.234	TCP	78	54807 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=2718205963 TS...
80	6.090338	50.17.194.57	10.200.187.194	TCP	66	443 → 54806 [ACK] Seq=1 Ack=1896 Win=44800 Len=0 TSval=2432564620 TSecr=11...
81	6.091762	50.17.194.57	10.200.187.194	TLSv1...	1304	Server Hello

The packet details pane for packet 74 (723 bytes) shows the following information:

- Section number: 1
- Interface id: 0 (en0)
- Packet flags: 0x00000002
- Encapsulation type: Ethernet (1)
- Arrival Time: Oct 13, 2024 01:05:50.951951000 PDT
- UTC Arrival Time: Oct 13, 2024 08:05:50.951951000 UTC
- Epoch Arrival Time: 1728806750.951951000
- [Time shift for this packet: 0.000000000 seconds]
- [Time delta from previous captured frame: 0.000036000 seconds]
- [Time delta from previous displayed frame: 0.000036000 seconds]
- [Time since reference or first frame: 6.021297000 seconds]
- Frame Number: 74
- Frame Length: 723 bytes (5784 bits)

The packet bytes pane shows the raw data of the frame, including the Ethernet II header, Internet Protocol Version 4 header, and the TLS Client Hello message.

## Response

