

Shivam Bhat
bhat41@purdue.edu, 0033760929

Q1 Read Chapter

Q2 lab3.pcap file included in zip

Q3
DHCP

The image shows a Wireshark packet capture window titled 'lab3_8.pcap'. The filter is set to 'dhcp'. The packet list shows four packets:

No.	Time	Source	Destination	Protocol	Length	Info
37	0.191828	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x6c983d75
49	0.243851	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x6c983d76
51	0.263628	10.186.112.9	10.186.117.223	DHCP	350	DHCP ACK - Transaction ID 0x6c983d76
52	0.263629	10.186.112.9	10.186.117.223	DHCP	350	DHCP ACK - Transaction ID 0x6c983d76

The packet details pane for packet 49 (DHCP Request) is expanded, showing the following information:

- > Frame 49: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits)
- > Ethernet II, Src: Apple_7f:dc:23 (1c:57:dc:7f:dc:23), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
- > Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
- > User Datagram Protocol, Src Port: 68, Dst Port: 67
- ✓ Dynamic Host Configuration Protocol (Request)
 - Message type: Boot Request (1)
 - Hardware type: Ethernet (0x01)
 - Hardware address length: 6
 - Hops: 0
 - Transaction ID: 0x6c983d76
 - Seconds elapsed: 0
 - > Bootp flags: 0x0000 (Unicast)
 - Client IP address: 0.0.0.0
 - Your (client) IP address: 0.0.0.0
 - Next server IP address: 0.0.0.0
 - Relay agent IP address: 0.0.0.0
 - Client MAC address: Apple_7f:dc:23 (1c:57:dc:7f:dc:23)
 - Client hardware address padding: 00000000000000000000
 - Server host name not given
 - Boot file name not given
 - Magic cookie: DHCP
 - > Option: (53) DHCP Message Type (Request)
 - > Option: (55) Parameter Request List
 - > Option: (57) Maximum DHCP Message Size
 - > Option: (61) Client identifier
 - > Option: (50) Requested IP Address (10.186.117.223)
 - > Option: (51) IP Address Lease Time
 - > Option: (12) Host Name
 - > Option: (255) End
 - Padding: 00000000

lab3_8.pcap

dhcp

No.	Time	Source	Destination	Protocol	Length	Info
37	0.191828	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x6c983d75
49	0.243851	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x6c983d76
51	0.263628	10.186.112.9	10.186.117.223	DHCP	350	DHCP ACK - Transaction ID 0x6c983d76
52	0.263629	10.186.112.9	10.186.117.223	DHCP	350	DHCP ACK - Transaction ID 0x6c983d76

Message type: boot reply (2)
Hardware type: Ethernet (0x01)
Hardware address length: 6
Hops: 2
Transaction ID: 0x6c983d76
Seconds elapsed: 0
> Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0
Your (client) IP address: 10.186.117.223
Next server IP address: 0.0.0.0
Relay agent IP address: 10.186.112.9
Client MAC address: Apple_7f:dc:23 (1c:57:dc:7f:dc:23)
Client hardware address padding: 000000000000000000
Server host name not given
Boot file name not given
Magic cookie: DHCP
> Option: (53) DHCP Message Type (ACK)
> Option: (54) DHCP Server Identifier (10.191.255.21)
> Option: (51) IP Address Lease Time
> Option: (1) Subnet Mask (255.255.240.0)
> Option: (3) Router
Length: 4
Router: 10.186.112.1
> Option: (6) Domain Name Server
> Option: (15) Domain Name
> Option: (44) NetBIOS over TCP/IP Name Server
> Option: (46) NetBIOS over TCP/IP Node Type
> Option: (255) End

ARP Announcement

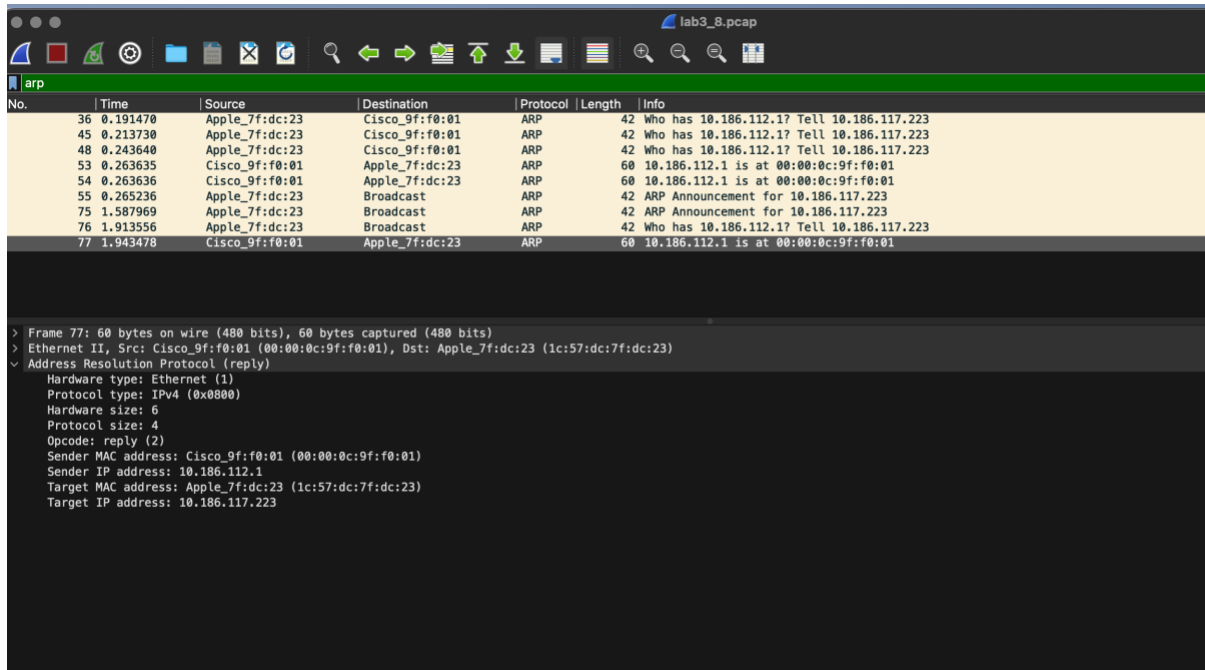
lab3_8.pcap

arp

No.	Time	Source	Destination	Protocol	Length	Info
36	0.191470	Apple_7f:dc:23	Cisco_9f:f0:01	ARP	42	Who has 10.186.112.1? Tell 10.186.117.223
45	0.213730	Apple_7f:dc:23	Cisco_9f:f0:01	ARP	42	Who has 10.186.112.1? Tell 10.186.117.223
48	0.243640	Apple_7f:dc:23	Cisco_9f:f0:01	ARP	42	Who has 10.186.112.1? Tell 10.186.117.223
53	0.263635	Cisco_9f:f0:01	Apple_7f:dc:23	ARP	60	10.186.112.1 is at 00:00:0c:9f:f0:01
54	0.263636	Cisco_9f:f0:01	Apple_7f:dc:23	ARP	60	10.186.112.1 is at 00:00:0c:9f:f0:01
55	0.265236	Apple_7f:dc:23	Broadcast	ARP	42	ARP Announcement for 10.186.117.223
75	1.587969	Apple_7f:dc:23	Broadcast	ARP	42	ARP Announcement for 10.186.117.223
76	1.913556	Apple_7f:dc:23	Broadcast	ARP	42	Who has 10.186.112.1? Tell 10.186.117.223
77	1.943478	Cisco_9f:f0:01	Apple_7f:dc:23	ARP	60	10.186.112.1 is at 00:00:0c:9f:f0:01

> Frame 55: 42 bytes on wire (336 bits), 42 bytes captured (336 bits)
> Ethernet II, Src: Apple_7f:dc:23 (1c:57:dc:7f:dc:23), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Address Resolution Protocol (ARP Announcement)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (1)
[Is gratuitous: True]
[Is announcement: True]
Sender MAC address: Apple_7f:dc:23 (1c:57:dc:7f:dc:23)
Sender IP address: 10.186.117.223
Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)
Target IP address: 10.186.117.223

ARP Response



No.	Time	Source	Destination	Protocol	Length	Info
36	0.191470	Apple_7f:dc:23	Cisco_9f:f0:01	ARP	42	Who has 10.186.112.1? Tell 10.186.117.223
45	0.213730	Apple_7f:dc:23	Cisco_9f:f0:01	ARP	42	Who has 10.186.112.1? Tell 10.186.117.223
48	0.243640	Apple_7f:dc:23	Cisco_9f:f0:01	ARP	42	Who has 10.186.112.1? Tell 10.186.117.223
53	0.263635	Cisco_9f:f0:01	Apple_7f:dc:23	ARP	60	10.186.112.1 is at 00:00:0c:9f:f0:01
54	0.263636	Cisco_9f:f0:01	Apple_7f:dc:23	ARP	60	10.186.112.1 is at 00:00:0c:9f:f0:01
55	0.265236	Apple_7f:dc:23	Broadcast	ARP	42	ARP Announcement for 10.186.117.223
75	1.587969	Apple_7f:dc:23	Broadcast	ARP	42	ARP Announcement for 10.186.117.223
76	1.913556	Apple_7f:dc:23	Broadcast	ARP	42	Who has 10.186.112.1? Tell 10.186.117.223
77	1.943478	Cisco_9f:f0:01	Apple_7f:dc:23	ARP	60	10.186.112.1 is at 00:00:0c:9f:f0:01

```
> Frame 77: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
> Ethernet II, Src: Cisco_9f:f0:01 (00:00:0c:9f:f0:01), Dst: Apple_7f:dc:23 (1c:57:dc:7f:dc:23)
> Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: Cisco_9f:f0:01 (00:00:0c:9f:f0:01)
  Sender IP address: 10.186.112.1
  Target MAC address: Apple_7f:dc:23 (1c:57:dc:7f:dc:23)
  Target IP address: 10.186.117.223
```

DHCP is used for IP address. DHCP stands for Dynamic Host Configuration Protocol, and it is a network protocol commonly used in IP networks to automatically assign IP addresses and other network configuration parameters to devices on a network. As part of this default gateways are also assigned.

We can see DHCPDISCOVER (**DHCP Discover**) messages where the device uses the **0.0.0.0** and **255.255.255.255** as the source address and destination address, respectively. The DHCP exchange close with the DHCKACK (after offer and request) where in the server acknowledges that the client can use the configuration. Here under the router we can see the gateway router ip. -**10.186.112.1** The offer and request are not show. This could be because of many reasons one of which could be encryption.

For mac addresses we use the ARP packets. ARP (Address Resolution Protocol) is a network protocol used to map IP (Internet Protocol) addresses to MAC (Media Access Control) addresses. The ARP request is broadcasted, and the device with the corresponding IP address responds with its MAC address. We can in line 77 my machine Apple_7f receives the resolved MAC address.

Hence First DHCP is used to obtain the ip address and gateway router ip following which ARP is used to obtain MAC addresses. We can see the DHCP ACK happens at 52 and MAC address is obtained via ARP at 53.

Also if we notice the packet sequence for this pcap we see that there is one ARP packet just before DHCP at 36,37 respectively. This could be a result of host already caching its previous IP and then probed to check if that's still unchanged or taken by someone else. Here it then sees through the DHCP reply that this IP is free and hence continues to take that for ARP. This use of caching speeds up the processes and helps handle failure fallbacks. None the less for new hosts , first DHCP will happen then ARP.

B)

DNS Query

No.	Time	Source	Destination	Protocol	Length	Info
464	21.177834	10.186.117.223	128.210.11.57	DNS	74	Standard query 0x8e1c A www.google.com
465	21.177893	10.186.117.223	128.210.11.57	DNS	74	Standard query 0x84f0 AAAA www.google.com
466	21.177899	10.186.117.223	128.210.11.57	DNS	74	Standard query 0xc220 HTTPS www.google.com
467	21.313251	128.210.11.57	10.186.117.223	DNS	90	Standard query response 0x8e1c A www.google.com A 142.250.190.68
468	21.313252	128.210.11.57	10.186.117.223	DNS	102	Standard query response 0x84f0 AAAA www.google.com AAAA 2607:f8b0:4009:80a::2004
469	21.313252	128.210.11.57	10.186.117.223	DNS	74	Standard query response 0xc220 HTTPS www.google.com
566	22.173157	10.186.117.223	128.210.11.57	DNS	80	Standard query 0xf5b3 A adservice.google.com
567	22.173170	10.186.117.223	128.210.11.57	DNS	80	Standard query 0x3eb4 AAAA adservice.google.com
568	22.173196	10.186.117.223	128.210.11.57	DNS	80	Standard query 0xa021 HTTPS adservice.google.com
569	22.309832	128.210.11.57	10.186.117.223	DNS	96	Standard query response 0xf5b3 A adservice.google.com A 142.250.190.34
570	22.309833	128.210.11.57	10.186.117.223	DNS	108	Standard query response 0x3eb4 AAAA adservice.google.com AAAA 2607:f8b0:4009:804::2002
572	22.309834	128.210.11.57	10.186.117.223	DNS	80	Standard query response 0xa021 HTTPS adservice.google.com
994	27.726845	10.186.117.223	128.210.11.57	DNS	73	Standard query 0xb94e A id.google.com
995	27.726882	10.186.117.223	128.210.11.57	DNS	73	Standard query 0x03b8 AAAA id.google.com
996	27.726923	10.186.117.223	128.210.11.57	DNS	73	Standard query 0x4260 HTTPS id.google.com
998	27.739982	128.210.11.57	10.186.117.223	DNS	89	Standard query response 0xb94e A id.google.com A 142.250.190.35
1038	27.748858	128.210.11.57	10.186.117.223	DNS	73	Standard query response 0x4260 HTTPS id.google.com
1068	27.750001	128.210.11.57	10.186.117.223	DNS	101	Standard query response 0x03b8 AAAA id.google.com AAAA 2607:f8b0:400c:c15::5e
1476	28.093682	10.186.117.223	128.210.11.57	DNS	75	Standard query 0x0ca9 A play.google.com
1477	28.093745	10.186.117.223	128.210.11.57	DNS	75	Standard query 0xa721 AAAA play.google.com

> Frame 464: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
 > Ethernet II, Src: Apple_7f:dc:23 (1c:57:dc:7f:dc:23), Dst: Cisco_9f:f0:01 (00:00:0c:9f:f0:01)
 > Internet Protocol Version 4, Src: 10.186.117.223, Dst: 128.210.11.57
 > User Datagram Protocol, Src Port: 64812, Dst Port: 53
 > Domain Name System (query)
 > Transaction ID: 0x8e1c
 > Flags: 0x0100 Standard query
 > Questions: 1
 > Answer RRs: 0
 > Authority RRs: 0
 > Additional RRs: 0
 > Queries
 > www.google.com: type A, class IN
 [Response In: 467]

DNS Response

No.	Time	Source	Destination	Protocol	Length	Info
464	21.177834	10.186.117.223	128.210.11.57	DNS	74	Standard query 0x8e1c A www.google.com
465	21.177893	10.186.117.223	128.210.11.57	DNS	74	Standard query 0x84f0 AAAA www.google.com
466	21.177899	10.186.117.223	128.210.11.57	DNS	74	Standard query 0xc220 HTTPS www.google.com
467	21.313251	128.210.11.57	10.186.117.223	DNS	90	Standard query response 0x8e1c A www.google.com A 142.250.190.68
468	21.313252	128.210.11.57	10.186.117.223	DNS	102	Standard query response 0x84f0 AAAA www.google.com AAAA 2607:f8b0:4009:80a::2004
469	21.313252	128.210.11.57	10.186.117.223	DNS	74	Standard query response 0xc220 HTTPS www.google.com
566	22.173157	10.186.117.223	128.210.11.57	DNS	80	Standard query 0xf5b3 A adservice.google.com
567	22.173170	10.186.117.223	128.210.11.57	DNS	80	Standard query 0x3eb4 AAAA adservice.google.com
568	22.173196	10.186.117.223	128.210.11.57	DNS	80	Standard query 0xa021 HTTPS adservice.google.com
569	22.309832	128.210.11.57	10.186.117.223	DNS	96	Standard query response 0xf5b3 A adservice.google.com A 142.250.190.34
570	22.309833	128.210.11.57	10.186.117.223	DNS	108	Standard query response 0x3eb4 AAAA adservice.google.com AAAA 2607:f8b0:4009:804::2002
572	22.309834	128.210.11.57	10.186.117.223	DNS	80	Standard query response 0xa021 HTTPS adservice.google.com
994	27.726845	10.186.117.223	128.210.11.57	DNS	73	Standard query 0xb94e A id.google.com
995	27.726882	10.186.117.223	128.210.11.57	DNS	73	Standard query 0x03b8 AAAA id.google.com
996	27.726923	10.186.117.223	128.210.11.57	DNS	73	Standard query 0x4260 HTTPS id.google.com
998	27.739982	128.210.11.57	10.186.117.223	DNS	89	Standard query response 0xb94e A id.google.com A 142.250.190.35
1038	27.748858	128.210.11.57	10.186.117.223	DNS	73	Standard query response 0x4260 HTTPS id.google.com
1068	27.750001	128.210.11.57	10.186.117.223	DNS	101	Standard query response 0x03b8 AAAA id.google.com AAAA 2607:f8b0:400c:c15::5e
1476	28.093682	10.186.117.223	128.210.11.57	DNS	75	Standard query 0x0ca9 A play.google.com
1477	28.093745	10.186.117.223	128.210.11.57	DNS	75	Standard query 0xa721 AAAA play.google.com

> Frame 467: 90 bytes on wire (720 bits), 90 bytes captured (720 bits)
 > Ethernet II, Src: Cisco_79:64:c1 (00:de:fb:79:64:c1), Dst: Apple_7f:dc:23 (1c:57:dc:7f:dc:23)
 > Internet Protocol Version 4, Src: 128.210.11.57, Dst: 10.186.117.223
 > User Datagram Protocol, Src Port: 53, Dst Port: 64812
 > Domain Name System (response)
 > Transaction ID: 0x8e1c
 > Flags: 0x0100 Standard query response, No error
 > Questions: 1
 > Answer RRs: 1
 > Authority RRs: 0
 > Additional RRs: 0
 > Queries
 > Answers
 > www.google.com: type A, class IN, addr 142.250.190.68
 [Request In: 464]
 [Time: 0.135417000 seconds]

To locate the google DNS resolution I used the following query
dns.qry.name contains "google.com".

I then searched for all the packet where it looked for google.com. There we can see queries followed by responses. In the response we can check the answers of DNS responses(Answers) field to identify the ip which is **142.250.190.68**

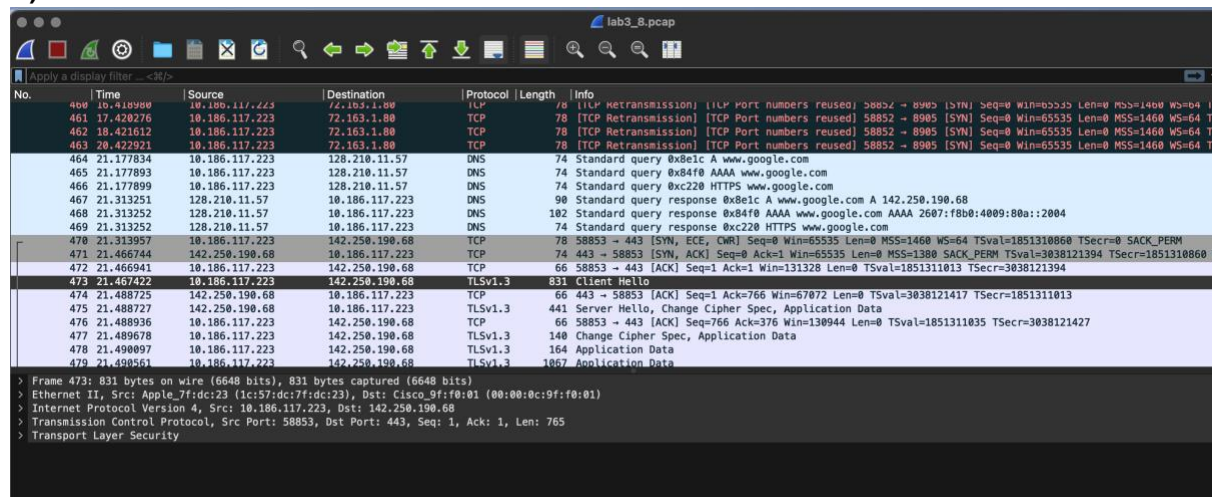
C)

I am not able to see any intra-AS or inter-AS routing. Wireshark records and examines network data at the packet level, enabling us to look inside packets as they are sent over a network. However, intra-AS and inter-AS routing are more complex higher-level network operations occurring at the network layer and are frequently invisible to Wireshark's packet-level analysis.

Routing information is exchanged between routers within and between autonomous systems, respectively, in intra-AS and inter-AS routing. The most efficient route for traffic to take both within and between networks is chosen using this information. This routing information is exchanged using routing protocols like OSPF, BGP which operate at a higher layer of network stack than what wireshark operates in.

Also, when using wireshark we are using WiFi to capture the packets. My machine connects to an AP which then enables a connection to the internet. Hence there is no need for AS routing in the host as its function is just to transmit packets. AP takes care of the routing

D)



lab3_8.pcap

Apply a display filter: <filter>

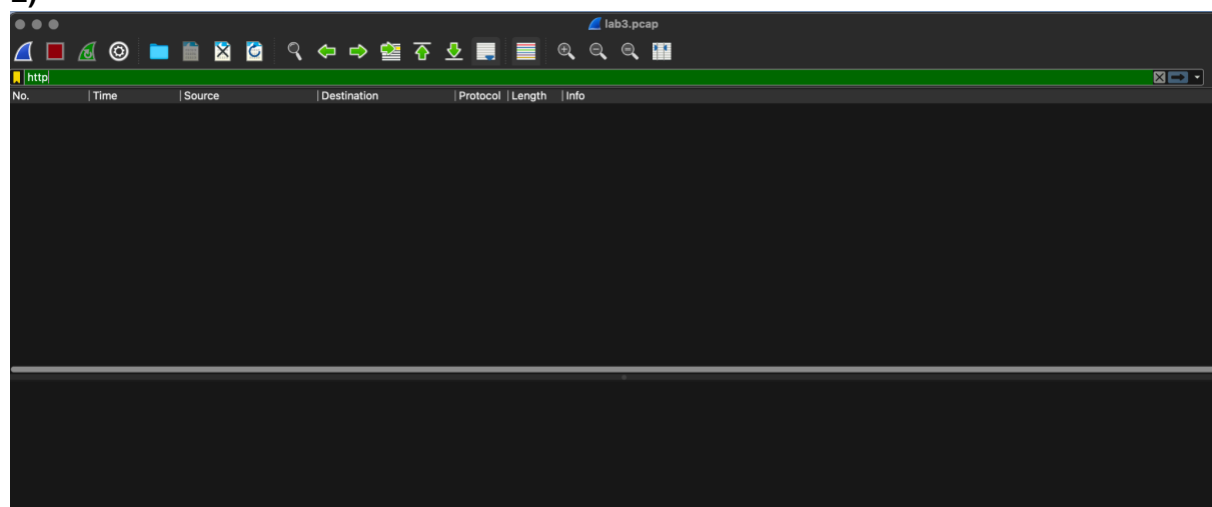
No.	Time	Source	Destination	Protocol	Length	Info
406	10.418988	10.186.117.223	72.163.1.80	TCP	78	[TCP Retransmission] [TCP Port numbers reused] 58852 → 8985 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 T
461	17.420276	10.186.117.223	72.163.1.80	TCP	78	[TCP Retransmission] [TCP Port numbers reused] 58852 → 8985 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 T
462	18.421612	10.186.117.223	72.163.1.80	TCP	78	[TCP Retransmission] [TCP Port numbers reused] 58852 → 8985 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 T
463	20.422921	10.186.117.223	72.163.1.80	TCP	78	[TCP Retransmission] [TCP Port numbers reused] 58852 → 8985 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 T
464	21.177834	10.186.117.223	128.210.11.57	DNS	74	Standard query 0x84f0 AAAA www.google.com
465	21.177893	10.186.117.223	128.210.11.57	DNS	74	Standard query 0xc220 HTTPS www.google.com
466	21.177899	10.186.117.223	128.210.11.57	DNS	74	Standard query 0xc220 HTTPS www.google.com
467	21.313251	128.210.11.57	10.186.117.223	DNS	90	Standard query response 0x84f0 AAAA www.google.com A 142.250.190.68
468	21.313252	128.210.11.57	10.186.117.223	DNS	102	Standard query response 0xc220 HTTPS www.google.com AAAA 2607:f8b0:4009:80a::2004
469	21.313252	128.210.11.57	10.186.117.223	DNS	74	Standard query response 0xc220 HTTPS www.google.com
470	21.313957	10.186.117.223	142.250.190.68	TCP	78	58853 → 443 [SYN, ECE, CWR] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=1851310860 TSecr=0 SACK_PERM
471	21.466744	142.250.190.68	10.186.117.223	TCP	74	443 → 58853 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1380 SACK_PERM TSval=3038121394 TSecr=1851310860
472	21.466941	10.186.117.223	142.250.190.68	TCP	66	58853 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0 TSval=1851311013 TSecr=3038121394
473	21.467422	10.186.117.223	142.250.190.68	TLSv1.3	831	Client Hello
474	21.488725	142.250.190.68	10.186.117.223	TCP	66	443 → 58853 [ACK] Seq=1 Ack=766 Win=67072 Len=0 TSval=3038121417 TSecr=1851311013
475	21.488727	142.250.190.68	10.186.117.223	TLSv1.3	441	Server Hello, Change Cipher Spec, Application Data
476	21.488936	10.186.117.223	142.250.190.68	TCP	66	58853 → 443 [ACK] Seq=766 Ack=376 Win=130944 Len=0 TSval=1851311035 TSecr=3038121427
477	21.489678	10.186.117.223	142.250.190.68	TLSv1.3	148	Change Cipher Spec, Application Data
478	21.490897	10.186.117.223	142.250.190.68	TLSv1.3	164	Application Data
479	21.490851	10.186.117.223	142.250.190.68	TLSv1.3	1867	Application Data

> Frame 473: 831 bytes on wire (6648 bits), 831 bytes captured (6648 bits)
 > Ethernet II, Src: Apple_7f:dc:23 (1c:57:dc:7f:dc:23), Dst: Cisco_9f:f0:01 (00:00:0c:9f:f0:01)
 > Internet Protocol Version 4, Src: 10.186.117.223, Dst: 142.250.190.68
 > Transmission Control Protocol, Src Port: 58853, Dst Port: 443, Seq: 1, Ack: 1, Len: 765
 > Transport Layer Security

We can see the TCP handshake in the first 3 lines- 470-472.

We can also use the 'tcp' filter in search bar

E)



lab3.pcap

http

No.	Time	Source	Destination	Protocol	Length	Info
-----	------	--------	-------------	----------	--------	------

Since google.com is using the TLS protocol for encryptions, I can't see any explicit HTTP messages. So there are no HTTP GET messages as well. I can see TLS packets with encrypted messages(payload) being sent from google.com

lab3_8.pcap

ip.dst==142.250.190.68 and tls

No.	Time	Source	Destination	Protocol	Length	Info
473	21.467422	10.186.117.223	142.250.190.68	TLSv1.3	831	Client Hello
477	21.489678	10.186.117.223	142.250.190.68	TLSv1.3	140	Change Cipher Spec, Application Data
478	21.490097	10.186.117.223	142.250.190.68	TLSv1.3	164	Application Data
479	21.490561	10.186.117.223	142.250.190.68	TLSv1.3	1067	Application Data
486	21.625493	10.186.117.223	142.250.190.68	TLSv1.3	97	Application Data
539	21.958196	10.186.117.223	142.250.190.68	TLSv1.3	105	Application Data
541	21.991586	10.186.117.223	142.250.190.68	TLSv1.3	341	Application Data
542	22.002140	10.186.117.223	142.250.190.68	TLSv1.3	195	Application Data
543	22.002795	10.186.117.223	142.250.190.68	TLSv1.3	295	Application Data
544	22.005216	10.186.117.223	142.250.190.68	TLSv1.3	240	Application Data
545	22.020292	10.186.117.223	142.250.190.68	TLSv1.3	193	Application Data
551	22.132684	10.186.117.223	142.250.190.68	TLSv1.3	105	Application Data
563	22.166624	10.186.117.223	142.250.190.68	TLSv1.3	105	Application Data
564	22.170527	10.186.117.223	142.250.190.68	TLSv1.3	463	Application Data
565	22.171391	10.186.117.223	142.250.190.68	TLSv1.3	195	Application Data
581	22.313342	10.186.117.223	142.250.190.68	TLSv1.3	105	Application Data
610	22.638605	10.186.117.223	142.250.190.68	TLSv1.3	105	Application Data
639	24.981085	10.186.117.223	142.250.190.68	TLSv1.3	195	Application Data
640	24.981473	10.186.117.223	142.250.190.68	TLSv1.3	228	Application Data
641	24.982131	10.186.117.223	142.250.190.68	TLSv1.3	225	Application Data
642	24.999912	10.186.117.223	142.250.190.68	TLSv1.3	210	Application Data
643	25.027315	10.186.117.223	142.250.190.68	TLSv1.3	195	Application Data
644	25.027753	10.186.117.223	142.250.190.68	TLSv1.3	229	Application Data
661	25.115207	10.186.117.223	142.250.190.68	TLSv1.3	105	Application Data
662	25.224931	10.186.117.223	142.250.190.68	TLSv1.3	195	Application Data
663	25.225464	10.186.117.223	142.250.190.68	TLSv1.3	230	Application Data
676	25.257913	10.186.117.223	142.250.190.68	TLSv1.3	105	Application Data
683	25.286943	10.186.117.223	142.250.190.68	TLSv1.3	105	Application Data
684	25.343168	10.186.117.223	142.250.190.68	TLSv1.3	195	Application Data

> Frame 478: 164 bytes on wire (1312 bits), 164 bytes captured (1312 bits)

> Ethernet II, Src: Apple_7f:dc:23 (1c:57:dc:7f:dc:23), Dst: Cisco_9f:f0:01 (00:00:0c:9f:f0:01)

> Internet Protocol Version 4, Src: 10.186.117.223, Dst: 142.250.190.68

> Transmission Control Protocol, Src Port: 58853, Dst Port: 443, Seq: 840, Ack: 376, Len: 98

> Transport Layer Security

 < TLSv1.3 Record Layer: Application Data Protocol: Hypertext Transfer Protocol

 Opaque Type: Application Data (23)

 Version: TLS 1.2 (0x0303)

 Length: 93

 Encrypted Application Data: 419a80ba643610094915c16f51588d0ddc7e4021f5cd8cfeff824fb4d5f88a7c73b9b010_

 [Application Data Protocol: Hypertext Transfer Protocol]

Q4

On Amber


```
data 70 $ python3 lab3.py A "lab3.pcap"
CASE A:
IPAddr: 10.186.112.1
MACAddr: 00:00:0c:9f:f0:01
data 71 $ python3 lab3.py B "lab3.pcap" "www.google.com"
CASE B:
IPAddr-DEST: 142.250.190.68
data 72 $ python3 lab3.py C "lab3.pcap" "www.google.com"
CASE C:
IPAddr-SRC: 10.186.117.223
IPAddr-DEST: 142.250.190.68
Port-DEST: 443
SYN: 1
ACK: 0
IPAddr-SRC: 142.250.190.68
IPAddr-DEST: 10.186.117.223
Port-DEST: 443
SYN: 1
ACK: 1
IPAddr-SRC: 10.186.117.223
IPAddr-DEST: 142.250.190.68
Port-DEST: 443
SYN: 0
ACK: 1
data 73 $ python3 lab3.py ALL "lab3.pcap" "www.google.com"
CASE A:
IPAddr: 10.186.112.1
MACAddr: 00:00:0c:9f:f0:01
CASE B:
IPAddr-DEST: 142.250.190.68
CASE C:
IPAddr-SRC: 10.186.117.223
IPAddr-DEST: 142.250.190.68
Port-DEST: 443
SYN: 1
ACK: 0
IPAddr-SRC: 142.250.190.68
IPAddr-DEST: 10.186.117.223
Port-DEST: 443
SYN: 1
ACK: 1
IPAddr-SRC: 10.186.117.223
IPAddr-DEST: 142.250.190.68
```

On Local

```
~/.Doc/P/PurduePrivate/l/c/l3/Lab3_0033760929 devCNS *4 ?14 > python3 lab3.py A "lab3.pcap" base
CASE A:
IPAddr: 10.186.112.1
MACAddr: 00:00:0c:9f:f0:01
~/.Doc/P/PurduePrivate/l/c/l3/Lab3_0033760929 devCNS *4 ?14 > python3 lab3.py B "lab3.pcap" "www.google.com"
CASE B:
IPAddr-DEST: 142.250.190.68
~/.Doc/P/PurduePrivate/l/c/l3/Lab3_0033760929 devCNS *4 ?14 > python3 lab3.py C "lab3.pcap" "www.google.com" base
CASE C:
IPAddr-SRC: 10.186.117.223
IPAddr-DEST: 142.250.190.68
Port-DEST: 443
SYN: 1
ACK: 0
IPAddr-SRC: 142.250.190.68
IPAddr-DEST: 10.186.117.223
Port-DEST: 443
SYN: 1
ACK: 1
IPAddr-SRC: 10.186.117.223
IPAddr-DEST: 142.250.190.68
Port-DEST: 443
SYN: 0
ACK: 1
~/.Doc/P/PurduePrivate/l/c/l3/Lab3_0033760929 devCNS *4 ?14 > python3 lab3.py ALL "lab3.pcap" "www.google.com" base
CASE A:
IPAddr: 10.186.112.1
MACAddr: 00:00:0c:9f:f0:01
CASE B:
IPAddr-DEST: 142.250.190.68
CASE C:
IPAddr-SRC: 10.186.117.223
IPAddr-DEST: 142.250.190.68
Port-DEST: 443
SYN: 1
ACK: 0
IPAddr-SRC: 142.250.190.68
IPAddr-DEST: 10.186.117.223
Port-DEST: 443
SYN: 1
ACK: 1
IPAddr-SRC: 10.186.117.223
IPAddr-DEST: 142.250.190.68
Port-DEST: 443
SYN: 0
ACK: 1
```

Instruction

For handshakes I print one complete handshake **SYN =1, SYN =1 ACK=1, SYN=0 ACK =1** as mentioned by TA at <https://campuswire.com/c/GC1863205/feed/381>

I have disabled QUIC as mentioned by TA in <https://campuswire.com/c/GC1863205/feed/377>

Command : Python3 lab3.py MODE FILEPATH WEBSITE

Import requirements

Install scapy

import logging

import logging is used for suppressing warning as mentioned by TA at <https://campuswire.com/c/GC1863205/feed/395>